МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
Факультет електроніки та інформаційних технологій

Кафедра комп'ютерних наук

Кваліфікаційна робота магістра

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ КЛАСИФІКАЦІЙНОГО АНАЛІЗУ КОРИСТУВАЧІВ ІНТЕРНЕТ МАГАЗИНУ

Здобувач освіти гр. ІН.м-12ан Юрій ЖУРАВЛЬОВ

Науковий керівник,
доцент кафедри комп'ютерних наук,
к.т.н., доцент В'ячеслав МОСКАЛЕНКО

в.о. завідувач кафедри
доцент кафедри комп'ютерних наук,
к.т.н., доцент Ігор ШЕЛЕХОВ

Суми 2022

Сумський державний університет

(назва вузу)

Факультет *ЕлІТ* Кафедра *Комп'ютерних наук*

Спеціальність *«122 - Комп'ютерні науки»*

Затверджую:

зав.кафедрою _____

_____

"_____" _____ 20___р.

ЗАВДАННЯ
НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) СТУДЕНТОВІ

Журавльову Юрію Олександровичу

(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) *Інформаційна технологія класифікаційного аналізу користувачів інтернет магазину.*

затверджую наказом по інституту від "____" _____ 20 ___ р. № _____

2. Термін здачі студентом закінченого проекту (роботи)_____

3. Вхідні данні до проекту (роботи) _____

_____

_____

_____

4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити) *1) аналіз проблеми аналізу трубних інспекцій; 2) формалізована постановка задачі дослідження; 3) опис інформаційної технології; 4) опис програмної реалізації; 6) аналіз результатів.*

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

_____

6. Консультанти до проекту (роботи), із значенням розділів проекту, що стосується їх

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання _____

Керівник _____
(підпис)

Завдання прийняв до виконання _____
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів дипломного проекту (роботи)	Термін виконання проекту (роботи)	Примітка
1.	Аналіз проблеми сегментації користувачів		
2.	Формалізована постановка задачі дослідження		
3.	Опис інформаційної технології		
4.	Опис програмної реалізації		
5.	Аналіз результатів		
6.	Оформлення пояснювальної записки до кваліфікаційної магістерської роботи		

Студент – дипломник _____
(підпис)

Керівник проекту _____
(підпис

РЕФЕРАТ

Примітка:47 сторінок, 33 рисунків, 1 таблиць, 1 додатків, 33 джерел.

Мета роботи– підвищення ефективності сегментного аналізу користувачів інтернет-магазину.

Об'єкт дослідження– формалізований процес сегментації користувачів інтернет-магазину.

Предмет дослідження– інформаційна технологія сегментації клієнтів інтернет-магазину з використанням машинного навчання.

Методи дослідження— класичні методи машинного навчання, штучні нейронні мережі.

СЕГМЕНТАЦІЯ, КЛАСИФІКАЦІЯ, КЛАСТЕРИЗАЦІЯ, ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ, СЕГМЕНТНИЙ АНАЛІЗ

ЗМІСТ

ВСТУП

В останні роки популярність інтернет-магазинів значно зросла. Це набагато ефективніше, зручніше і має набагато більше товарів, ніж традиційні магазини. Одна з їх найкращих особливостей це рекомендації, які більшість електронних магазинів надають своїм клієнтам на основі їх уподобань.

Усі переваги онлайн-магазинів досягаються завдяки тому, що такі магазини можуть збирати різноманітні дані, як сесії клієнтів, час і тривалість сеансу, які товари переглядає клієнт, які товари купує клієнт, звідки почався сеанс користувача. Після цього вся ця інформація аналізується для побудови оптимальної стратегії обслуговування електронного магазину.

Одним із способів такої класифікації є сегментація клієнтів.

Отже, цілі цієї роботи полягають у тому, щоб дослідити, як ми можемо сегментувати клієнтів електронного магазину, якими методами ми можемо їх сегментувати, і розробити програмне рішення, яке аналізуватиме надані дані онлайн-магазину та на основі даних буде додавати користувача до певного сегменту.

Мета роботи– підвищення ефективності сегментного аналізу користувачів інтернет-магазину.

Об'єкт дослідження– формалізований процес сегментації користувачів інтернет-магазину.

Предмет дослідження– інформаційна технологія сегментації клієнтів інтернет-магазину з використанням машинного навчання.

Методи дослідження— класичні методи машинного навчання, штучні нейронні мережі.

1. ОГЛЯД ІНФОРМАЦІЇ ТА ПОСТАНОВКА ЗАДАЧІ

1.1. Сучасний стан і тенденції розвитку сегментного аналізу клієнтів електронного магазину

Сегментація клієнтів — це процес поділу клієнтів на різні типи на основі подібності властивостей і характеристик, які вони мають[1–3]. Але як ми можемо сегментувати клієнтів? Які особливості ми повинні враховувати?

Сегментація клієнтів використовується з різних причин, наприклад:

- Надання клієнтам правильних рекомендацій на основі їхніх смаків.
- Створення таргетованої реклами.
- Надавати кращі послуги.
- Підвищення лояльності клієнтів.
- Розуміння типів клієнтів, з якими працює інтернет-магазин.

Загалом сегментацію клієнтів можна розділити на два типи: «сегментація клієнтів на основі того, ким вони є» і «сегментація клієнтів на основі того, що вони роблять».[2,4,5]. Виходячи з особливостей, які ми розглядаємо, ці типи можна розділити на різні менші моделі, наприклад[6]:

- Сегментація за демографічною ознакою (типові ознаки: вік, стать, дохід, освіта тощо).
- Сегментація за місцем знаходження (типові ознаки: країна, штат, місто тощо).
- Сегментація за використовуваними технологіями (типові характеристики: використання мобільних пристроїв, використання комп’ютерів, програми, програмне забезпечення тощо).

– Сегментація за поведінкою (типові ознаки: схильності та часті дії, особливість або використання товару, звички, уподобання тощо).

Звичайно, ці моделі можна змінювати або комбінувати (наприклад, поєднати демографічну та географічну в одну).

Тепер давайте розглянемо інструменти, які можна використовувати для сегментації клієнтів.

Google Analytics (GA) – це служба, яку надає Google для аналізу та створення детальної статистики використання веб-сайту користувачами на основі даних, зібраних із сеансів користувача (Рисунок 1.1). Його можна використовувати для сегментації клієнтів.

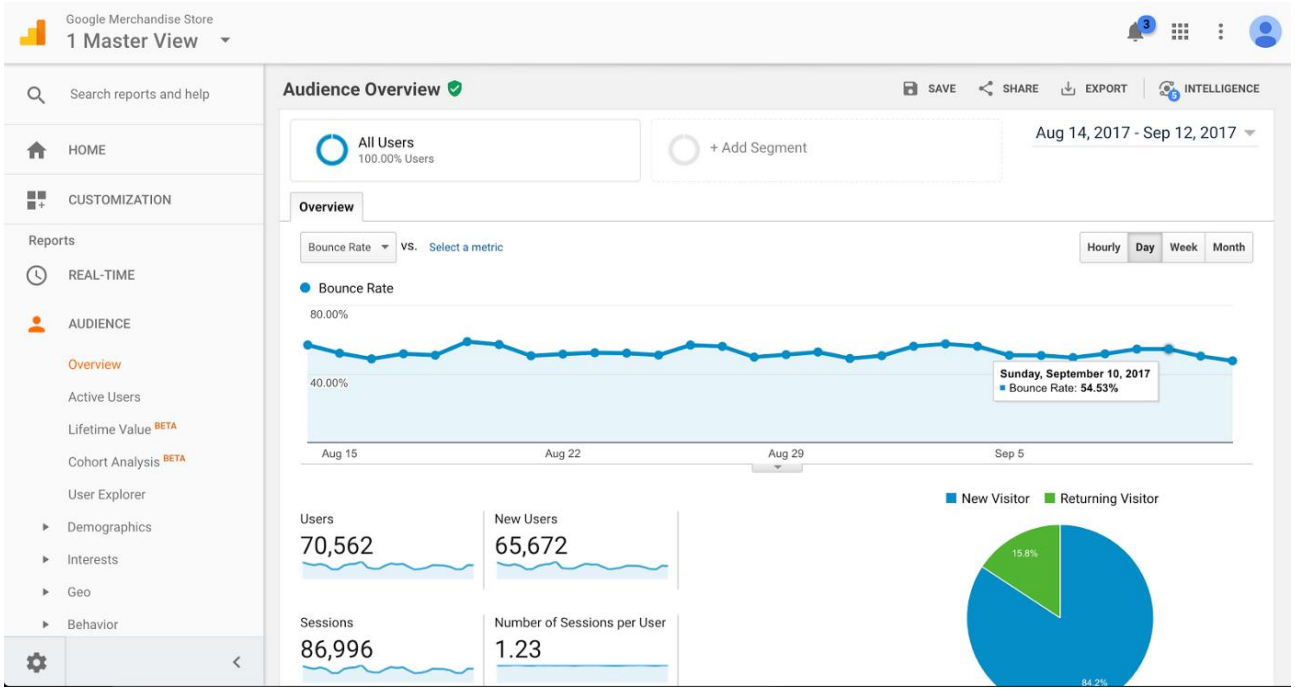

Рисунок 1.1– Google Analytics

У GA ми можемо використовувати попередньо створені сегменти або створювати власні (Рисунок 1.2)[7].

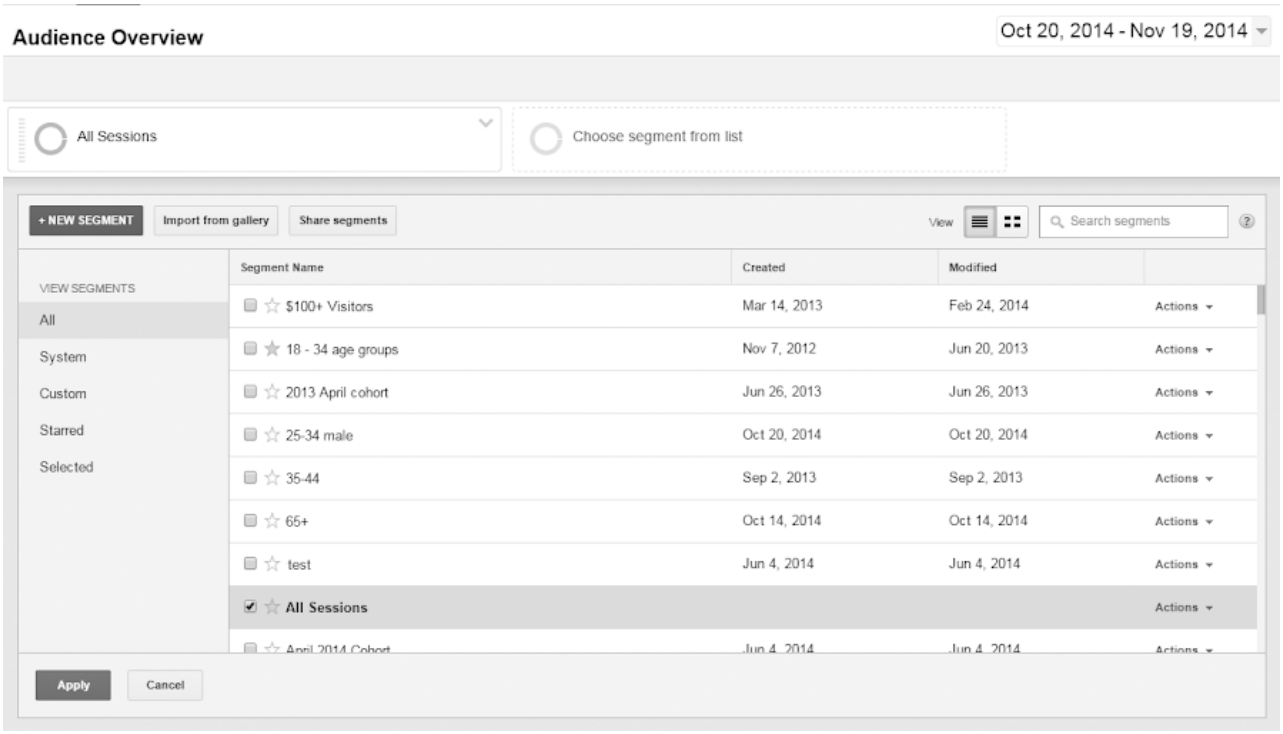

Рисунок 1.2 – Google Analytics, створення сегментів

Для створення сегментів в GA ми можемо встановити параметри, розділені на вкладках демографічні дані, технології, поведінка, дата першого сеансу, джерела трафіку, електронна комерція (Рисунок 1.3).

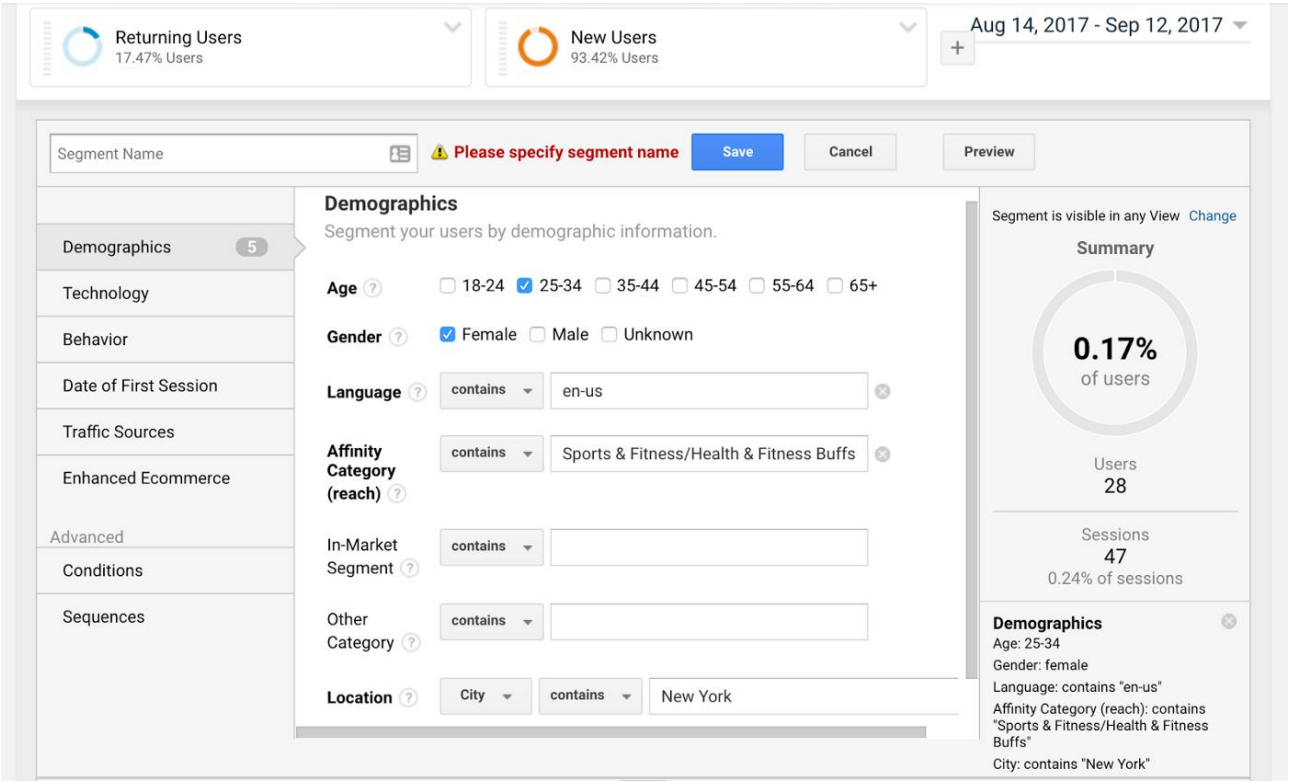

Рисунок 1.3 – Параметри сегментів в Google Analytics

Google Analytics також має галерею рішень (Рисунок 1.4), тому також є можливість імпортувати та експортувати власні сегменти та ділитися ними зі спільнотою.

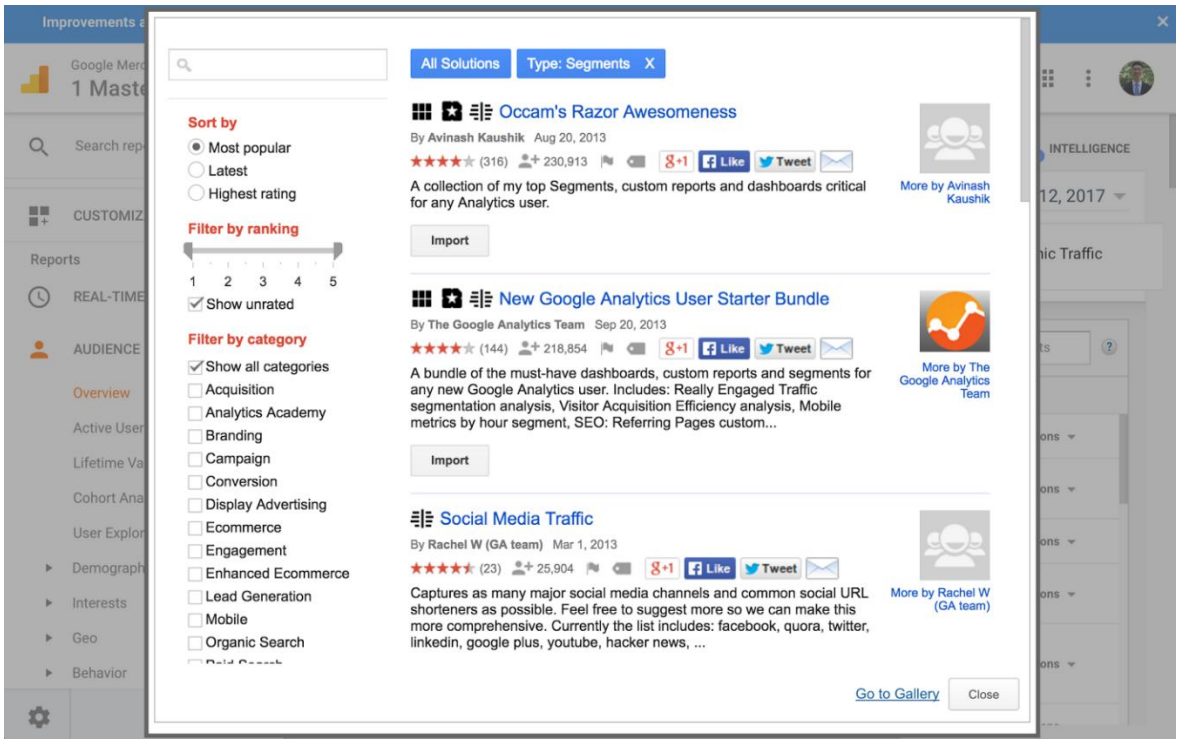

Рисунок 1.4 – Галерея аналітичних рішень.

Google Analytics надає безкоштовні інструменти для сегментації користувачів веб-сайту на основі сеансів користувачів на цьому веб-сайті. Однак сегменти повністю визначаються користувачами (потрібно встановити вік, стать, мову тощо для фільтрації користувачів), і вони не створюються на основі подібності автоматично.

Qualtrics (Рисунок1.5) — це програмне забезпечення для сегментації клієнтів, яке пропонує інструменти сегментації для ваших клієнтів і продуктів[1].

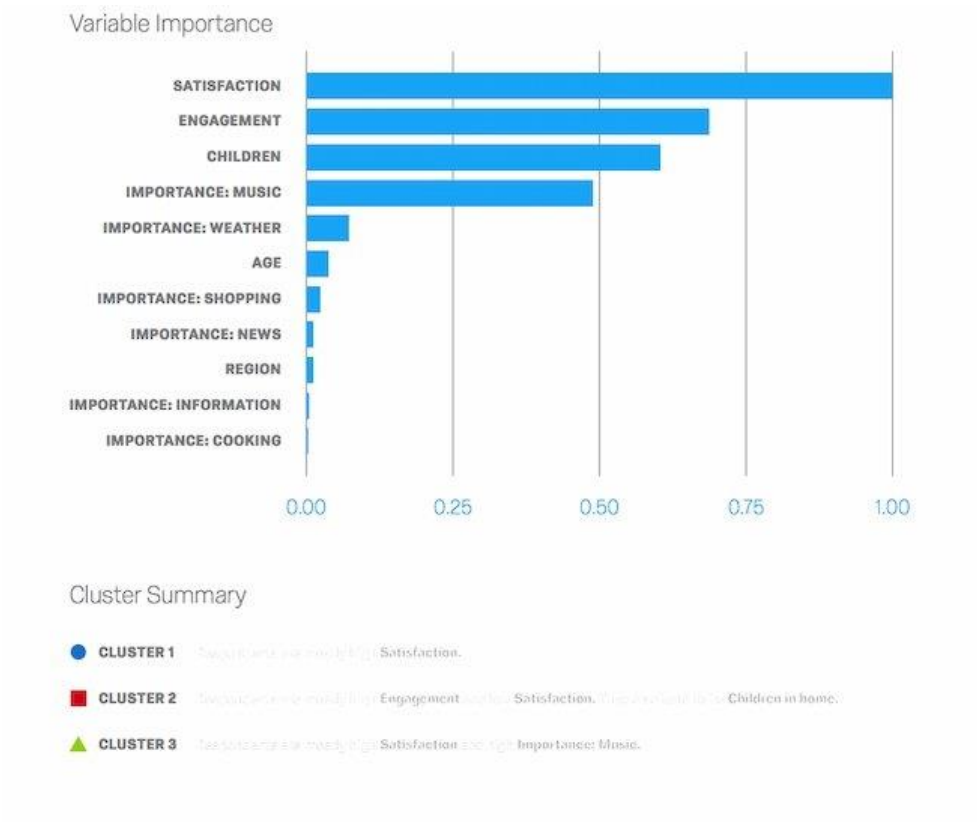

Рисунок1.5– Qualtrics

Qualtrics має можливості машинного навчання, щоб допомогти вам навчитися нових способів сегментувати своїх клієнтів. Однак це не безкоштовно і коштує 1500 доларів на рік.

З Baremetrics (Рисунок 1.6), ви можете легко отримати доступ до ключових фінансових показників і проаналізувати, як різні групи клієнтів впливають на дохід. Коротше кажучи, ви можете відстежувати MRR, ARR, ARPU, пробні версії, оновлення, скасування, відшкодування тощо.[8].

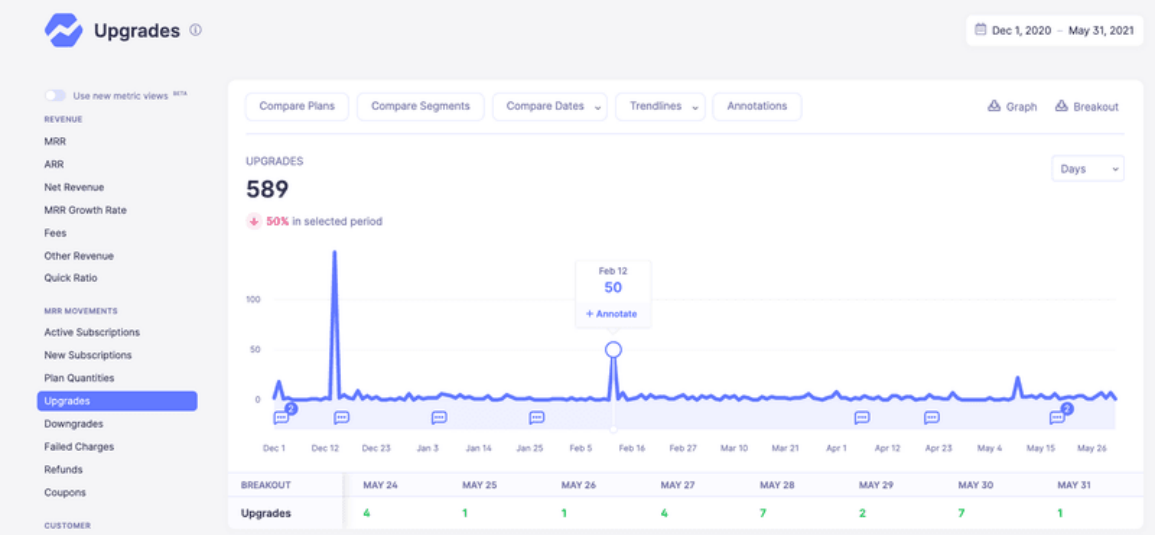

Рисунок 1.6 – Baremetrics

Це зручний інструмент для початку роботи та має безкоштовну версію. Однак ви не можете визначити функції, які вам потрібні для ваших сегментів, ви можете використовувати лише попередньо створені, а ціна цього інструменту з усіма функціями починається від 129 доларів США на місяць.

Отже, головна відмінність між безкоштовними та платними інструментами полягає в тому, що для безкоштовних інструментів вам потрібно визначити всі параметри сегмента (наприклад, вік, стать, країна походження користувача тощо), але платні версії надають рекомендації або автоматично створені сегменти. Це головна функція, яка повинна бути забезпечена в рішенні, на яке спрямована ця робота.

1.2.Моделі та методи кластерного аналізу даних

Завдання кластеризації подібне до завдання сегментації, яка полягає у групуванні кількох об’єктів у підмножини (кластери) таким чином, щоб об’єкти з одного кластера були більш схожі один на одного, ніж об’єкти з інших кластерів за будь-яким критерієм.

Зробимо короткий огляд алгоритмів кластеризації.

Ієрархічна кластеризація — це загальне сімейство алгоритмів кластеризації, які будують вкладені кластери шляхом їх послідовного злиття або розбиття. Ця ієрархія кластерів представлена у вигляді дерева (або дендрограми). Корінь дерева – це унікальний кластер, який збирає всі зразки, а листя – це кластери лише з одним зразком[9,10].

Об’єкт АНС виконує ієрархічну кластеризацію, використовуючи висхідний підхід: кожне спостереження починається з власного кластера, а кластери послідовно об’єднуються.

Етапи виконання АНС[11]:

1. На початку ми маємо K кластерів і точок даних. Кожну точку даних слід розглядати як один кластер.
2. Сформуйте новий кластер, приєднавшись до найближчих точок даних. Тепер у нас є кластери K-1.
3. Повторюйте крок 2, доки K не дорівнюватиме 0 і більше не залишиться точок даних.
4. Нарешті сформувавши один великий кластер, ми можемо використовувати дендрограми, щоб розділити кластери на кілька кластерів залежно від варіанту використання.

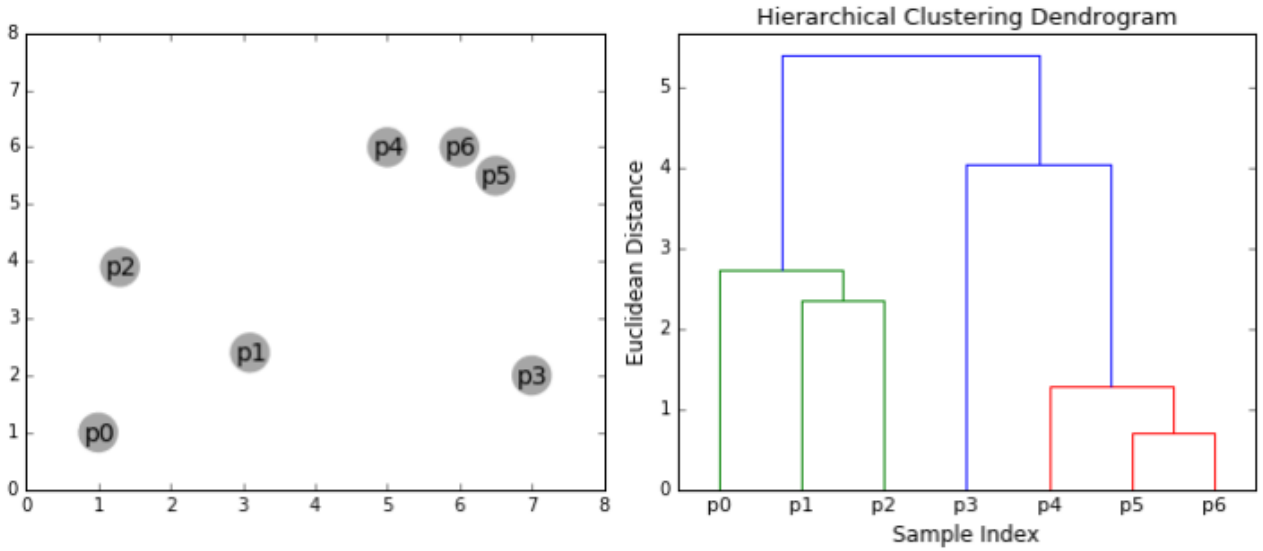

Рисунок 1.7 – Дендрограмма ієрархічної кластеризації

Переваги[10,11]:

- Легко реалізувати.
- Немає заздалегідь визначеної кількості кластерів.

Недоліки[10,11]:

- Об’єкти, неправильно згруповані на будь-яких кроках на початку, не можна перегрупувати.
- Алгоритми ієрархічної кластеризації не забезпечують унікального поділу набору даних.
- Погано справляється з викидами.

Алгоритм K-середніх розбиває заданий набір даних на попередньо визначену (K) кількість кластерів за допомогою певної метрики відстані. Центр кожного кластера/групи називається центроїдом.[10,11]

Алгоритм k-середніх розбиває набір N вибірок X на K непересічних кластерів C, кожен з яких описується середнім значенням μ_i вибірок у кластері.

Середні значення зазвичай називають «центроїдами» кластера; зауважте, що вони, загалом, не є точками з X, хоча вони живуть в одному просторі[9].

Алгоритм K-середніх спрямований на вибір центроїдів, які мінімізують інерцію, або критерій суми квадратів всередині кластерів(1.1).

$$\sum_{i=0}^{n} \min(\|x_i - \mu_i\|^2), \ \ \mu_i \in C \tag{1.1}$$

Переваги[10–12]:

- Просто реалізувати.
- Його можна масштабувати до масивних наборів даних, а також швидше працює для великих наборів даних.
- Він дуже часто адаптується до нових зразків.
- Низька трудомісткість.$O(n)$

Недоліки[10–12]:

- Погано, якщо кластери мають різну форму геометрії.
- Алгоритм K-Means чутливий до викидів.
- Зі збільшенням кількості вимірів масштабованість зменшується.
- Не гарантує досягнення глобального оптимуму.

BIRCH (Balanced iterative reducing and clustering using hierarchies) будує дерево під назвою Clustering Feature Tree (CFT) для заданих даних. Дані, по суті, стискаються з втратами даних до набору вузлів Clustering Feature (CF Nodes). Вузли CF мають кілька підкластерів, які називаються підкластерами функцій кластеризації (субкластери CF), і ці підкластери CF, розташовані в некінцевих вузлах CF, можуть мати вузли CF як дочірні.[9,10,13]

Переваги[10,12]:

- Ефективно витрачає простір пам’яті.
- Стійкий до викидів.
- Низька трудомісткість.$O(n)$
- Підходить для великих наборів даних.

Недоліки[10,12]:

- Не підходить для виявлення кластеру довільної форми.
- Не підходить для масивних наборів даних.

Кластеризація за Mean Shift спрямована на виявлення крапель у плавній щільності зразків. Це алгоритм, заснований на центроїді, який працює шляхом оновлення кандидатів на центроїди як середнє значення точок у певному регіоні. Потім ці кандидати фільтруються на етапі постобробки, щоб усунути майже дублікати, щоб сформувати остаточний набір центроїдів.[9,14]

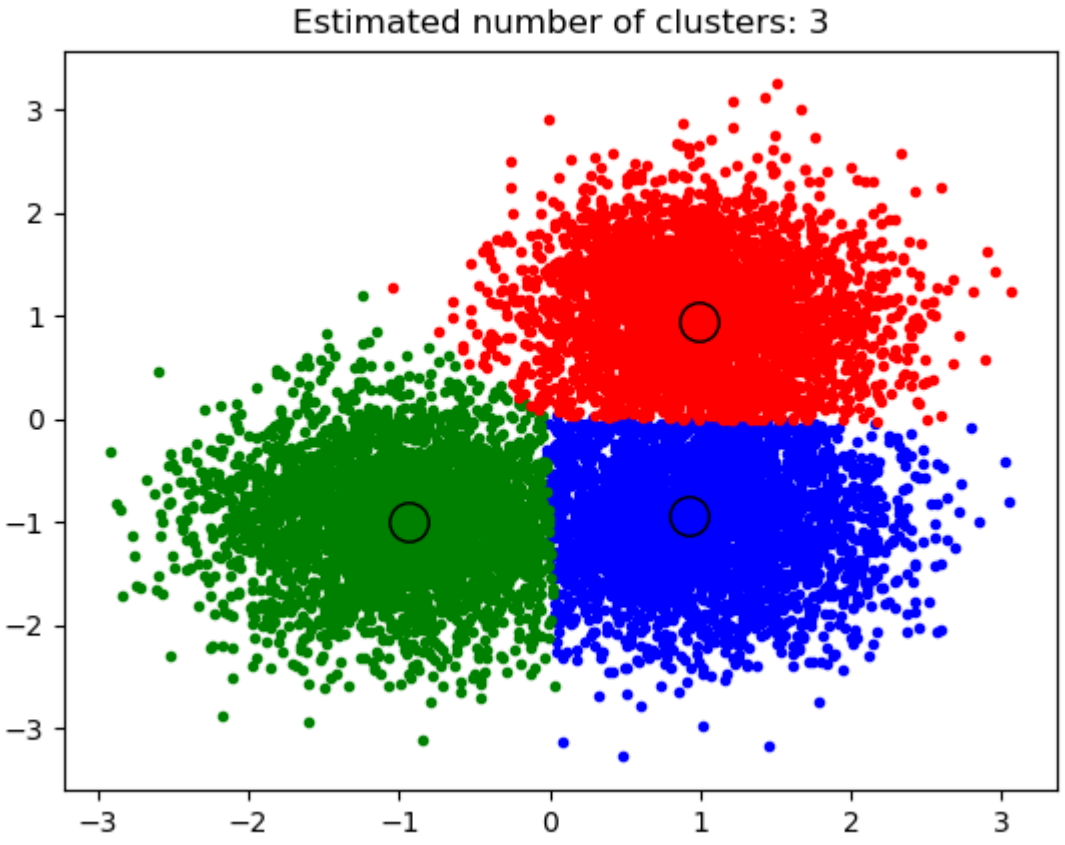

Рисунок 1.8 – Приклад середньої зміни

Враховуючи потенційний центроїд x_i для ітерації t, кандидат оновлюється відповідно до наступного рівняння(1.2).

$$x_i^{t+1} = m(x_i^t) \tag{1.2}$$

Де $N(x_i)$ – околиці зразків у межах заданої відстані навколо x_i та m – вектор середнього зсуву, який обчислюється для кожного центроїда, який вказує на область максимального збільшення щільності точок. Це обчислюється за допомогою наступного рівняння, фактично оновлюючи центроїд як середнє значення зразків у його околицях(1.3).

$$m(x_i) = \frac{\sum_{x_j \in N(x_i)} K(x_j - x_i) x_j}{\sum_{x_j \in N(x_i)} K(x_j - x_i)} \tag{1.3}$$

Переваги[10,11,14]:

- Не потрібно робити жодних припущень моделі, як у K-середніх.
- Він також може моделювати складні кластери, які мають неопуклі форми.
- Для нього потрібен лише один параметр під назвою bandwidth, який автоматично визначає кількість кластерів.
- Немає проблеми локальних мінімумів, як у K-середніх.
- Немає проблем, створених через викиди.

Недоліки[10,11,14]:

- Алгоритм середнього зсуву погано працює у випадку великої розмірності, коли кількість кластерів різко змінюється.
- Ми не маємо прямого контролю над кількістю кластерів, але в деяких програмах нам потрібна певна кількість кластерів.
- Він не може розрізняти значущі та безглузді модуси.

Алгоритм DBSCAN (Density-based spatial clustering of applications with noise) розглядає кластери як області високої щільності, розділені областями низької щільності. Завдяки цьому досить загальному погляду, кластери, знайдені DBSCAN, можуть мати будь-яку форму, на відміну від k-середніх, які припускають, що кластери мають опуклу форму. Центральним компонентом DBSCAN є концепція кернових проб, тобто проб, що знаходяться в зонах високої щільності. Таким чином, кластер – це набір основних зразків, кожен з яких знаходиться близько один до одного (вимірюється деякою мірою відстані), і набір неосновних зразків, які близькі до основного зразка (але самі по собі не є основними зразками).[9,15]

Переваги[10,12]:

- Може виявити кластери довільної форми.
- Знаходить скупчення, оточене іншими скупченнями.
- Надійний до виявлення контурів (шум).

Недоліки[10,12]:

- Чутливий до параметрів кластеризації minPoints і EPS.
- Не вдається визначити кластер, якщо щільність змінюється та якщо набір даних занадто просторий.
- Вибірка впливає на показники щільності.

GMM (Gaussian Mixture Models) припускають, що існує кілька розподілів Гауса, і кожен з них представляє кластер. Таким чином, Гауссова модель має тенденцію групувати разом точки даних, які належать одному розподілу.[9,12,16]

По-перше, ми повинні визначити n_clusters. Оптимальна кількість кластерів - це значення, яке мінімізує інформаційний критерій Акаіке (AIC) або байєсівський інформаційний критерій (BIC).

Переваги[10,12,17]:

- Основні два кроки алгоритму EM, тобто E-крок і M-крок, часто досить прості для багатьох проблем машинного навчання з точки зору реалізації.
- Рішення M-кроків часто існує в закритій формі.
- Завжди гарантовано, що значення ймовірності буде зростати після кожної ітерації.

Недоліки [10,12,17]:

- Має повільну конвергенцію.

- Він чутливий до початкової точки, збігається лише до локального оптимуму.
- Він не може виявити K (ймовірність продовжує зростати з кількістю кластерів)
- Він враховує як прямі, так і зворотні ймовірності.

Для сегментації в цій роботі буде використано алгоритм k-mins, оскільки він є масштабованим і швидким для великих наборів даних і дуже часто адаптується до нових прикладів. У роботі також зроблені деякі оцінки[17]де k-середні показали найкращий результат.

1.3.Постановка задачі

Отже, головним завданням є розробка інформаційної системи, яка аналізуватиме та автоматично розподілятиме клієнтів на сегменти за схожими ознаками на основі їх поведінки.

Нехай надано набір даних деяких сеансів користувача інтернет-магазину, який містить сторінку, де користувач переглянув/купив товар, категорію товару та бренд виробника. Виходячи з цього, нам потрібно сегментувати клієнтів за їхніми вподобаннями за категоріями та брендами.

Потрібно реалізувати наступні кроки:

1. Підготувати дані, які відображатимуть user_id і кількість сторінок, де користувач переглянув/купив товар певного бренду та категорії.
2. Застосувати метод k-середніх для підготовлених даних.
3. Знайти оптимальну кількість сегментів k.
4. Дайте кластерам опис на основі подібності параметрів.

2. ОПИС ТЕХНОЛОГІЇ МАШИННОГО НАВЧАННЯ ДЛЯ СЕГМЕНТАЦІЇ КОРИСТУВАЧІВ ВЕБ-РЕСУРСУ

2.1.Алгоритм кластерізації користувачів інтернет магазину

У постановці задачі алгоритмом кластеризації обрано метод k-середніх.

Алгоритм k-середніх – один з алгоритмів машинного навчання для вирішення задачі кластеризації. Алгоритм k-середніх – неієрархічний, ітераційний метод кластеризації. Задача алгоритму полягає в розбитті вибірки X на визначену кількість k кластерів $S = \{S_1, S_2, \dots, S_k\}$ на основі їх наближеності до середнього значення $\mu_i, i = 1, \dots, k$ у кожному кластері, також називаємого центроїдом [18–21]. Тобто алгоритм полягає в пошуку описаному рівнянням (2.1).

$$\arg\min_S \sum_{i=1}^{k} \sum_{x \in S_i} (\|x - \mu_i\|^2) \qquad (2.1)$$

Вхідні дані методу:

– Набір з n векторів $X = \{x_1, x_2, \dots, x_n\}$.

– Число кластерів $k, k \in N, k \leq n$.

Результатом роботи методу є набір з n цілих чисел $L_1, L_2, \dots, L_n$, де L_k – значення кластеру до якого належить вектор x_k, $k = 1, \dots, n$.

Алгоритм можна описати за допомогою блоксхеми (Рисунок 2.1).

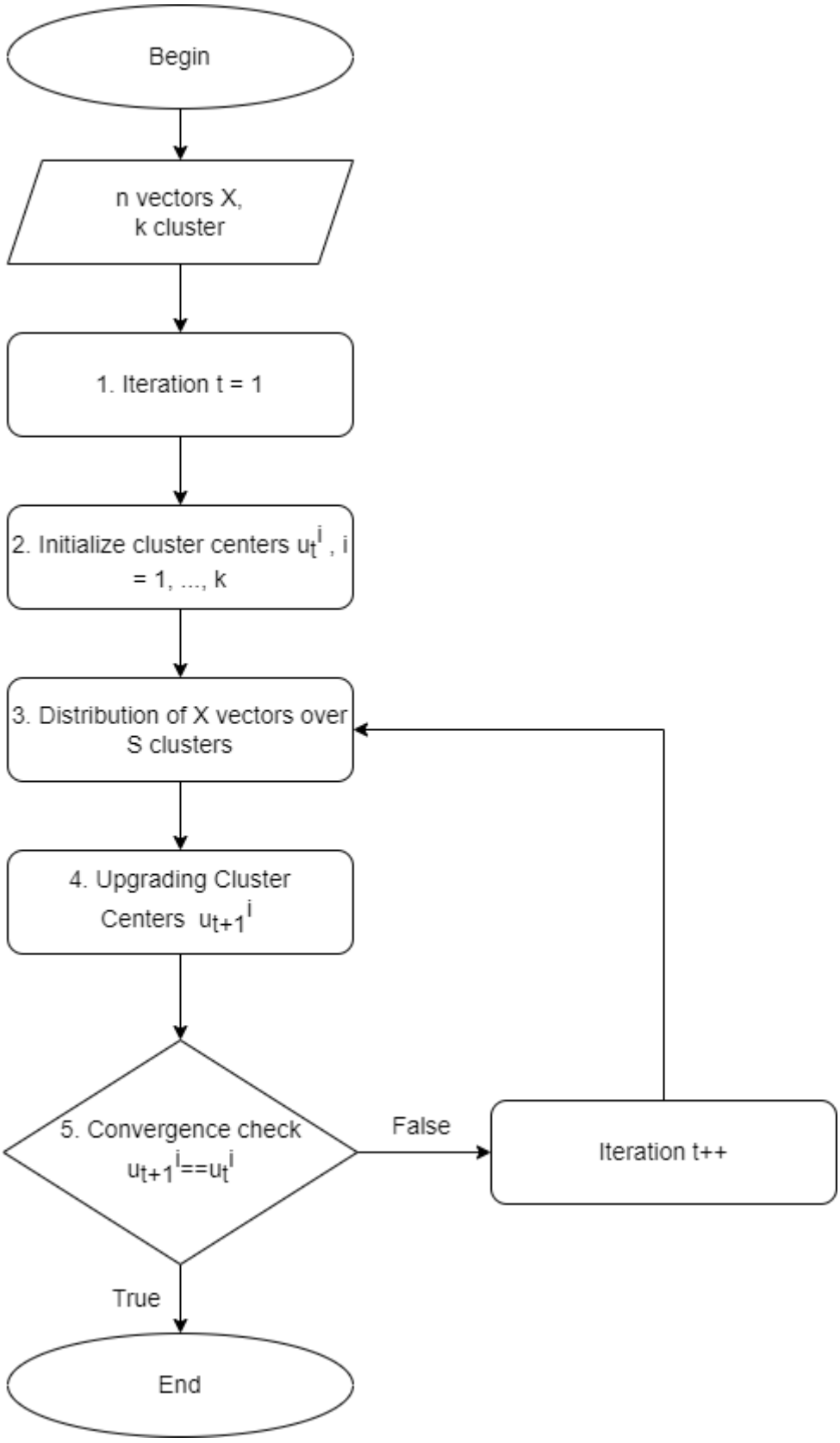

Рисунок 2.1 – Блоксхема алгоритму k-середніх

Перший крок – створюємо змінну $t = 1$ що буде відповідати номеру ітерації методу.

Другий крок – ініціаілзація кластерів[22]. Обираються точки $\mu_i, i = 1, \dots, k$, що будуть розглядатись як початкові центри кластерів на ітерації t. Найбільш розповсюдженими є дві стратегії: метод Forgy, в якому точки обираються повністю випадково, та метод випадкового розподілення (Random Partitioning), в якому кожному вектору $x_i \in X, i = 1, \dots, n$ випадково обирається кластер $S_1, \dots, S_k$, після чого для кожного отриманого кластера вираховується середнє значення $\mu_1, \dots, \mu_k$.

Третій крок – розподілення векторів по кластерам (2.2).

$$\forall x_i \in X, i = 1, \dots, n : x_i \in S_j \leftrightarrow j = \arg min_k \|x - \mu_i\|^2 \quad (2.2)$$

Даний крок можна представити у вигляді графа (Рисунок 2.2). Кожна пара веторів $x_i, i = 1, \dots, n$ та центрів кластерів $\mu_i, i = 1, \dots, k : (x_i, \mu_i)$ потрапляють на вузол d на якому обчислюється відстань між ними. Далі кожен вузол d зв'язаний з певним вузлом x_i передається на один вузол m, де вектору x_i присвоюється нове значення кластера $L_1, \dots, L_n$, такі що відповідають формулі (2.3).

$$\forall x_i, i = 1, \dots, n, x_i \in S_j \leftrightarrow L_i = j \quad (2.3)$$

Обчислення відстані d також можна представити у вигляді графа (Рисунок 2.3) де x_{iz}, μ_{iz} – координати ветора та центра кластра відповідно.

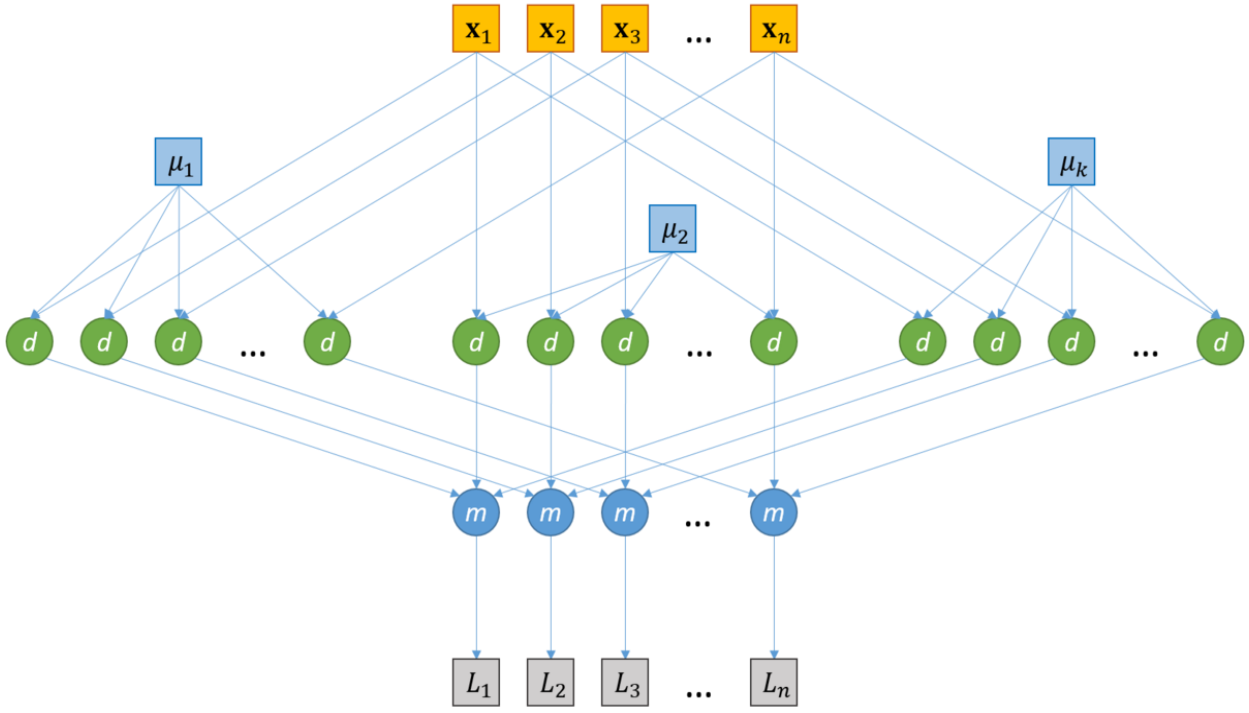

Рисунок 2.2 – Схема розподілення векторів по кластерам

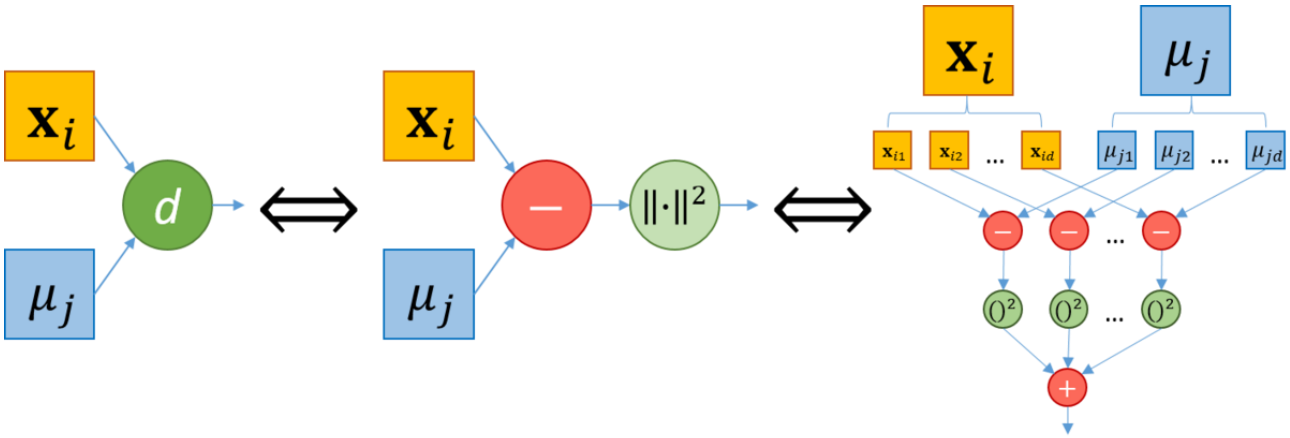

Рисунок 2.3 – Схема обчислення відстані між центром кластера та вектором

Четвертий крок – перерахунок центрів кластерів (2.4).

$$\forall i = 1, \dots, k : \ \mu_i^{t+1} = \frac{1}{|S_i|} \sum_{x \in S_i} x \qquad (2.4)$$

Даний крок можна представити у вигляді графа (Рисунок 2.4). Вихідними вузлами даного графа є вектори $x_i, i = 1, \dots, n$ та значення кластера $L_1, \dots, L_n$. Всі вектори x_i подаються в вузли +1,...,+k, кожен вузол +m, m=1,...,k, відповідає

операції додавання векторів кластера з номером m. Мітки кластера $L_1, \dots, L_n$ також передаються на вузли S_m, $m = \mathbf{1}, \dots, k$, де обчислюється кількість векторів кожного класу. Далі сума отримана на вузлах +m ділиться на кількість отриману на вузлах S_m і таким чином знаходиться новий центр кожного класу μ_m.

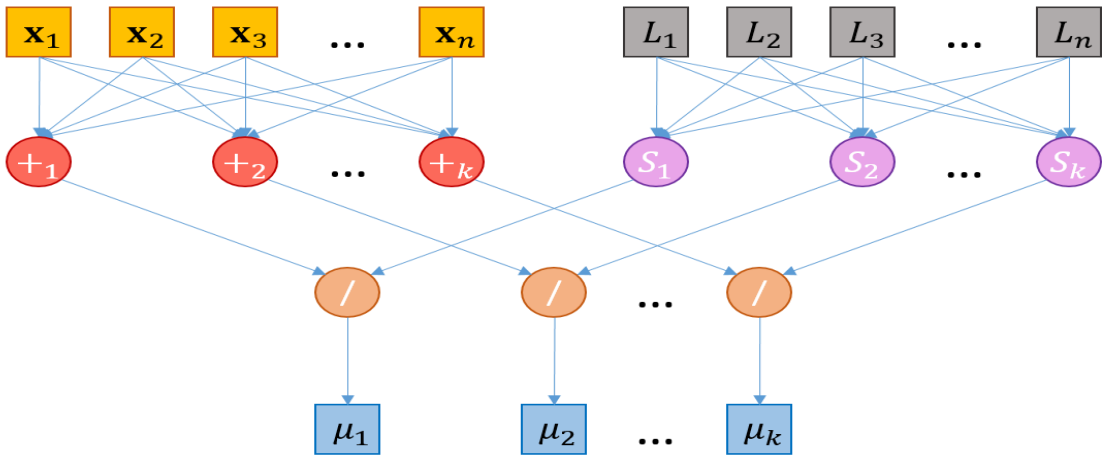

Рисунок 2.4 – Схема перерахунку центрів кластерів

П’ятий крок – перевірка збіжності. Центри класів для наступної ітерації повинні збігатись з центрами поточної ітерації (2.5). Якщо умова виконана, то роботу алгоритму завершено, якщо ні то збільшуємо номер ітерації $t = t + 1$ та повертаємось до кроку номер три.

$$\exists i \in \overline{1,k} : \mu_i^t \equiv \mu_i^{t+1} \qquad (2.5)$$

Розрахуємо складність виконання алгоритму[23].

Нехай $\Theta_{centroid}^{d,m}$ – складність обчислення центроїду кластера, число елементів якого дорівнює m, в d-мірному просторі. Складність обчислення рахується за формулою (2.6).

$$\Theta_{centroid}^{d,m} = m \cdot d \text{ додавань} + d \text{ ділень} \qquad (2.6)$$

$\Theta^{d}_{distance}$ – складність обчислення відстані між двома d-мірними векторами. Складність обчислення рахується за формулою (2.7).

$$\Theta^{d}_{distance} = d \text{ віднімань} + d \text{ множень } + (d-1) \text{ додавань} \quad (2.7)$$

Складність кроку ініціалізації k кластерів з m кількістю елементів в d-мірному просторі $\Theta^{k,d,m}_{init}$ для стратегії Forgy обчислюється за формулою (2.8), для стратегії випадкового розподілення – за формулою (2.9).

$$\Theta^{k,d,m}_{init} = 0 \quad (2.8)$$

$$\Theta^{k,d,m}_{init} = k \cdot \Theta^{d,m}_{centroid} \quad (2.9)$$

Складність кроку розподілення d-мірних векторів по k кластерам $\Theta^{k,d}_{distribute}$ обчислюється за формулою (2.10).

$$\Theta^{k,d}_{distribute} = \Theta^{d}_{distance} \cdot n \cdot k \quad (2.10)$$

Складність перерахунку центроїдів k кластерів з m кількістю елементів в d-мірному просторі $\Theta^{k,d,m}_{recenter}$ облчислюється за формулою (2.11).

$$\Theta^{k,d,m}_{recenter} = k \cdot \Theta^{d,m}_{centroid} \quad (2.11)$$

Нехай алгоритм зійшовся на ітерації i. Тоді тимчасова складність алгоритму $\Theta^{d,n}_{k-means}$ обчислюється за формулою (2.12).

$$\Theta^{d,n}_{k-means} \leq \Theta^{k,d,m}_{init} + i(\Theta^{k,d}_{distribute} + \Theta^{k,d,m}_{recenter}) \quad (2.12)$$

Спростивши формулу (2.12) отримаємо (2.13).

$$\Theta^{d,n}_{k-means} \sim O(ikdn) \quad (2.13)$$

2.2.Оптимізація алгоритму кластеризації

Одним з основних недоліків методу k-середніх є те, що кількість кластерів є вхідним параметром методу, тобто не обирається автоматично в процессі роботи алгоритму. Одже необхідно оптимізувати вибір даного параметру.

Одним із найпопулярніших методів оптимізації кількості кластерів є метод локтя. Ідея методу полягає в послідовному виконанні методу k-середніх для різної кількості класерів k (наприклад від 1 до 10) [24–26]. Для кожної кількості кластерів k необхідно обчислити суму квадратів похибки (SSE) (2.14) де μ_i – центр кластера i, x – вектор, що належить кластеру S_i.

$$SEE_k = \sum_{i=1}^{k} \sum_{x \in S_i} (\|x - \mu_i\|^2) \qquad (2.14)$$

Наступний крок – побудувати графік залежності SSE від кількості кластерів k (Рисунок 2.5). Зі збільшенням кількості кластерів k SSE буде зменшуватись до 0 (досягне 0 при $k = n$, де n – кількість вхідних векторів). Необхідно обрати таке найменше k при якому значення SSE буде достатньо мале та після якого тенденція зниження SSE значно сповільниться. На графіку це відбувається в точці названій “локтем”.

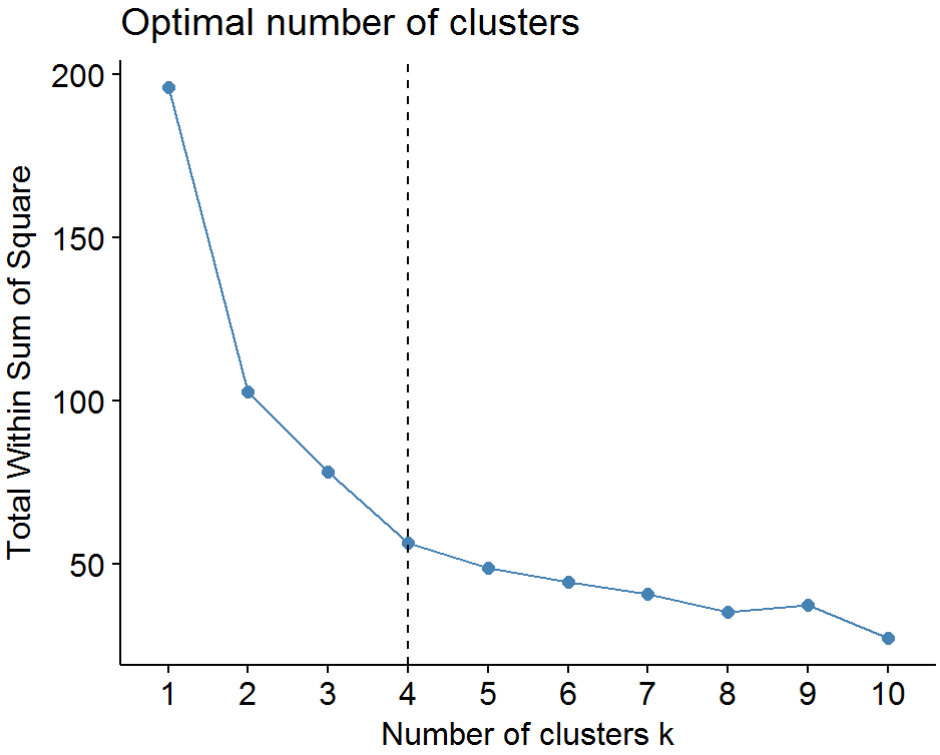

Рисунок 2.5 – Приклад методу локтя

Іншим методом є метод силуета. Для кожного знайденого кластера k потрібно обчилсити ширину силуету (2.15) [27–30].

$$Sil_k = \frac{b(x_i, c_k) - a(x_i, c_k)}{min[b(x_i, c_k), a(x_i, c_k)]} \quad (2.15)$$

$a(x_i, c_k)$ – середня відстань від $x_i \in c_k$ до інших об’єктів кластеру c_k (2.16). $b(x_i, c_k)$ – середня відстань від $x_i \in c_k$ до інших об’єктів іншого кластеру $c_l, l \neq k$ (2.17).

$$a(x_i, c_k) = \frac{1}{|c_k|} \sum_{x_j \in c_k} \|x_i - x_j\| \quad (2.16)$$

$$b(x_i, c_k) = min_{c_l \in C \setminus c_k} \{ \frac{1}{|c_l|} \sum_{x_j \in c_l} \|x_i - x_j\| \} \quad (2.17)$$

На основі отриманих значень можна побудувати діаграму силуетів для заданої кількості кластерів (Рисунок 2.6).

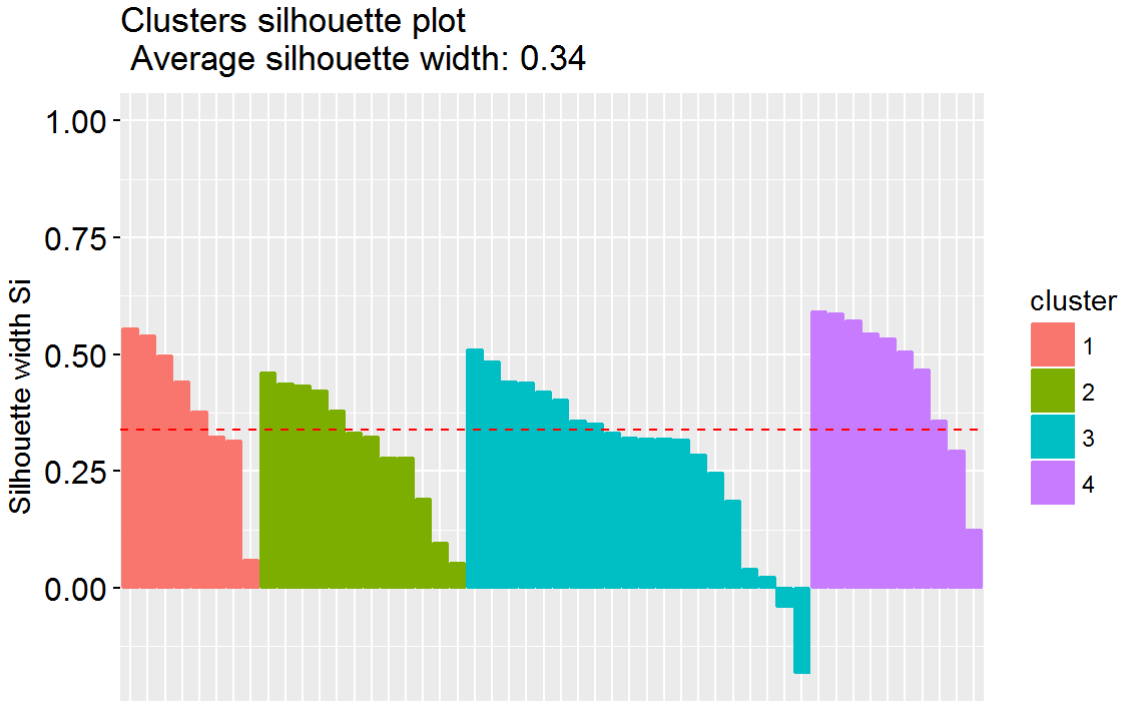

Рисунок 2.6 – Приклад діаграми силуетів

Загальне середнє з отриманих значень і буде визначати якість кластеризації (2.18).

$$Sil = \frac{1}{n}\sum_{k\leq n}^{k=1} Sil(k) \qquad (2.18)$$

Слід зауважити, що $-1 \leq Sil \leq 1$ і чим блище оцінка до 1 тим краще.

Таким чином для пошуку оптимального значення кількості кластерів k необхідно виконати метод k-найменших для певного діапазону k та знайти для якого значення k результат методу силуетів буде найбільший (Рисунок 2.1).

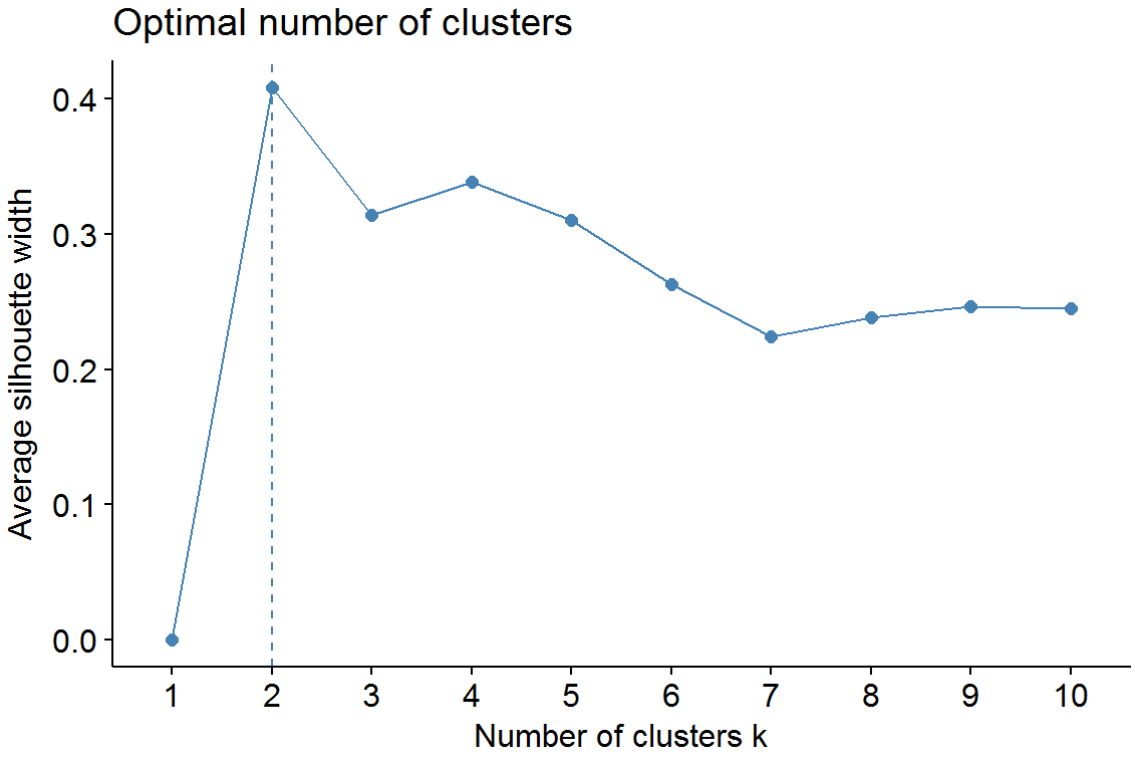

Рисунок 2.7 – Приклад знаходження оптимальної кількості кластерів методом силуету

У даній роботі буде використано обидва методи для порівняння результатів та вибору кращого значення кількості кластерів.

3. РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ СЕГМЕНТАЦІЇ КОРИСТУВАЧІВ ВЕБ-РЕСУРСУ

3.1. Опис програмної реалізації інформаційної технології сегментації користувачів веб-ресурсу

Інформаційна технологія приймає на вхід файл з даними про користувацькі сессії та на основі них присвоює користувачу значення певного сегменту. У запропонованому варіанті аналіз відбувається в декілька етапів, зв'язок між якими показано на функціональній схемі (Рисунок 3.1).

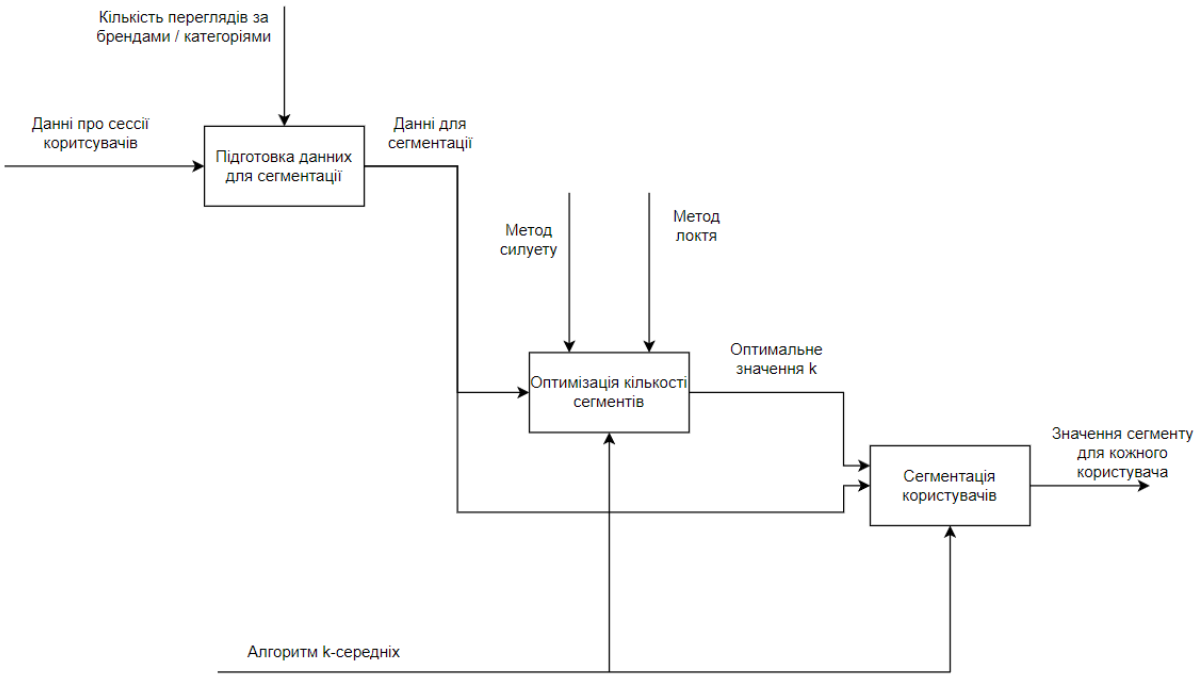

Рисунок 3.1 – Функціональна схема сегментації користувачів веб-магазину

На Рисунок 3.1 показано, що спочатку данні перетворюють таким чином що в наборі показана інформація про те яку кількість сторінок відвідав користувач за категоріями та брендами. Потім відбувається оптимізація вхідного параметру кількості сегментів (кластерів) для виконання алгоритму k-середніх за допомогою методу силуету та локтя. Останній крок – сегментація користувачів за допомогою алгоритму k-середніх на основі сформованого в

першому кроці наборі даних та оптимального значення кількості сегментів отриманого в другому кроці.

Jupyter Notebook та Visual Studio Code використовувалось як середеовище для розробки. Jupyter Notebook – це інтерактивний інструмент, що дозволяє виконувати окремі блоки коду та одразу бачити результат, який зберігається разом з кодом у файлі проекту, отже для перегляду результатів немає необхідності в повторному виконанні ресурсозатратних блоків коду. Крім того Jupyter дозволяє створювати комірки з маркерованим текстом, що дозволяє зрозуміло описати дії, які виконуються в наступному блоці коду. Розробка здійснювалась на мові програмування Python 3. Розроблене програмне забезпечення потребує використання сторонніх бібліотек з використанням менеджеру pip, опис яких наведено в. Повний програмний код представлений в додатку А.

Таблиця 3.1 – Опис використаних бібліотек

Назва бібліотеки	Опис
NumPy	NumPy — це основний пакет для наукових обчислень на Python [31]. Це бібліотека Python, яка надає багатовимірні масиви, різні похідні об’єкти (такі як замасковані масиви та матриці), а також набір процедур для швидких операцій над масивами, включаючи математичні, логічні, маніпуляції формою, сортування, вибір, введення/виведення. , дискретне перетворення Фур'є, базова лінійна алгебра, основні статистичні операції, випадкове моделювання та багато іншого.
Pandas	Pandas — це бібліотека з відкритим кодом, яка надає високопродуктивні, прості у використанні структури даних і інструменти аналізу даних для мови програмування Python [32].

Продовження таблиці 3.2

Назва бібліотеки	Опис
Matplotlib	Matplotlib — це комплексна бібліотека для створення статичних, анімованих та інтерактивних візуалізацій на Python в двовимірному та тривимірному просторі.
Scikit-Learn	Scikit-learn — це безкоштовна програмна бібліотека машинного навчання для мови програмування Python, яка надає функціональність для створення та тренування різноманітних алгоритмів класифікації, регресії та кластеризації, таких як лінійна регресія, random forest та градієнтний бустинг.[33]

3.2.Формування данних для аналізу

Для ілюстрації роботи методу візьмемо дані з сайту Kaggle https://www.kaggle.com/datasets/mkechinov/ecommerce-behavior-data-from-multi-category-store. Після завантаження дані мають наступний вигляд (Рисунок 3.2). Всього набір містить один мільйон записів.

	event_time	event_type	product_id	category_id	category_code	brand	price	user_id	user_session
0	2019-11-01 00:00:00 UTC	view	1003461	2053013555631882655	electronics.smartphone	xiaomi	489.07	520088904	4d3b30da-a5e4-49df-b1a8-ba5943f1dd33
1	2019-11-01 00:00:00 UTC	view	5000088	2053013566100866035	appliances.sewing_machine	janome	293.65	530496790	8e5f4f83-366c-4f70-860e-ca7417414283
2	2019-11-01 00:00:01 UTC	view	17302664	2053013553853497655	NaN	creed	28.31	561587266	755422e7-9040-477b-9bd2-6a6e8fd97387
3	2019-11-01 00:00:01 UTC	view	3601530	2053013563810775923	appliances.kitchen.washer	lg	712.87	518085591	3bfb58cd-7892-48cc-8020-2f17e6de6e7f
4	2019-11-01 00:00:01 UTC	view	1004775	2053013555631882655	electronics.smartphone	xiaomi	183.27	558856683	313628f1-68b8-460d-84f6-cec7a8796ef2

Рисунок 3.2 – Данні з сайту Kaggle

Далі необхідно видалити стовпці, що не будуть брати участі в аналізі, а саме: «event_time», «product_id», «category_id», «price», «user_session» (Рисунок 3.3).

	event_type	category_code	brand	user_id
0	view	electronics.smartphone	xiaomi	520088904
1	view	appliances.sewing_machine	janome	530496790
2	view	NaN	creed	561587266
3	view	appliances.kitchen.washer	lg	518085591
4	view	electronics.smartphone	xiaomi	558856683

Рисунок 3.3 – Данні, що беруть участь в аналізі

Як видно з Рисунок 3.3 категорія товару має також підкатегорію. Для аналізу ідеально було б залишити це як є, але виходячи з обмеженості апаратних ресурсів далі будемо використовувати лише назву категорії (Рисунок 3.4).

	event_type	category_code	brand	user_id
0	view	electronics	xiaomi	520088904
1	view	appliances	janome	530496790
2	view	NaN	creed	561587266
3	view	appliances	lg	518085591
4	view	electronics	xiaomi	558856683

Рисунок 3.4 – Данні, що беруть участь в аналізі без підкатегорій

Виведемо топ-10 категорій за популярністю (Рисунок 3.5). Як ми бачимо одні категорії користуються значно більшим попитом за інші. Залишимо перші 10 категорій та інші перейминуемо в “others”.

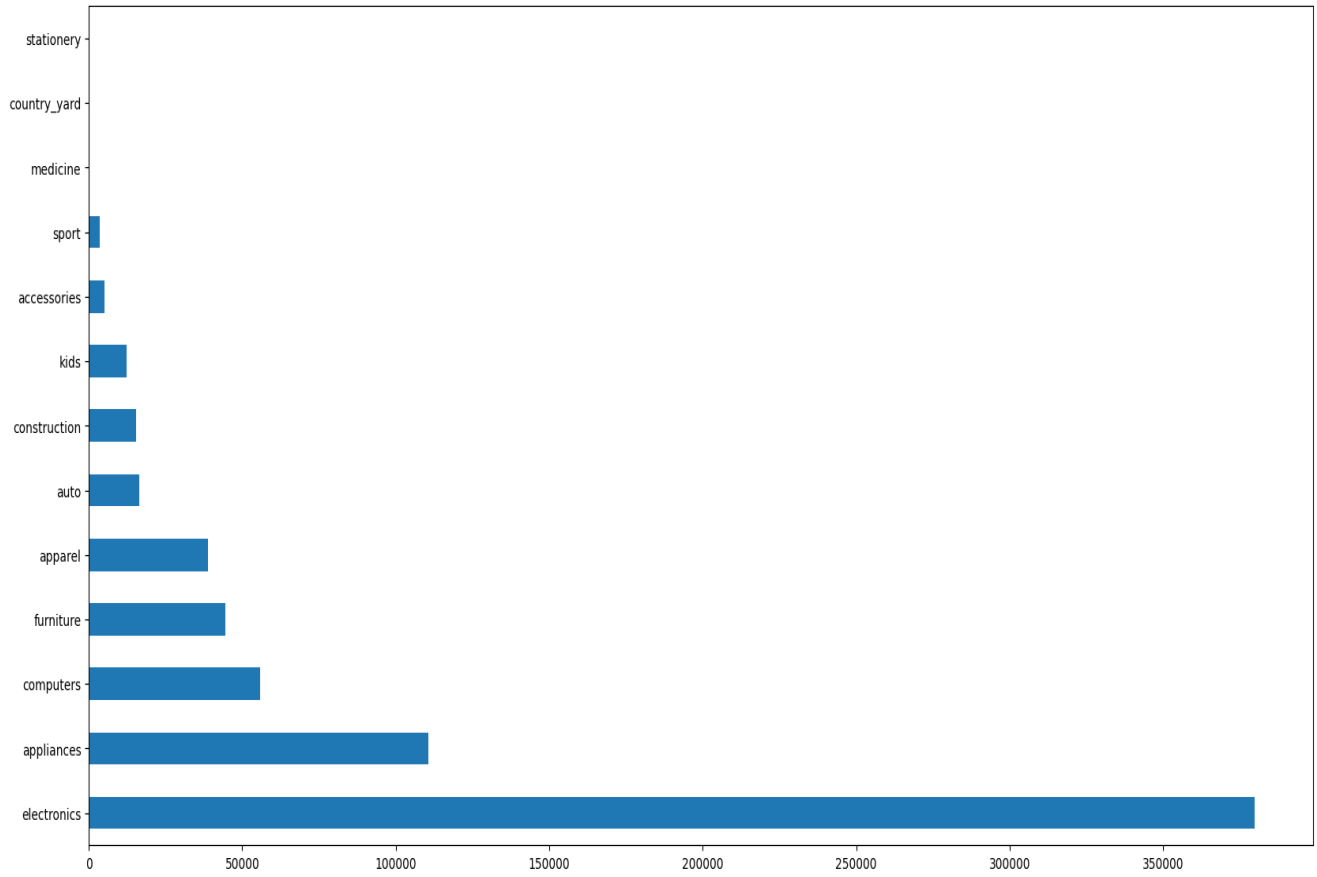

Рисунок 3.5 – Популярність категорій

На основі отриманих данних створимо новий набір, що містить інформацію про переглянуті сторінки певної категорії певним користувачем (Рисунок 3.6). На рисунку вмістились не всі колонки даного набору, їх всього одинадцять: category_count_accessories, category_count_apparel, category_count_appliances, category_count_auto, category_count_computers, category_count_construction, category_count_electronics, category_count_furniture, category_count_kids, category_count_other, category_count_sport.

user_id	category_count_accessories	category_count_apparel	category_count_appliances	category_count_auto	category_count_computers	category_count_cons
274969076	0.0	0.0	0.0	0.0	0.0	
275256741	0.0	0.0	0.0	0.0	1.0	
295643776	0.0	0.0	0.0	0.0	0.0	
296465302	0.0	0.0	0.0	0.0	0.0	
319315209	0.0	0.0	0.0	0.0	0.0	

Рисунок 3.6 – Набір даних про перегляди за категоріями

Виведемо топ-10 брендів (Рисунок 3.7). Розрив між найбільш популярними та найменш популярними брендами набагато менший за аналогічний розрив між категоріями та все ще достатньо великий. Залишимо перші 10 брендів та інші перейминуемо в “others”.

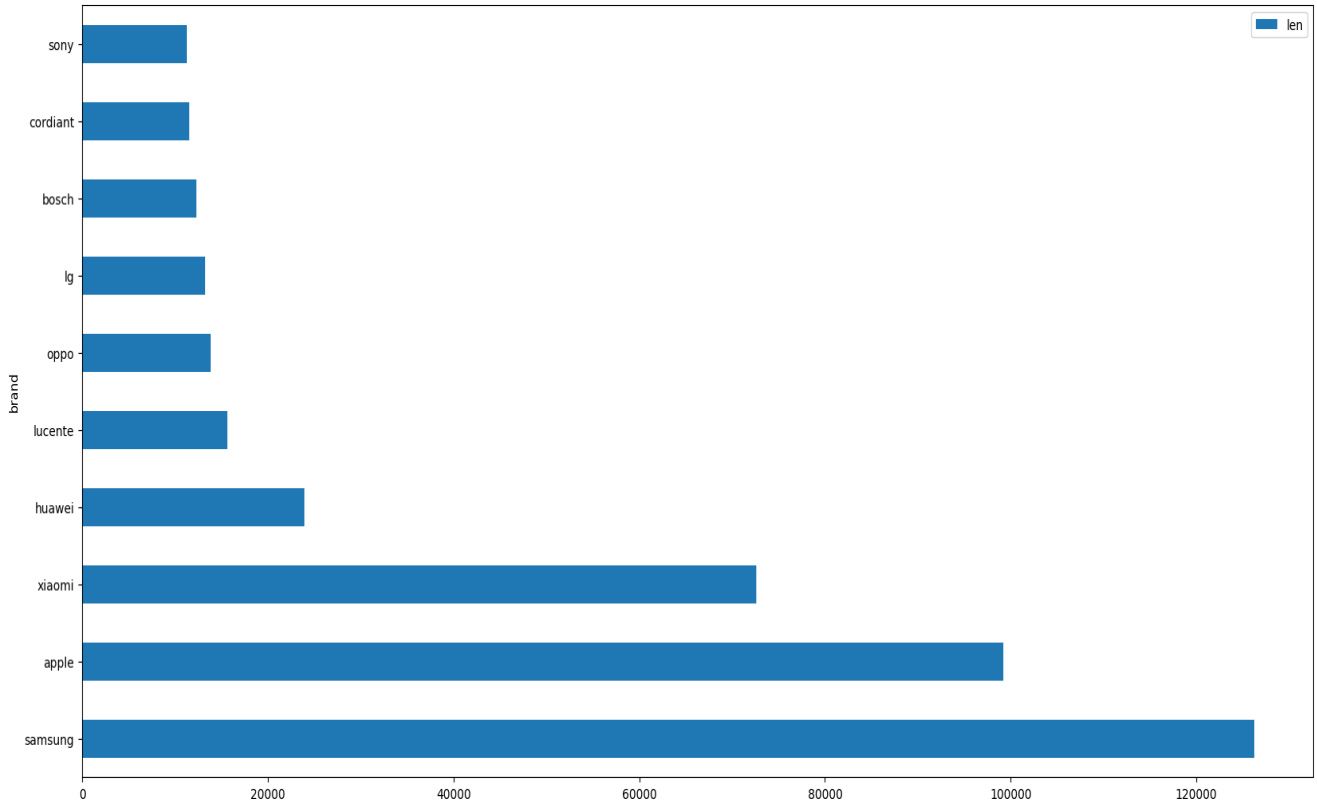

Рисунок 3.7 – Популярність брендів

На основі отриманих данних створимо новий набір, що містить інформацію про переглянуті сторінки з товаром кожного з брендів певним користувачем (Рисунок 3.8). На рисунку вмістились не всі колонки даного набору, їх всього одинадцять: brand_count_apple, brand_count_bosch, brand_count_cordiant, brand_count_huawei, brand_count_lg, brand_count_lucente, brand_count_oppo, brand_count_other, brand_count_samsung, brand_count_sony, brand_count_xiaomi.

user_id	brand_count_apple	brand_count_bosch	brand_count_cordiant	brand_count_huawei	brand_count_lg	brand_count_lucente	brand_count_oppo	b
274969076	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
275256741	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
295643776	1.0	0.0	0.0	0.0	0.0	0.0	0.0	
296465302	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
319315209	0.0	0.0	0.0	0.0	0.0	0.0	0.0	

Рисунок 3.8 – Набір данних про перегляди за брендами

Об’єднаємо набори про перегляди за брендами та за категоріями в один набір по їх індексу – id користувача (Рисунок 3.9). Отриманий набір містить 170337 записи. Данний набір будемо використовувати для сегментації.

user_id	category_count_accessories	category_count_apparel	category_count_appliances	category_count_auto	...	brand_count_other	brand_count_samsung	brand_count_sony	brand_count_xiaomi
274969076	0.0	0.0	0.0	0.0	...	0.0	0.0	3.0	0.0
275256741	0.0	0.0	0.0	0.0	...	1.0	0.0	0.0	0.0
295643776	0.0	0.0	0.0	0.0	...	7.0	0.0	0.0	0.0
296465302	0.0	0.0	0.0	0.0	...	25.0	0.0	0.0	0.0
319315209	0.0	0.0	0.0	0.0	...	3.0	0.0	0.0	0.0

5 rows × 22 columns

Рисунок 3.9 – Зведений набір даних

3.3. Оптимізація кількості кластерів

Для оптимізації кількості кластерів було обрано проміжок від 3 до 30 та використано два методи: локтя (Рисунок 3.10) та силуету (Рисунок 3.11).

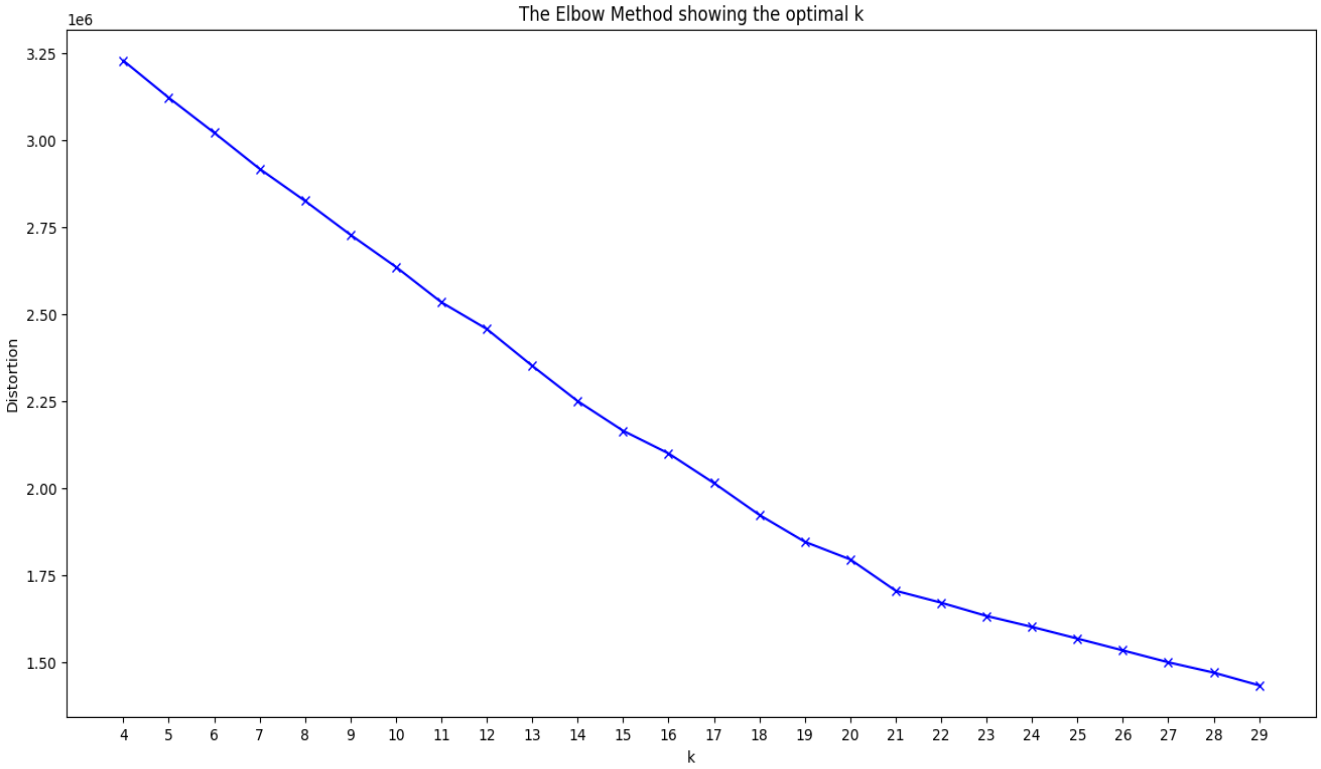

Рисунок 3.10 – Оптимізація методом локтя

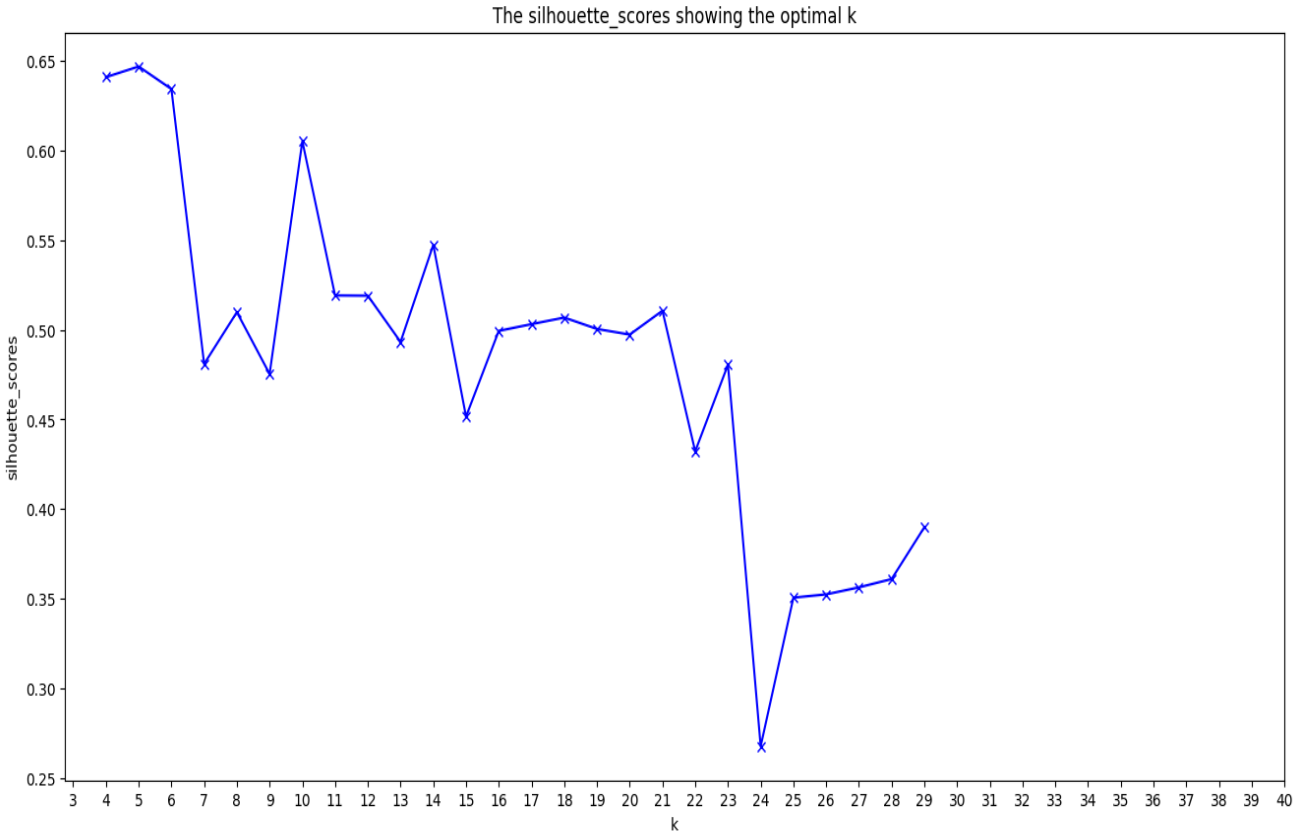

Рисунок 3.11 – Оптимізація методом силуету

Функція описана методом локтя (Рисунок 3.10) рівномірно спадає на всьому проміжку, тому складно виділити «лікоть». Суб'єктивно спадання сповільнюється в точці 21, але недостатньо ади вважати дану точку оптимальною кількістю кластерів.

Побудова графіку для методу силуету (Рисунок 3.11) зайняла значно більше часу (приблизно 3 години) але натомість графік значно більш ілюстративний. З нього чітко видно що значення в точці 5 – найбільше.

Отже значення кількості кластерів k=5 вважаємо оптимальним.

3.4. Аналіз результатів машинного навчання

У результаті роботи алгоритму k-найменших вдалось розприділити 170337 користувачів на 5 кластерів. Представимо розподіл у вигляді кругової діаграми (Рисунок 3.12). Як ми бачимо кластери розподілені нерівномірно і один з них значно переважає над іншими.

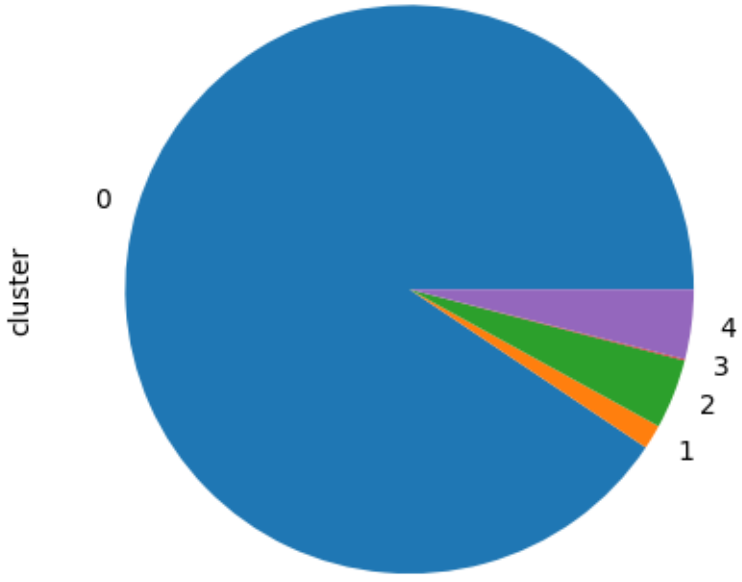

Рисунок 3.12 – Розподіл кластерів

Спробуємо проаналізувати вподобання коритсувачів, що відносяться до певного кластеру.

Побудуємо графіки середнього (Рисунок 3.13), максимального (Рисунок 3.14) та мінімального (Рисунок 3.15) значення кількості переглядів сторінок за вибраними категоріями. З графіків можна зробити висновок, що користувачі, що відносяться до кластеру 0 – користувачі, що нечасто користуються сайтом та вподобання яких складно визначити, 1 – надають перевагу товарам в категорії побутова техніка, 2 – електроніка, 3 – спорт, 4 – товарам в інших категоріях.

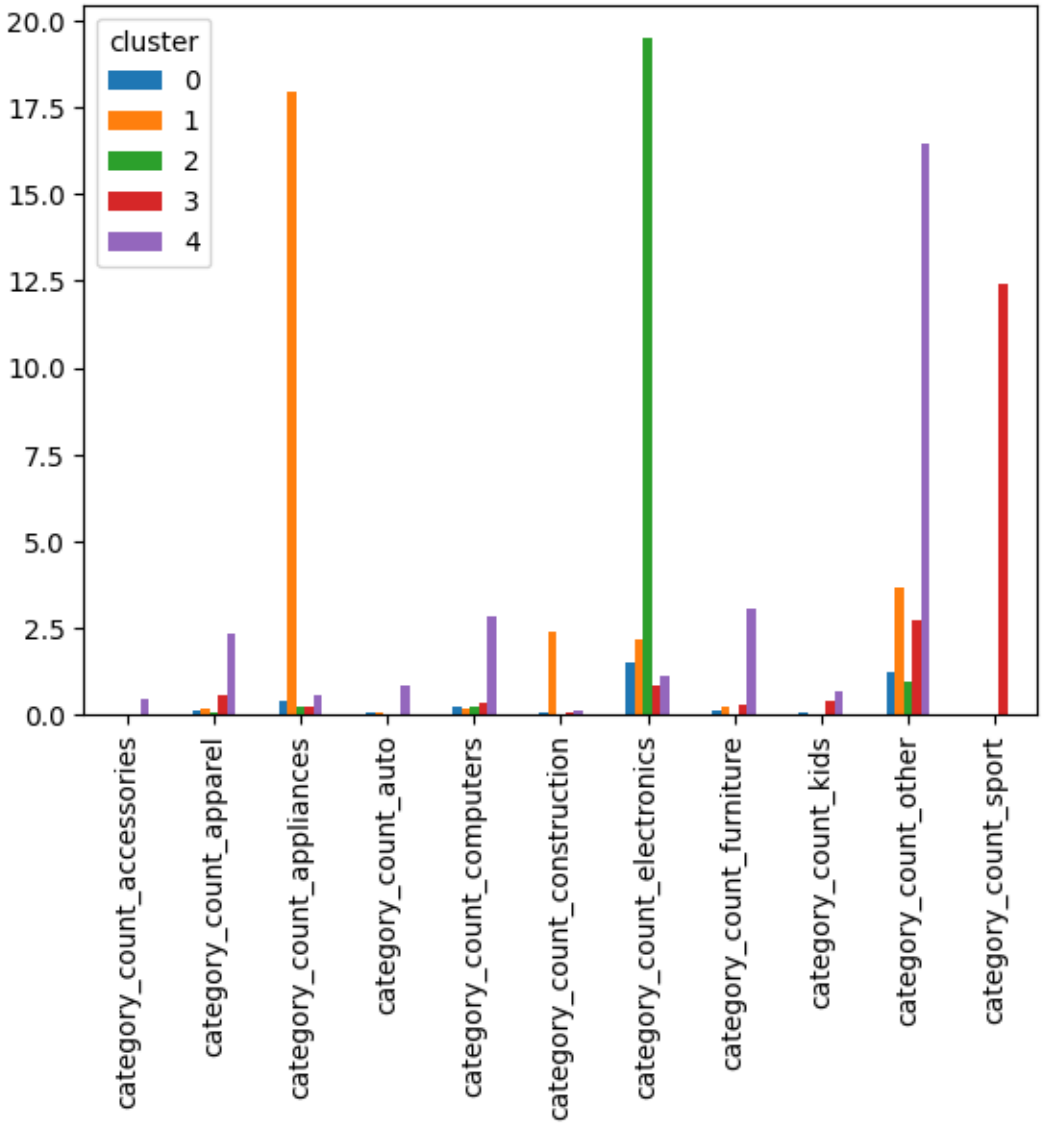

Рисунок 3.13 – Середнє значення кількості преглядів за категоріями

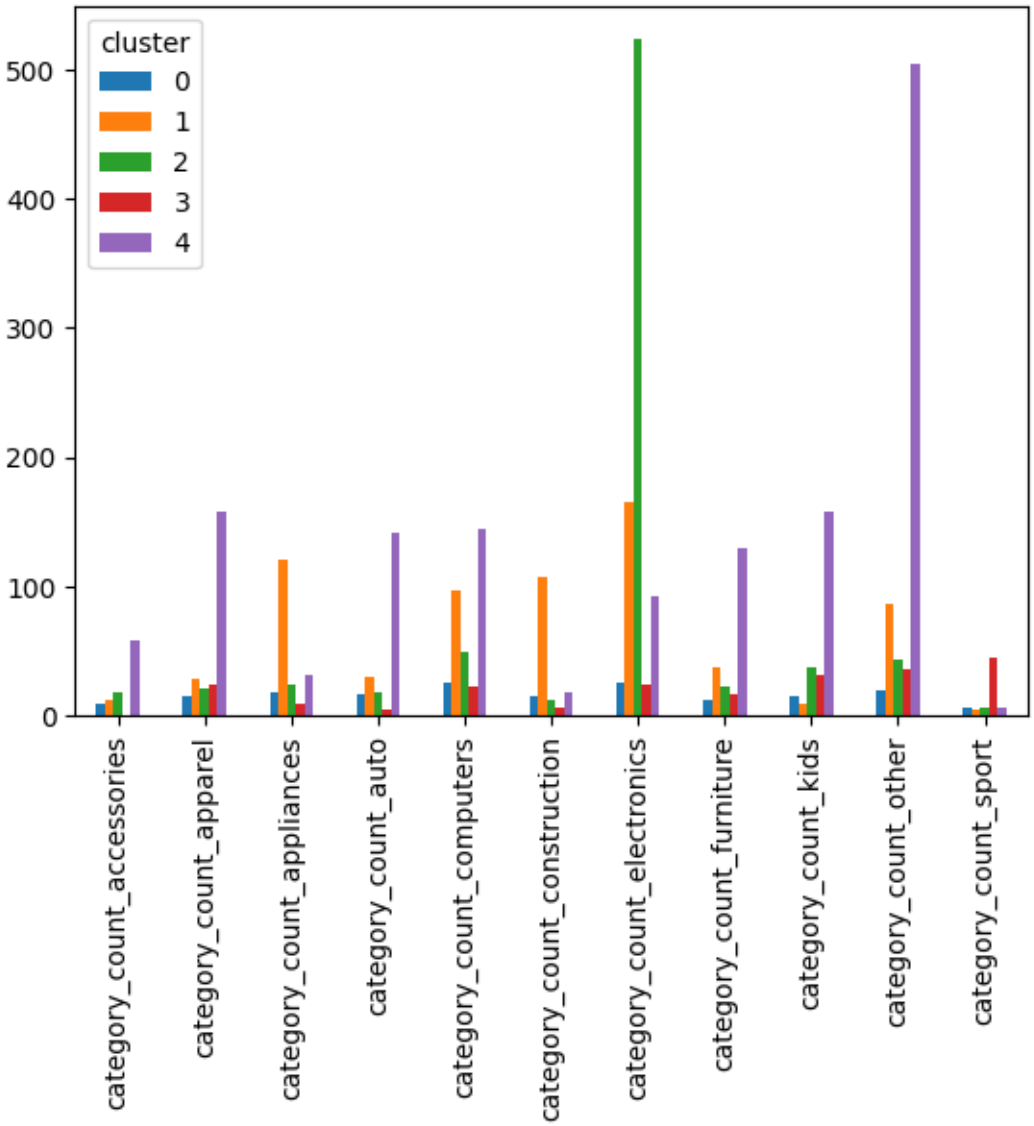

Рисунок 3.14 – Максимальне значення переглядів за категоріями

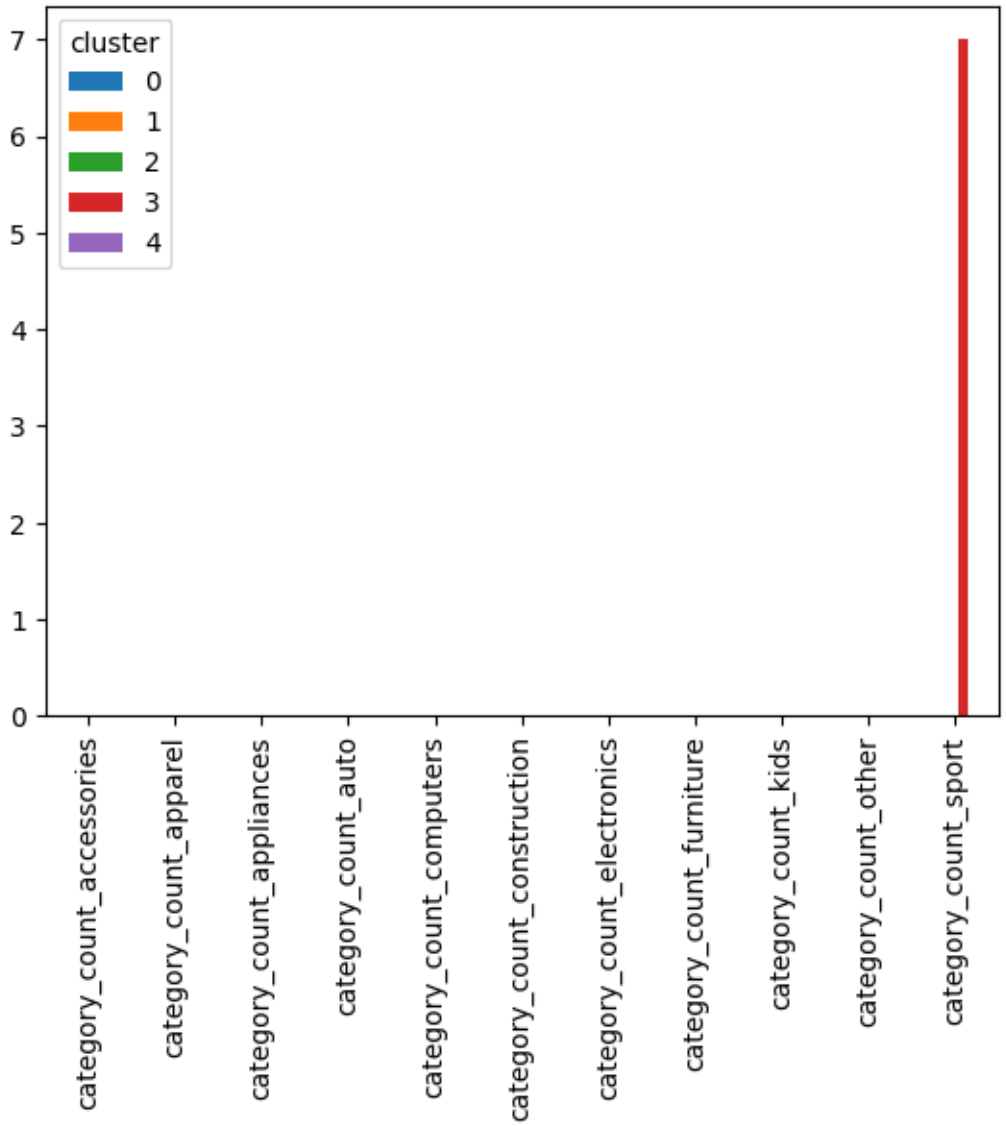

Рисунок 3.15 – Мінімальне занчення переглядів за категоріями

Побудуємо графіки середнього (Рисунок 3.16), максимального (Рисунок 3.17) та мінімального (Рисунок 3.18) значення кількості переглядів сторінок за вибраними брендами. З графіків видно, що користувачі більшості кластерів надають перевагу менш популярним брендам. Виключення становлять користувачі з кластеру 2 (що надають перевагу категорії електроніка), які також цікавляться брендами Apple, Huawei, Samsung, Xiaomi, та користувачі кластеру 1 (що надають перевагу категорії побутова техніка), які також цікавляться брендами Bosch, LG та Samsung.

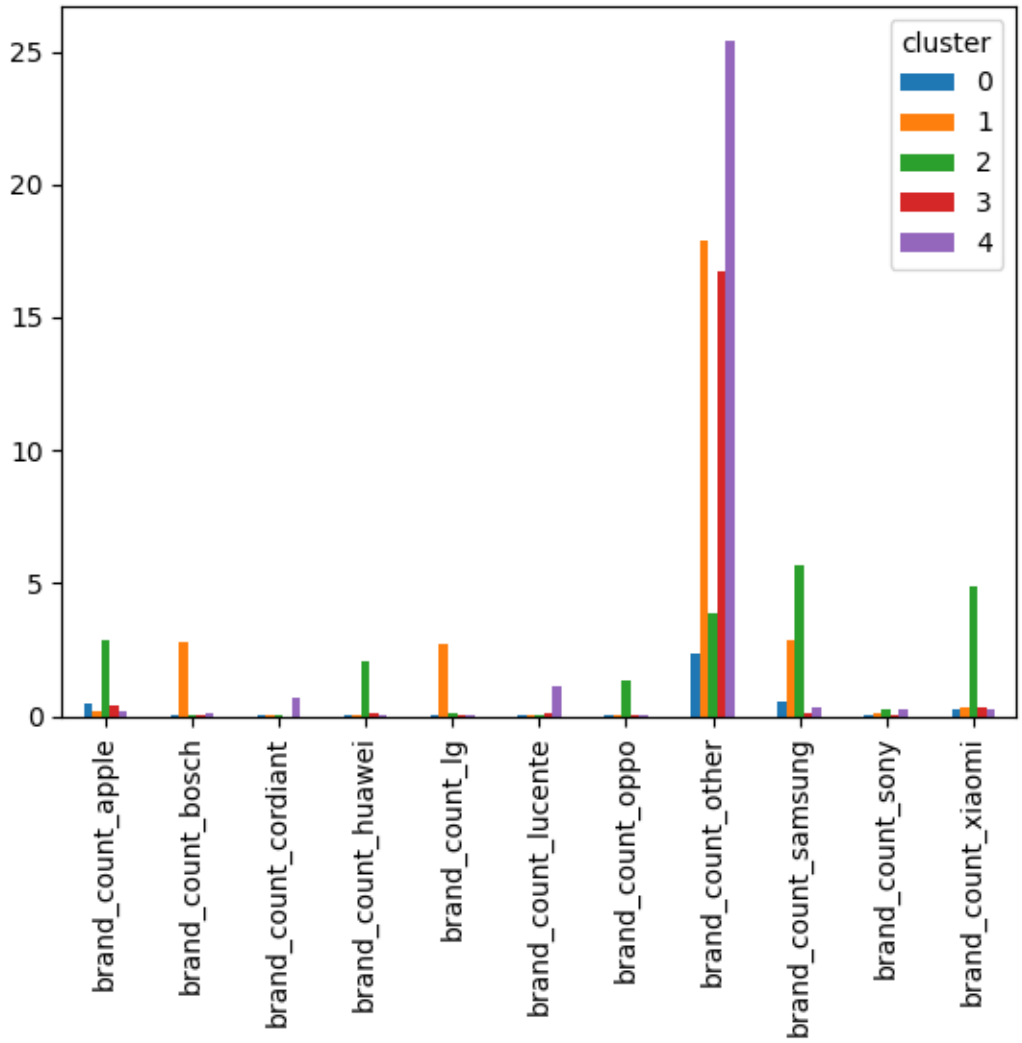

Рисунок 3.16 – Середнє значення кількості преглядів за брендами

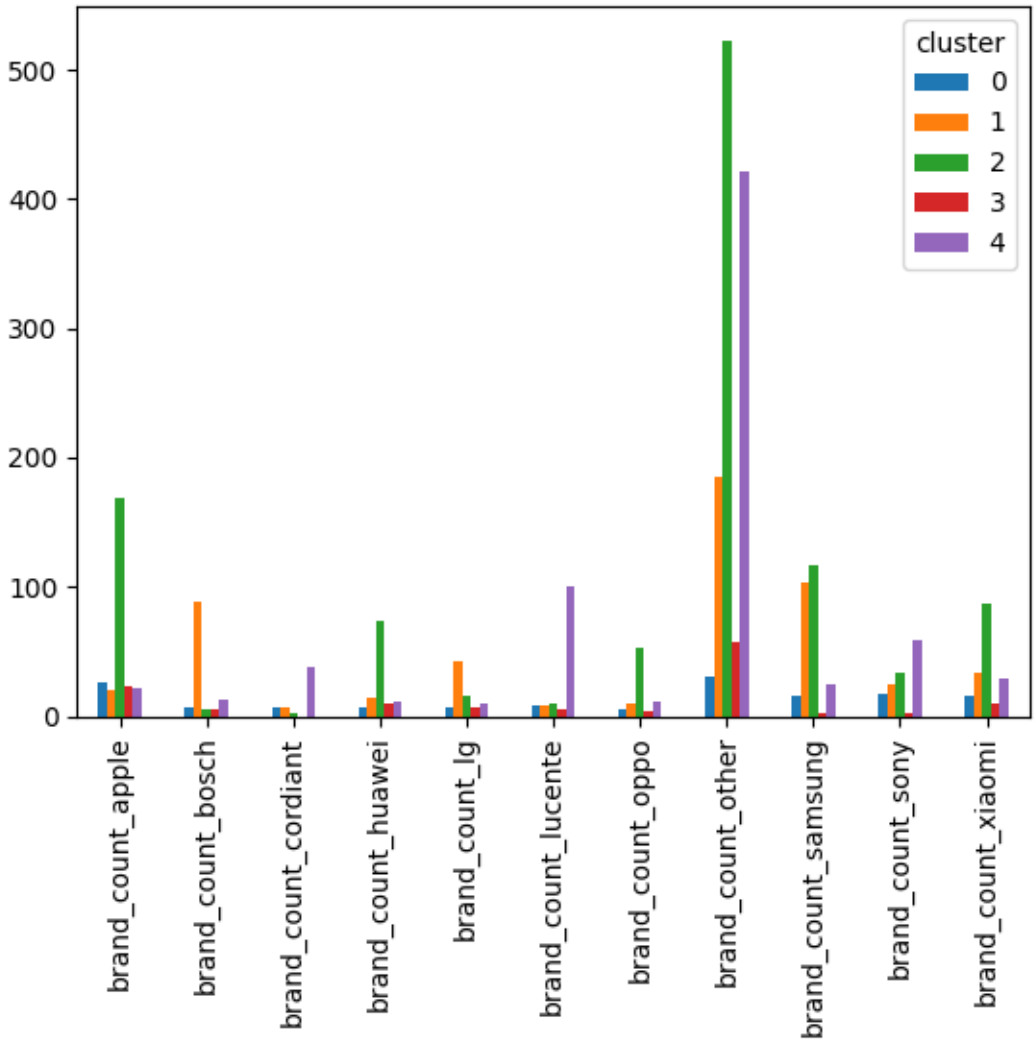

Рисунок 3.17 – Максимальне значення переглядів за брендами

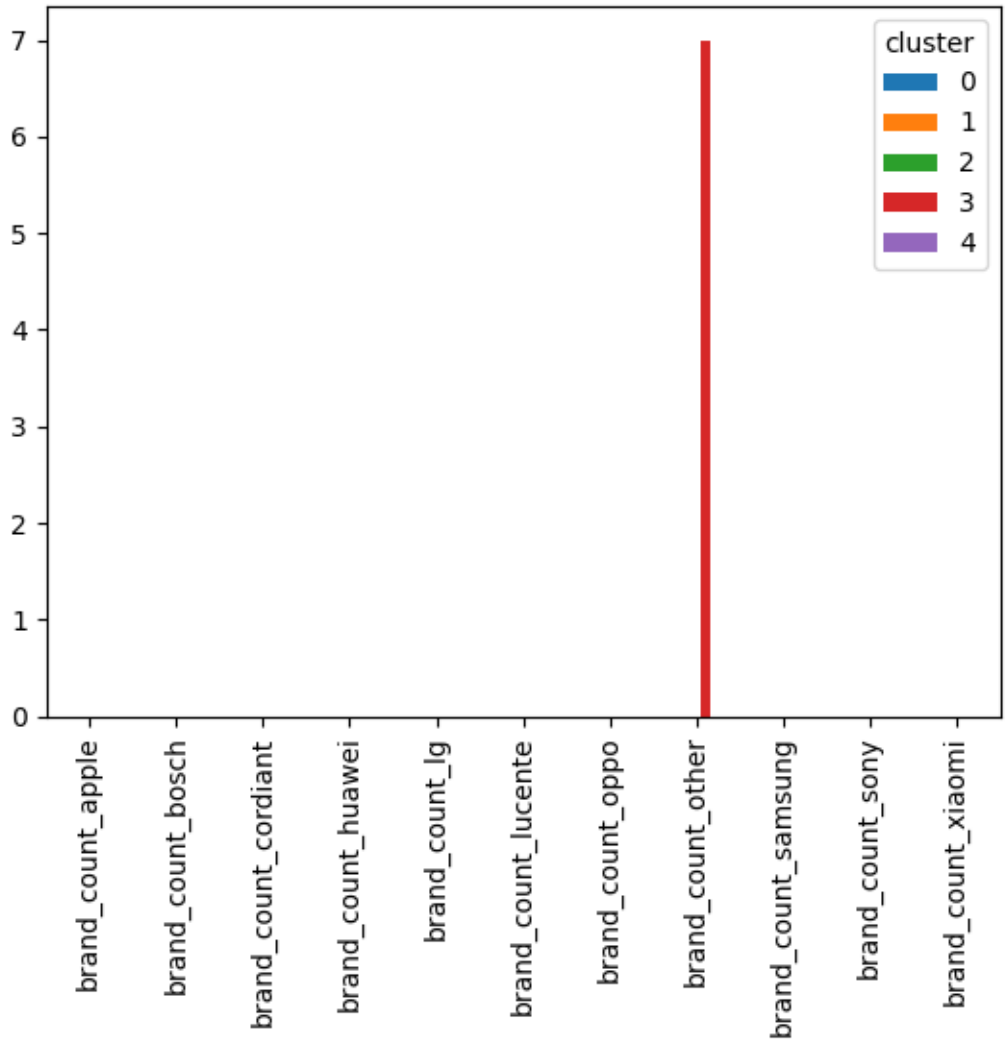

Рисунок 3.18 – Мінімальне занчення переглядів за брендами

ВИСНОВОК

Під час виконання роботи було розглянуто проблему сегментації користувачів інтернет магазину, а саме сегментації на основі поведінки та вподобань клієнтів.

В ході інформаційного огляду було проаналізовано декілька популярних програмних рішень, а також методів кластреизації, що можуть бути використані для вирішення проблем. Для програмної реалізації інформаційної технології було обрано метод k-середніх через швидкість роботи та простоту реалізації.

Результатом реалізації стала програма, написана на мові Python з використанням платформи Jupyter Notebook. Дана програма першим корком на основі набору данних про користувацькі сессії формує данні для навчання, а саме кілкість відвідин користувасчем сторінок, що містять інформацію про товар певного бренду та категорії. Далі оптимізує занчення кількості кластерів k за допомогою методів локтя та силуету, після чого за алгоритмом k-середніх присвоює значення кластера (сегмента) кожному користувачеві.

В ході аналізу результатів виявилось що найбільший сегмент – користувачі, що рідко, або одноразово користувалиь сайтом. Інших користувачів можна чітко розділити за категоріями, яким вони надають перевагу. Значення кількості переглядів за брендами виявилось значно менш значущим для кластеризації, оскільки, враховуючи обмеженість апаратних ресурсів, для кластеризації використувувались топ-10 найпопулярніших брендів та категорій, а більшість з популярних брендів випускають товар, що належить до категорії електроніка.

СПИСОК ЛІТЕРАТУРИ

1. Customer Segmentation: How to Effectively Segment Users & Clients [Electronic resource]. URL: https://blog.hubspot.com/service/customer-segmentation (accessed: 06.10.2022).
2. Varad R Thalkar. Customer Segmentation Using Machine Learning // International Journal of Scientific Research in Computer Science, Engineering and Information Technology. 2021.
3. Christy A.J. et al. RFM ranking – An effective approach to customer segmentation // Journal of King Saud University - Computer and Information Sciences. 2021. Vol. 33, № 10.
4. Sari J.N. et al. Review on customer segmentation technique on ecommerce // Adv Sci Lett. 2016. Vol. 22, № 10.
5. Zhou J., Wei J., Xu B. Customer segmentation by web content mining // Journal of Retailing and Consumer Services. 2021. Vol. 61.
6. Different Types of Consumer Segmentation | Audiense [Electronic resource]. URL: https://resources.audiense.com/blog/types-of-consumer-segmentation (accessed: 13.10.2022).
7. Build new segments - Analytics Help [Electronic resource]. URL: https://support.google.com/analytics/answer/3124493?hl=en#zippy=%2Cin-this-article (accessed: 06.10.2022).
8. Customer Segmentation Tools: 10 Best Tools for SaaS Businesses in 2022 [Electronic resource]. 2022. URL: https://userpilot.com/blog/customer-segmentation-tools/ (accessed: 06.10.2022).
9. 2.3. Clustering — scikit-learn 1.1.2 documentation [Electronic resource]. URL: https://scikit-learn.org/stable/modules/clustering.html#hierarchical-clustering (accessed: 17.10.2022).
10. Rasyid L.A., Andayani S. Review on Clustering Algorithms Based on Data Type: Towards the Method for Data Combined of Numeric-Fuzzy Linguistics // J Phys Conf Ser. Institute of Physics Publishing, 2018. Vol. 1097, № 1.
11. Exploring Clustering Algorithms: Explanation and Use Cases - neptune.ai [Electronic resource]. URL: https://neptune.ai/blog/clustering-algorithms (accessed: 17.10.2022).
12. Dang S. Performance Evaluation of Clustering Algorithm Using Different Datasets // IJARCSMS. 2015. Vol. 3. P. 167–173.

13. Zhang T., Ramakrishnan R., Livny M. BIRCH: An Efficient Data Clustering Method for Very Large Databases // Proceedings of the 1996 ACM SIGMOD international conference on Management of data - SIGMOD '96. New York, New York, USA: ACM Press.

14. ML - Clustering Mean Shift Algorithm [Electronic resource]. URL: https://www.tutorialspoint.com/machine_learning_with_python/machine_learning_with_python_clustering_algorithms_mean_shift.htm (accessed: 17.10.2022).

15. Schubert E. et al. DBSCAN Revisited, Revisited // ACM Transactions on Database Systems (TODS). ACM PUB27 New York, NY, USA , 2017. Vol. 42, № 3.

16. Customer Segmentation with Clustering Algorithms in Python | by Muhammet Bektaş | Medium [Electronic resource]. URL: https://medium.com/@mbektas/customer-segmentation-with-clustering-algorithms-in-python-be2e021035a (accessed: 17.10.2022).

17. EM algorithm and Gaussian Mixture Model (GMM) | by Oxanne | CodeX | Medium [Electronic resource]. URL: https://medium.com/codex/em-algorithm-and-gaussian-mixture-model-gmm-6ea5e0cf9d6e (accessed: 17.10.2022).

18. Hartigan J.A., Wong M.A. Algorithm AS 136: A K-Means Clustering Algorithm // Appl Stat. 1979. Vol. 28, № 1.

19. Deng Y., Gao Q. A study on e-commerce customer segmentation management based on improved K-means algorithm // Information Systems and e-Business Management. 2020. Vol. 18, № 4.

20. Hartigan A., Wong M.A. A K-Means Clustering Algorithm // Journal of the Royal Statistical Society. 1979. Vol. 28, № 1.

21. Likas A., Vlassis N., J. Verbeek J. The global k-means clustering algorithm // Pattern Recognit. 2003. Vol. 36, № 2.

22. Peña J.M., Lozano J.A., Larrañaga P. An empirical comparison of four initialization methods for the K-Means algorithm // Pattern Recognit Lett. 1999. Vol. 20, № 10.

23. Ahmed M., Seraj R., Islam S.M.S. The k-means algorithm: A comprehensive survey and performance evaluation // Electronics (Switzerland). 2020. Vol. 9, № 8.

24. Cui M. Introduction to the K-Means Clustering Algorithm Based on the Elbow Method // Accounting, Auditing and Finance. 2020. Vol. 1, № 1.

25. Purnima B., Arvind K. EBK-Means: A Clustering Technique based on Elbow Method and K-Means in WSN // Int J Comput Appl. 2014. Vol. 105, № 9.

26. Shi C. et al. A quantitative discriminant method of elbow point for the optimal number of clusters in clustering algorithm // EURASIP J Wirel Commun Netw. 2021. Vol. 2021, № 1.

27. Sembiring Brahmana R.W., Mohammed F.A., Chairuang K. Customer Segmentation Based on RFM Model Using K-Means, K-Medoids, and DBSCAN Methods // Lontar Komputer : Jurnal Ilmiah Teknologi Informasi. 2020. Vol. 11, № 1.

28. Puspitasari N., Widians J.A., Setiawan N.B. Customer segmentation using bisecting k-means algorithm based on recency, frequency, and monetary (RFM) model // Jurnal Teknologi dan Sistem Komputer. 2020. Vol. 8, № 2.

29. SAPUTRA D.M., SAPUTRA D., OSWARI L.D. Effect of Distance Metrics in Determining K-Value in K-Means Clustering Using Elbow and Silhouette Method. 2020.

30. Silhouette Analysis in K-means Clustering | by Mukesh Chaudhary | Medium [Electronic resource]. URL: https://medium.com/@cmukesh8688/silhouette-analysis-in-k-means-clustering-cefa9a7ad111 (accessed: 17.11.2022).

31. NumPy documentation — NumPy v1.23 Manual [Electronic resource]. URL: https://numpy.org/doc/stable/ (accessed: 29.11.2022).

32. pandas documentation — pandas 1.5.2 documentation [Electronic resource]. URL: https://pandas.pydata.org/docs/ (accessed: 29.11.2022).

33. Getting Started — scikit-learn 1.1.3 documentation [Electronic resource]. URL: https://scikit-learn.org/stable/getting_started.html (accessed: 29.11.2022).

ДОДАТОК А

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
np.set_printoptions(precision=2)
```

```
data_file = "C:/univ/diploma/2019-Nov.csv.zip"
data = pd.read_csv(data_file, sep=',', nrows=1000000)
data.head()
```

```
data = data.drop(['event_time','product_id','category_id','price','user_session'], axis=1)
data.head()
```

	event_type	category_code	brand	user_id
0	view	electronics.smartphone	xiaomi	520088904
1	view	appliances.sewing_machine	janome	530496790
2	view	NaN	creed	561587266
3	view	appliances.kitchen.washer	lg	518085591
4	view	electronics.smartphone	xiaomi	558856683

```
data['category_code'] = data['category_code'].str.split(".",expand=True)[0]
data.head()
```

	event_type	category_code	brand	user_id
0	view	electronics	xiaomi	520088904
1	view	appliances	janome	530496790
2	view	NaN	creed	561587266
3	view	appliances	lg	518085591
4	view	electronics	xiaomi	558856683

```
top_categories = list(data.groupby('category_code').brand.agg([len]).sort_values(
  by='len', ascending=False)[:10].index)
data.loc[(~data['category_code'].isin(top_categories)),"category_code"] = "other"
```

```
by_category = data.groupby(['user_id', 'category_code']).agg(
    category_count=pd.NamedAgg(column="event_type", aggfunc="count")
).reset_index()
by_category = by_category.pivot(index = 'user_id', columns='category_code').fillna(0)
by_category.columns = list(map("_".join, by_category.columns))
by_category.head()
```

```
brands = data.loc[data.brand.notnull()]
top_brands = brands.groupby('brand').brand.agg([len]).sort_values(
  by='len', ascending=False)
top_brands_names = list(top_brands[:10].index)
top_brands.head(10).plot(kind='barh', figsize=(20,10))
```

```
data.loc[(~data['brand'].isin(top_brands_names)),"brand"] = "other"
```

```
by_brand = data.groupby(['user_id', 'brand']).agg(
    brand_count=pd.NamedAgg(column="event_type", aggfunc="count")
).reset_index()
by_brand = by_brand.pivot(index = 'user_id', columns='brand').fillna(0)
by_brand.columns = list(map("_".join, by_brand.columns))
by_brand.head()
```

```
features = pd.merge(by_category, by_brand, how = 'left', on='user_id')
features.head()
```

```
matrix = features.values
scaler = StandardScaler()
scaler.fit(matrix)
scaled_matrix = scaler.transform(matrix)
```

```
silhouette_scores  = []
distortions = []
n_clusters = range(4, 30, 1)
```

```
for n_clu in n_clusters:
  kmeans = KMeans(n_clusters=n_clu, random_state=7)
  clusters_clients =  kmeans.fit_predict(scaled_matrix)
  silhouette_avg = silhouette_score(scaled_matrix, clusters_clients)
  silhouette_scores.append(silhouette_avg)
  distortions.append(kmeans.inertia_)
```

```
 plt.figure(figsize=(16,8))
 plt.plot(n_clusters, distortions_, 'bx-')
 plt.xlabel('k')
 plt.xticks(np.arange(4,30,1))
 plt.ylabel('Distortion')
 plt.title('The Elbow Method showing the optimal k')
 plt.show()
```

```
plt.figure(figsize=(16,8))
plt.plot(n_clusters, silhouette_scores_, 'bx-')
plt.xlabel('k')
plt.xticks(np.arange(3,41,1))
plt.ylabel('silhouette_scores')
plt.title('The silhouette_scores showing the optimal k')
plt.show()
```

```
kmeans = KMeans(n_clusters=5, random_state=7)
clusters_clients =  kmeans.fit_predict(scaled_matrix)
features['cluster'] = clusters_clients
features.head()
```

```
category_columns = []
brand_columns = []
for x in features.columns:
    if 'category' in x:
        category_columns.append(x)
    if 'brand' in x:
        brand_columns.append(x)
```

```
features.groupby(['cluster'])[category_columns].mean().transpose().plot(kind='bar')
```

```
features.groupby(['cluster'])[category_columns].min().transpose().plot(kind='bar')
```

```
features.groupby(['cluster'])[category_columns].max().transpose().plot(kind='bar')
```

```
features.groupby(['cluster'])[brand_columns].mean().transpose().plot(kind='bar')
```

```
features.groupby(['cluster'])[brand_columns].min().transpose().plot(kind='bar')
```

```
features.groupby(['cluster'])[brand_columns].max().transpose().plot(kind='bar')
```

```
features.groupby(['cluster'])['cluster'].count().plot(kind='pie')
```