Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет *Комп'ютерних наук*
(повна назва)
Кафедра *Системотехніки*
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

другий (магістерський)
(освітньо-кваліфікаційний рівень)

Розробка та дослідження рекомендаційної системи на основі кластеризації користувачів
(тема роботи)

Виконав: студент V курсу, групи *ІТПм-22-2*
спеціальності 122 – «Комп'ютерні науки та інформаційні технології»
(шифр і назва спеціальності)

Тип програми Освітньо-професійна
(освітньо-професійна або освітньо-наукова)
Освітня програма Інформаційні технології проектування
(повна назва освітньої програми)

Скрит І.П.
(прізвище, ініціали)

Керівник *доц. Ситнікова П.Е.*
(прізвище, ініціали)

Допускається до захисту
Зав. кафедри СТ _____ *проф. Гребеннік І.В.*
(підпис) (прізвище, ініціали)

2024 р.

Я, як студент ХНУРЕ, розумію і підтримую політику закладу із академічної доброчесності. Я не надавала і не одержувала недозволену допомогу під час підготовки кваліфікаційної роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

«17» січня 2024 р. *Скрит І.П.*

Кваліфікаційна робота не містить відомостей заборонених до відкритого опублікування.

Керівник кваліфікаційної роботи _____

Кваліфікаційна робота виконана у відповідності до стандартів, що діють в Україні.

Керівник кваліфікаційної роботи _____

Попередній захист проведено __17.01.2024__

Керівник кваліфікаційної роботи *Ситнікова П.Е.*

Харківський національний університет радіоелектроніки

(назва вищого навчального закладу)

Факультет *Комп'ютерних наук* **Кафедра** *Системотехніки*

Спеціальність *122 «Комп'ютерні науки»*

Освітньо-кваліфікаційний рівень *другий* (*магістерський*)

Тип програми *освітньо-професійна*

Освітня програма *Інформаційні технології проектування*

ЗАТВЕРДЖУЮ:

Зав. Кафедри *СТ*

проф. Гребеннік І.В.

" _____ " _____ *2024 р.*

ЗАВДАННЯ
НА АТЕСТАЦІЙНУ РОБОТУ

Студентові *Скрит Ірині Петрівні*

(прізвище, ім'я, по батькові)

1. Тема роботи *Розробка та дослідження рекомендаційної системи на основі кластеризації користувачів*

затверджена наказом по університету від " *16* **"** *жовтня* **2023р. №***582Ст*

2. Термін подання студентом роботи *19 січня 2024 р.*

3. Вихідні дані до роботи (проекту) *Розробити рекомендаційну систему для магазину косметичних товарів на основі кластеризації користувачів.Перелік використовуваних програмних засобів: ОС Microsoft Windows v.10 або вище, утиліта командного рядка MySQL Command Line Cliet;веб додаток Jupyter Notebook та мова програмування Python .*

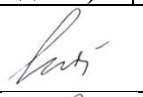
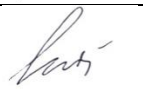
4. Зміст пояснювальної записки (перелік питань, що потрібно розробити)

4.1 *Вступ.* ***4.2*** *Аналіз предметної області.* ***4.2.1*** *Огляд та аналіз предметної області.* ***4.2.2****.Використання рекомендаційних систем в електронній комеруції.* ***4.2.3*** *Аналіз застосування досліджуваних методів в існуючих системах.* ***4.2.4****. Постановка задачі кваліфікаційної роботи.* ***4.3*** *Дослідження методів вирішення задачі.* ***4.3.1.*** *Типи, методи та особливості побудови рекомендаційних систем* ***4.3.1.1.*** *Колаборативна фільтрація* ***4.3.1.2.*** *Фільтрація на основі контенту****.*** ***4.3.2.*** *Основні алгоритми рекомендаційних систем****.*** ***4.3.2.1.*** *Кореляція Пірсона.* ***4.3.2.2*** *Алгоритми кластеризації.* ***4.3.2.3.*** *Алгоритм найближчого сусідства****.*** ***4.3.3.*** *Математичний опис алгоритму кластеризації користувачів****.*** ***4.3.3.1.*** *Кластерний аналіз.* ***4.3.3.2.*** *Класифікація алгоритмів кластеризації даних.* ***4.3.4.*** *Формальна постановка задачі кластеризації даних.* ***4.3.5.*** *Алгоритм K-means.* ***4.4.*** *Програмна реалізація кластеризації користувачів.* ***4.4.1.*** *Переваги машинного навчання для сегментації користувачів.* ***4.4.2.*** *Обгрунтування вибору середовища розробки****.*** ***4.4.3.****Обгрунтування вибору бібліотек.* ***4.4.4.*** *Підготовка даних для кластеризації.* ***4.4.5.*** *Рекомендації на основі кластеризації****.***

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

Креслення, схеми, плакати та/або комп'ютерні ілюстрації (слайди) на аркушах формату А4, що включаються до тексту пояснювальної записки або складу додатків: схема процесу кластеризації; блок-схема алгоритму K-means; схема моделі бази даних; екранні форми розроблених компонентів

6. Консультанти з роботи із зазначенням розділів роботи, що їх стосуються

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		(підпис)	(дата)
Аналіз предметної області.	*доц. Ситнікова П.Е.*		*10.11.2023*
Опис прийнятих проектних рішень	*доц. Ситнікова П.Е.*		*20.12.2023*

7. Дата видачі завдання *16 жовтня 2023 р.*

КАЛЕНДАРНИЙ ПЛАН

Пор. №	Назва етапів атестаційної роботи	Термін виконання етапів роботи	Примітка
1.	*Отримання завдання атестаційної роботи*	*15.11.23*	*Виконано*
2.	*Аналіз завдання, літератури та аналогів з теми атестаційної роботи*	*16.11 —25.11.23*	*Виконано*
3.	*Огляд та опис можливих рішень завдання*	*25.11 — 28.11.23*	*Виконано*
4.	*Структурне проектування системи*	*28.11 – 01.12.23*	*Виконано*
5.	*Вибір середовища для розробки системи*	*01.12– 05.12.23*	*Виконано*
6.	*Кластеризація та розробка рекомендаційної системи*	*05.12 — 15.12.23*	*Виконано*
7.	*Тестування розробленої інформаційної системи*	*15.12 — 20.12.23*	*Виконано*
8.	*Оформлення пояснювальної записки та документації до системи*	*20.12 — 31.12.23*	*Виконано*
9.	*Оформлення графічної частини та презентаційних матеріалів комп'ютерного захисту*	*10.01.24*	*Виконано*
10.	*Представлення на рецензування*	*17.01.24*	*Виконано*
	Представлення кваліфікаційної роботи до екзаменаційної комісії	*19.01.24*	*Виконано*

Студент _____ Скрит І.П.
(підпис)

Керівник роботи _____ доцент Ситнікова П.Е
(підпис)

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи: 58 сторінок, 23 рис., 25 джерела інформації, 9 лістингів.

ІНТЕРНЕТ МАГАЗИН, РЕКОМЕНДАЦІЙНА СИСТЕМА, КЛАСТЕРИЗАЦІЯ, КЛАСИФІКАЦІЯ, ПРОЕКТУВАННЯ, ЕЛЕКТРОННА КОМЕРЦІЯ.

Мета досліджень: розробка рекомендаційної системи на основі кластеризації клієнтів магазину з можливостями подальшого вдосконалення та розвитку.

Об'єктом досліджень кваліфікаційної роботи є процес рекомендації товарів магазину клієнтам, які розподілені по кластерам за певними характеристиками.

Предметом досліджень кваліфікаційної роботи є кластеризація користувачів системи електронної комерції з продажу косметики за наявними ознаками та побудова рекомендацій згідно отриманих результатів кластеризації.

У ході дослідження обраної тематики проекту проведено аналіз предметної області, також проаналізовано популярні аналоги системи та визначено вимоги до реалізації. Обрано метод кластеризації на основі якого побудована система рекомендацій магазину, описано кроки побудови та результати роботи отриманої системи.

Сфера застосування – вдосконалення системи електронної комерції магазину косметики, збільшення продажів, задоволених клієнтів та розвиток магазину загалом.

ABSTRACT

Explanatory note to the qualification work: 58 pages, 23 figures, 25 sources of information, 9 listing.

INTERNET SHOPS, RECOMMENDATION SYSTEM, CLUSTERIZATION, CLASSIFICATION, DESIGN, ELECTRONIC COMMERCE.

The purpose of research: development of a recommendation system based on the clustering of store customers with opportunities for further improvement and development.

The object of research of qualification work is the process of recommending store products to customers, which are divided into clusters according to certain characteristics.

The subject of the research of the qualification work is the clustering of the users of the e-commerce system for the sale of cosmetics according to the available features and the construction of recommendations according to the obtained clustering results.

During the study of the selected subject of the project, an analysis of the subject area was carried out, popular analogues of the system were also analyzed and requirements for implementation were determined. The clustering method was chosen, on the basis of which the store recommendation system was built, the construction steps and the results of the resulting system were described.

The scope of application is to improve the cosmetics store's e-commerce system, increase sales, satisfied customers, and develop the store in general.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАК, ОДИНИЦЬ І ТЕРМІНІВ

БД – база даних;

ІС – інформаційна система;

Кластер – група однакових або подібних елементів, зібраних разом або близько один до одного;

СУБД – система управління базами даних;

Центроід – середнє арифметичне положень всіх точок фігури. Визначення поширюється на будь-який об'єкт в n-вимірному просторі;

K-means – популярний метод кластеризації, впорядкування множини об'єктів у порівняно однорідні групи;

Python – високорівнева мова програмування, її використовують для розробки вебзастосунків, програмного забезпечення, машинного навчання;

SQL – мова структурованих запитів.

ВСТУП

У сучасному розвиненому світі на зміну традиційному маркетингу, заснованому на людському досвіді, приходить електронна комерція, основними елементами якої є майданчик (сайт, обліковий запис тощо), канали залучення трафіку (контекстна реклама, таргетована реклама), системи обробки замовлень, робота із клієнтами. Але навіть за зміні виду та роботи маркетингової сфери, основними маркетинговими цілями завжди залишаються: досягнення прибутку та стабільний попит на товари та послуги.

Системи рекомендацій мають вирішальне значення в електронній комерції для надання персоналізованих рекомендацій щодо продуктів на основі поведінки користувачів. Вони покращують взаємодію з користувачами, збільшують конверсії та сприяють утриманню клієнтів, адаптуючи пропозиції до індивідуальних уподобань. Ці системи відіграють важливу роль в ефективному управлінні запасами, адаптуються до мінливих тенденцій користувачів і пропонують конкурентну перевагу, забезпечуючи більш задовільний і персоналізований досвід покупок.

Сфери застосування цього проекту – системи електронної комерції будь-якого направлення, що потребують збільшення продажів та заохочення більшої кількості клієнтів. Також для прокату фільмів, книжок і тд.

Наразі існує багато магазинів, які не мають рекомендаційних систем взагалі, тому ідея актуальна та потребує якнайшвидшої реалізації, та надасть переваги на конкурентному ринку сьогодення.

Пояснювальна записка виконана згідно з методичними вказівками[1].

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Огляд та аналіз предметної області

Сьогодні інтернет-економіка розвивається дуже швидко, зростає попит на нову інформацію, яка відіграє роль факторів виробництва та стратегічних ресурсів. Це призвело до відкриття та розвитку нових інформаційних послуг. Одним з таких видів є інтернет-торгівля - це використання мережі інтернет для продажу товарів обраної тематики.

Ринок досить розвинений, конкуренція велика, тому потрібно досить ретельно слідкувати за тенденціями, бо в даний час все досить швидко змінюється. Можна обирати або довгостроковий бізнес або сезонний що буде приносити прибуток досить невеликий проміжок часу, але тоді виникає потреба кожного сезону оновлювати тематику товарів магазину, що є не досить вдалим рішенням.

З програмної точки зору, інформаційні система інтернет-магазину є складною і взаємопов'язаною структурою, призначеною для полегшення операцій цифрової роздрібної торгівлі. Основні сторінки інтернет-магазину це каталог товарів, вони відповідають вітринам в торгових точках, а консультантів замінюють підказки, інструкції та описи. Все інше як у звичайному магазині. Навіть у інтерфейсі інтернет-магазину присутні звичні для користувачів офлайн магазинів елементи, наприклад віртуальний «кошик», куди ми звикли складати обрані для покупки товари та оформлення замовлення - це своєрідна покупка товарів на касі магазину [2].

Продаж косметичних товарів – це лише один із багатьох видів реалізації товарів, від інших відрізняється досить специфічною аудиторією. Деякі категорії косметики не є першочерговим товаром та існують категорії товарів що підходять досить вузькому кому клієнтів. Найвдалішим з точки зору успішного бізнесу буде представлення якомога більшого асортименту для охоплення та задоволення потреб більшої кількості клієнтів. Але на перших

етапах розробки та просування нового продукту у вигляді сайту, требі зосередитися на якості послуг що надаватимуться, а не кількості асортименту.

В останні роки онлайн-покупки демонструють експоненційне зростання, що призводить до достатку вибору продуктів, що ускладнює користувачам пошук продуктів, що відповідають їх перевагам та потребам. Системи рекомендацій, також відомі як рекомендаційні системи, спрямовані на пом'якшення цієї проблеми шляхом аналізу даних користувача та надання персоналізованих рекомендацій щодо продуктів. Ці системи мають програми в різних областях: від електронної комерції та контентних платформ до соціальних мереж [3].

Не зважаючи на відносно недавній час виникнення потреби та інтересу користувача до рекомендаційних систем, різноманіття використовуваних рішень надзвичайно широке. Це значною мірою спричинено як відсутністю однозначно сформованих підходів до рішення задачі надання рекомендацій інформаційними системами, так і принциповою неможливістю створення єдиного підходу до задач такого типу, адже вибір оптимального розв’язку має критичну залежність від: даних, що ними маніпулює система, їх структурованості, характерної природи та стрімкості перетворення у дійсному інформаційному середовищі з часом, структуру рекомендаційної системи представлено на рисунку 1.1.

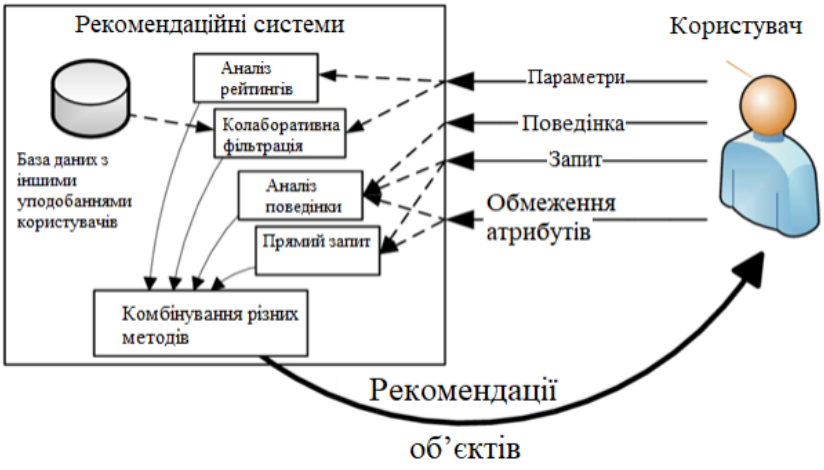


Рисунок 1.1 – структура рекомендаційної системи

Наступним кроком розглянемо поняття кластеризації або кластерного аналізу — це статистична процедура, задача якої полягає в розбитті вибірки об'єктів на підмножини, що не перетинаються і називаються кластерами. Кожен кластер має складається зі схожих об'єктів, а об'єкти різних кластерів мають істотно відрізнятися один від одного. Задача кластеризації відноситься до статистичної обробки, а також до широкого класу задач навчання без вчителя [4].

В даній кваліфікаційній роботі кластери – це підмножини дій користувачів на сайті магазину, а об’єкти кластерів – це користувачі (клієнти) магазину.

Для створення будь-якого проекту треба виконати попереднє проектування, оглянути найпопулярніші та успішні аналоги, взяти основні ідеї для реалізації а також додати власні задумки для отримання якісного продукту, що буде відрізнятися від аналогів та завоює прихильність аудиторії та такому великому ринку конкурентів.

Мета розробки – впровадження елементів рекомендаційної системи для магазину з продажу косметики на основі кластеризації користувачів.

1.2 Використання систем рекомендацій в електронній комерції

Системи рекомендацій для електронної комерції на веб-сайтах відіграють ключову роль у залученні постійних клієнтів, що є важливим показником для інтернет-магазинів. Це сприяє зменшенню витрат, заохочуючи клієнтів до додаткових покупок. Технологія рекомендаційних систем не є новинкою, що з'явилася одночасно з розвитком інтернет-магазинів та потокових сервісів. Застосування рекомендаційних систем охоплює різні галузі, такі як пошук фільмів, музики, наукових статей, а також різні сектори, включаючи роздрібну торгівлю, соціальні мережі, електронну комерцію та онлайн-банкінг.

У досконалому зображенні того, як recommender systems є застосовані і їх неспроможні можуть бути в Netflix, відновили company offering video content

через rental і streaming services. У своїх ранніх днях, Netflix керує своїми функціями за допомогою повідомлень VHS і DVD tapes, де користувачі будуть переглядати вміст, відновити диски, і отримати нові ones. Збільшення позитивної згоди було критичною фокусом для Netflix.

Різноманітні компанії пропонують широкий спектр продуктів, починаючи від книг, музичних підписок, фільмів і електроніки до товарів для дому, автомобілів і нерухомості. Навчальні онлайн-платформи, як-от Coursera, Udemy та Prometheus, пропонують широкий вибір курсів із різних предметів, як платних, так і безкоштовних. Зростаюча різноманітність альтернативних продуктів на платформах електронної комерції, як демонструють такі сервіси, як Amazon і Alibaba, ставить перед користувачами проблему у виборі правильних продуктів. Незважаючи на наявність пошукових фільтрів, усі товари мають чітко визначені атрибути, такі як книги та музика. Вирішенням цієї проблеми є ефективна система рекомендацій [5].

У сфері електронної комерції рекомендації зазвичай включають списки товарів, представлені у форматі Top-N. Чим вище пункт у списку, тим релевантнішим він сприймається користувачем. В даний час практично кожен інтернет-магазин містить певні рекомендації. Це не дивно, оскільки правильно налаштовані системи рекомендацій можуть значно підвищити дохід, рейтинг кліків (CTR), конверсії та інші важливі показники [5]. Крім того, вони можуть мати помітний позитивний вплив на взаємодію з користувачем, впливаючи на показники, які важче підрахувати кількісно, але дуже важливі для онлайн-бізнесу, такі як задоволеність клієнтів і розмір прибутку.

1.3 Аналіз застосування досліджуваних методів в існуючих системах

Провідні українські інтернет-магазини косметики змагаються за першість, пропонуючи широкий асортимент продукції, перевершуючи один одного в задоволенні потреб покупців та демонструючи безліч брендів, недоступних у звичайних магазинах. Вони також пропонують привабливі ціни

та акції. Варто підкреслити явні переваги онлайн-покупок, у тому числі відсутність фізичних черг, ширший вибір товарів порівняно з офлайн-аналогами, можливість скасування замовлень, перегляд каталогів товарів, не виходячи з дому, а також зручність виконання онлайн-платежів або розрахунків після отримання придбаних товарів. Нижче представлені деякі з найвідоміших українських інтернет-магазинів що будуть розглянуті як приклад: "Eva", "Watsons", "Makeup".

1. Лінія магазинів EVA[6] – найбільша мережа магазинів що відповідає за красу та здоров’я кожного клієнта, надає широкий асортимент парфумерії, косметики, аксесуарів, дитячої косметики та товарів для догляду, також побутових товарів як власних брендів так і широко розповсюджених на ринку зарубіжних брендів. Зараз момент компанія має понад 1000 фірмових магазинів по всій території Україні, а також власний інтернет-магазин EVA.UA. Планується подальше розширення кількості магазинів та вдосконалення сервісу в інтернеті, бо зараз це пріоритетне направлення. Огляд сторінки товару представлено на рисунку 1.2.

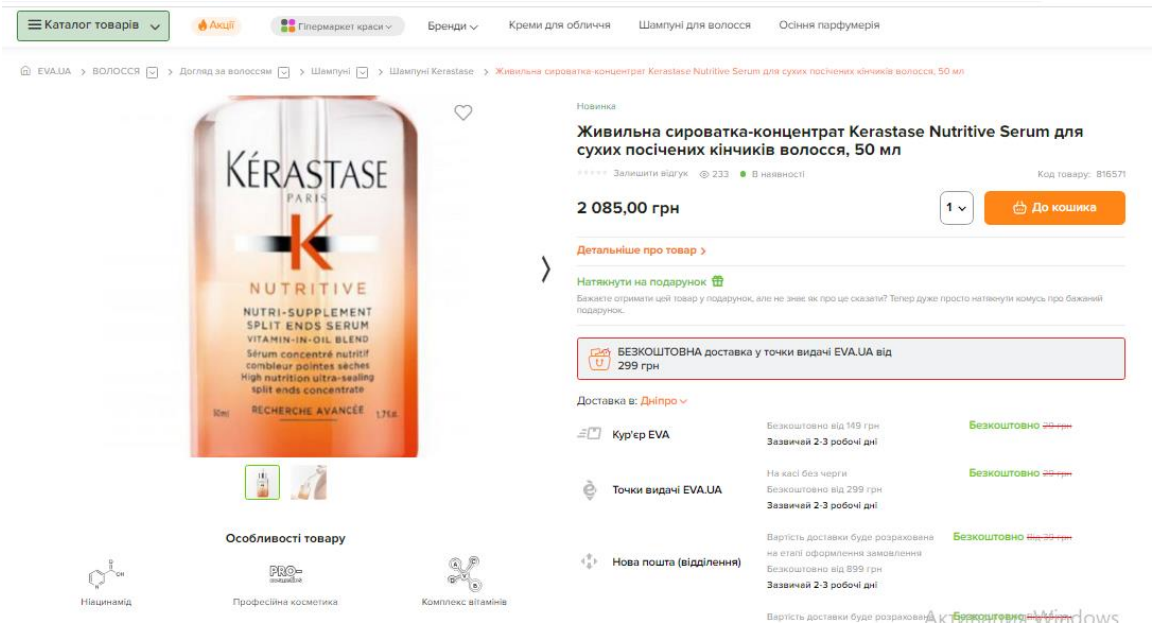


Рисунок 1.2 – картка товару в магазині EVA

Після детального огляду та тестування функціоналу системи електронної комерції магазину EVA було виділено основні бізнес-функції для користувача ІС, а саме:

– реєстрація та авторизація користувача у системі;

– надання інформації про магазин;

– перегляд каталогу товарів;

– перегляд картки товару із детальною інформацією про товар;

– формування кошику товарів та можливістю редагування наповнення;

– оформлення замовлення/скасування замовлення;

– оплата замовлення;

– відстеження замовлення;

– вихід із системи магазину.

Перелічені вище бізнес-функції є стандартними набором для успішної роботи будь-якого онлайн магазину, але в даний час цього не достатньо для перекриття потреб більшої кількості клієнтів. Тому магазини в більшості вдосконалюють функціонал і одним із прикладів таких функцій є рекомендації товарів (рисунок 1.3) підібрані відповідно інтересів клієнтів.

Розділ «Покупці також цікавились» відповідає за рекомендацію товарів згідно того що придбали інші клієнти разом із переглянутим товаром. Цей розділ формується за допомогою вивчення поведінки клієнтів магазину і групування схожих, і залежно від того в групу яких користувачів новий клієнт попадає так і формуються його рекомендації.

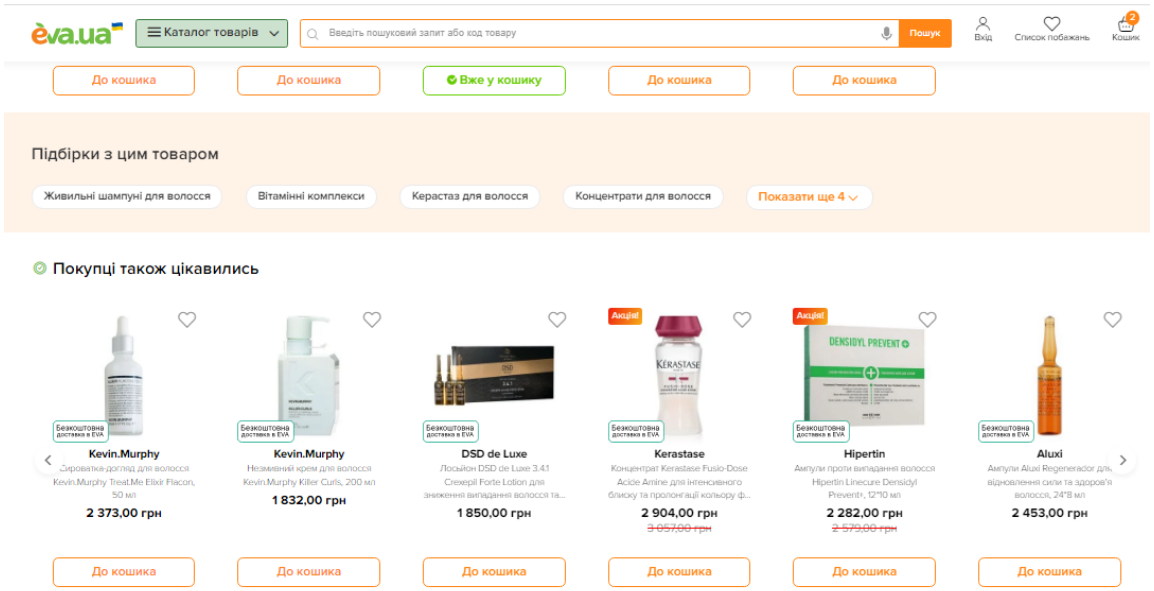


Рисунок 1.3 – рекомендації для клієнтів магазину EVA

Із недоліків даного магазину, згідно теми дослідження, можна виділити відсутність персональних рекомендацій, що засновані на вже придбаних продуктах магазину, або ж на переглянутих конкретно даним клієнтом товарів. Також було б доцільне опитування щодо особливостей клієнта, відповіді на які допоможуть чітко розділити групи товарів на підходящі та не підходящі і формувати каталог індивідуально. Вказаний функціонал був б доречним, оскільки предметна область, яка розглядається, стосується продажу косметичних товарів, де ключові міркування охоплюють унікальність клієнта, а також безпеку та належність, пов’язану з використанням косметики, яку вони обрали.

2. MakeUp [7] – один з найпотужніших онлайн-б’юті-ритейлерів Європи, MAKEUP вже успішно працює в 31 країні. Кожен день ми продовжуємо працювати для вас, розширюючи горизонти MAKEUP та створюючи унікальний б’юті-світ. Український представник «MakeUp» не має фізичних магазинів на території України, тому його діяльність зосереджена у мережі Інтернет що здійснюється за допомогою веб-сайту makeup.ua.

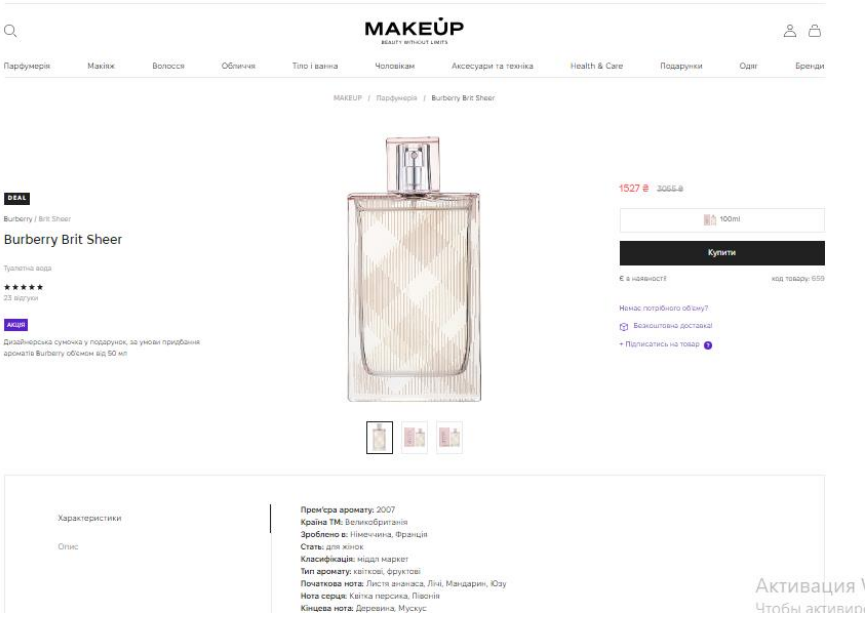


Рисунок 1.4 – картка товару в магазині MakeUp

Як і у розглянутому в попередньому підпункті інформаційної системи з продажу косметичних засобів, у «MakeUp» наявні аналогічні бізнес-процеси але з більш розширеним функціоналом:

– загальна інформація про діяльність магазину;

– реєстрація нового акаунта, або авторизація у вже існуючий;

– представлення каталогу товарів із додатковим функціоналом у вигляді фільтрації, сортування за категоріями і тд.;

– повнотекстовий пошук товарів на сторінці ;

– формування та редагування кошику;

– створення замовлення із можливістю відстежувати етапи обробки в особистому кабінеті/ скасування замовлення;

– можливість доставки поштовими службами країни або кур’єром магазину;

– програма лояльності для постійних клієнтів;

– можливість підписки на інформування про акційні пропозиції;

– вихід із системи.

Також варто відмітити дизайн інтернет-магазину, він значно відрізняється від попереднього, має більш лаконічний, трендовий дизайн, де вся увага направлена саме на продукт.

«MakeUp» має більш розширений розділ із рекомендаціями для клієнтів, представлений на рисунку 1.5, а саме:

"Схожі товари" – міститься перелік товарів що за характеристиками співпадають із обраним.

"Інші клієнти також купили" – міститься перелік товарів, які клієнти оформили в одне замовлення разом із обраним товаром. Тобто рекомендація на основі поєднання товарів в замовленнях іншими клієнтами.

"Спеціально для вас" – міститься перелік товарів згідно із особистими вподобаннями, схожі товари на ті, що переглядались клієнтом раніше.

Даний магазин має більш розширену рекомендаційну систему для товарів, що є вагомим плюсом для залучення клієнтів та збільшення товарообігу.

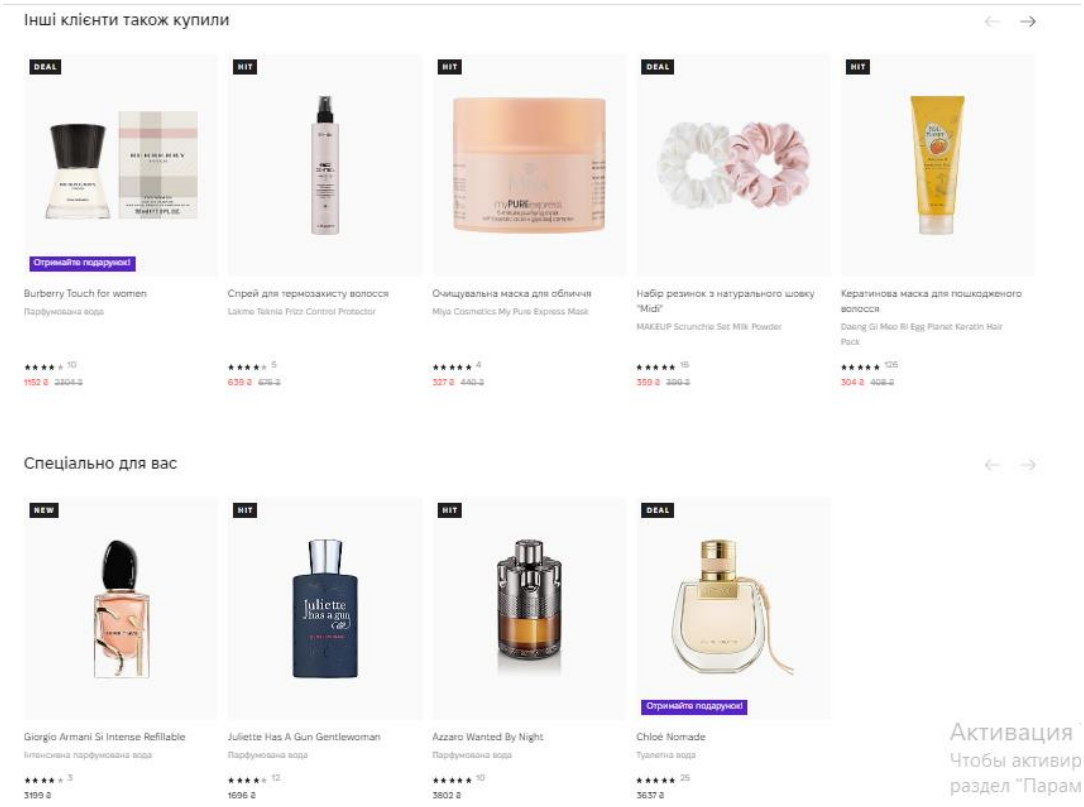


Рисунок 1.5 – рекомендації клієнтів магазину MakeUp

Не дивлячись на те що система рекомендує багато товарів із різними способами підбору каталогу, а саме за популярністю покупки або ж за кількістю переглядів, а також особисті рекомендації для кожного клієнта, залишається незрозумілим логіка за якою формуються ці списки товарів. Також виникає сумнів у правильності та доцільності підібраних товарів в категорії "Спеціально для вас", бо ніяких анкет для збору додаткової інформації про клієнта, потрібної для підбору косметики не було надано.

2. Постановка задачі кваліфікаційного проектування

Задача кваліфікаційної роботи стоїть в систематизації отриманих знань за період навчання та дослідженні методів що застосовуються при розробці рекомендаційних систем, формуванні чітких вимог до проекту.

Проаналізувавши існуючі рішення та розробки у даній сфері, була поставлена задача розробити рекомендаційну систему на основі попередньо спроектованого інтернет-магазину косметичних товарів, який міг би конкурувати з відомими онлайн магазинами країни та надавати рекомендації нових товарів, працювати без оцінок користувачів та показувати гарні результати. За допомогою розробленої рекомендаційної системи клієнт, окрім користування стандартним функціоналом інтернет-магазину, може отримати список товарів підібраних індивідуально під власні потреби та особливості, або ж переглянути товари які користуються популярністю у інших клієнтів магазину зі схожими інтересами.

Об’єкт дослідження – процес побудови рекомендацій на основі методу кластеризації користувачів.

Метою даного дослідження є систематична розробка та впровадження системи персоналізованих рекомендацій для інтернет-магазину косметики з використанням методів кластеризації користувачів. Основна мета – підвищити залученість користувачів та збільшити продажі за рахунок адаптації рекомендацій щодо косметичних продуктів до індивідуальних уподобань та

поведінки. У дослідженні розглядаються унікальні проблеми, пов'язані з динамічним характером косметичної індустрії та різноманітними споживчими уподобаннями. Запропонована система об'єднує кластеризацію користувачів для створення персоналізованих рекомендацій, що зрештою підвищує задоволеність користувачів та збільшує продажі на конкурентному ринку роздрібної торгівлі косметикою.

Завдання дослідження:

– порівняти існуючі підходи до формування рекомендацій;
– розглянути детально алгоритм кластеризації користувачів;
– створити математичний опис розроблюваної системи;
– визначити дані які є найбільш репрезентативними показниками вподобань користувача;
– проаналізувати способи відображення рекомендацій;
– розробити тестовий додаток, що буде надавати рекомендації для користувача базуючись на алгоритмі кластеризації K-means;
– протестувати розроблений додаток на взаємодію із користувачами системи та зробити висновки щодо ефективності розроблюваної системи;
– провести оцінку можливого подальшого розвитку системи.

При впровадженні запропонованого компонента в інформаційну систему необхідно вирішити проблеми, що виникають, пов'язані з його використанням:

– недостатність даних про клієнта магазину – користувачі постійно прагнуть скоротити час взаємодії з системою та не хочуть ділитися інформацією для зворотного зв’язку з системою. Ця проблема поширена в усіх системах рекомендацій, оскільки рекомендації спираються виключно на дані із профілю користувача. Отже, чим менше інформації надають користувачі, тим більш обмеженим стає набір відповідних рекомендацій. Таким чином, дуже важливо точно обрати дані, необхідні для створення рекомендацій, і мінімізувати дії клієнта;
– суперечливість даних: людська помилка або складнощі із заповненням

даних про себе (недостатні знання) можуть привести до невідповідності даних;

– неоднозначність даних: Неточне маркування товарів є потенційною проблемою, оскільки може призвести до неправильної класифікації ідентичних продуктів з різним маркуванням. Ця невідповідність може призвести до надання клієнту продуктів, які не відповідають його вимогам та потребам.

2 ДОСЛІДЖЕННЯ МЕТОДІВ ВИРІШЕННЯ ЗАДАЧІ

2.1 Типи, методи та особливості побудови рекомендаційних систем

Рекомендаційні системи відіграють ключову роль у покращенні користувальницького досвіду та забезпеченні успіху бізнесу у сфері онлайн-торгівлі. Метою цього наукового тексту є надання всебічного огляду типів рекомендаційних систем та в подальшому вибору одного із них для реалізації у власному проекті. Розуміючи основоположні принципи та методології, компанії можуть оптимізувати свої стратегії рекомендацій, щоб задовольняти різноманітні потреби та вподобання своїх клієнтів. Існує декілька методів побудови рекомендаційних систем, основні це: колаборативна фільтрація та фільтрація на основі контенту також гібридні моделі що включають обидва методи одночасно (рис. 2.1) [8].

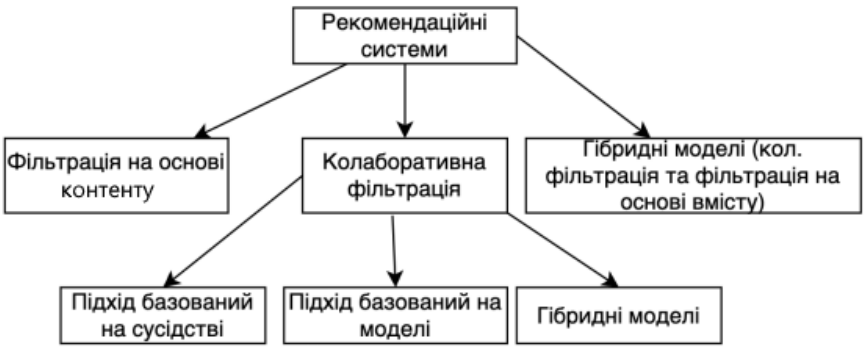


Рисунок 2.1 – Методи реалізації рекомендаційних систем

2.1.1 Колаборативна фільтрація

Це досить популярний метод у рекомендаційних системах, який використовує поведінку та вподобання користувачів для прогнозування того, які продукти чи елементи можуть сподобатися користувачеві.

Цей метод формує рекомендації, спираючись на характерні риси раніше придбаних товарів (модель поведінки клієнта, побудована на основі придбаних ним товарів раніше). Така модель повинна формуватися тільки з урахуванням минулої поведінки клієнта, але найефективніше враховувати історії запитів ще декількох клієнтів зі схожими інтересами. У тих випадках, коли колаборативна фільтрація бере до уваги поведінку інших користувачів, вона використовує інформацію про групу (groupknowledge) до створення рекомендацій з урахуванням схожості користувачів. По суті рекомендації засновані на автоматичному співробітництві безлічі клієнтів та на виділенні тих користувачів, які виявляють схожі переваги або шаблони поведінки.

Існує два основних типи колаборативної фільтрації: на основі користувачів та на основі елементів.

У колаборативній фільтрації на основі користувачів рекомендації створюються на основі вподобань користувачів, схожих на цільового користувача. Ідея полягає в тому, що якщо два користувачі мають подібні смаки та вподобання, вони, швидше за все, цікавляться схожими продуктами.

Наприклад, розглянемо сценарій, коли користувачі А та користувачі Б придбали схожі продукти в онлайн-магазині та їм сподобалися. Якщо користувач А купує новий товар, який користувач В ще не бачив, система може порекомендувати цей товар користувачеві В, припускаючи, що вони мають схожі смаки.

Колаборативна фільтрація на основі елементів зосереджена на пошуку схожості між елементами, а не користувачами як попередня. Якщо користувачеві подобається певний товар, система рекомендує інші товари, схожі на цей за характеристиками.

Наприклад, якщо користувач виявив цікавість до певної книги, система може порекомендувати інші книги, які сподобалися користувачам, яким також сподобалася ця книга.

Алгоритм колаборативної системи повинен знайти користувача, який оцінив конкретний предмет, та порахувати коефіцієнт кореляції векторів їх оцінок усім предметам у базі даних. Для цього можна скористатися методом k-найближчих сусідів, взяти користувачів із найвищими коефіцієнтами кореляції та подивитися, як вони оцінювали конкретний предмет. При цьому важливо розділити кожну оцінку користувачів на середню оцінку, щоб збільшити точність [9].

Переваги:

– незалежність користувача: спільна фільтрація не покладається на явне знання елементів; замість цього використовується взаємодія з користувачем, що робить його адаптованим до різних типів елементів;

– випадковість: спільна фільтрація може познайомити користувачів з елементами, які вони, можливо, не знайшли за допомогою чітких рекомендацій, сприяючи випадковому відкриттю;

– динамічна адаптація: у міру розвитку взаємодії користувачів спільне фільтрування може динамічно адаптуватися до мінливих уподобань користувачів.

Недоліки:

– проблема холодного запуску: с пільна фільтрація важко під час роботи з новими користувачами або елементами, які мають обмежену історію взаємодії, відома як проблема холодного запуску;

– розрідженість: у сценаріях із рідкісними даними про взаємодію між користувачами та елементами знайти суттєву подібність між користувачами чи елементами стає складно.

2.1.2 Фільтрація на основі контенту

Фільтрування на основі контенту — це популярний підхід у системах рекомендацій, який рекомендує елементи користувачам на основі характеристик або особливостей самих елементів. На відміну від

колаборативної фільтрації, яка базується на взаємодії між користувачем і елементом і схожості між користувачами, фільтрація на основі контенту зосереджується на атрибутах елементів і вподобаннях окремого користувача. Цей метод особливо ефективний у ситуаціях, коли явні взаємодії користувача з елементами є рідкісними, або коли намагаються рекомендувати елементи з певними атрибутами.

Представлення предмета: у фільтрації на основі контенту кожен елемент представлений набором дескрипторів або ознак. Ці функції можуть включати текстові описи, метадані або будь-яку іншу відповідну інформацію про елемент.

Наприклад, у системі рекомендацій фільмів характеристики фільму можуть включати жанр, режисера, акторів і текстовий опис сюжету.

Профіль користувача: система підтримує профіль користувача, який фіксує вподобання користувача на основі його минулих взаємодій або явних відгуків. Профіль користувача створюється шляхом аналізу характеристик елементів, з якими користувач позитивно взаємодіяв.

Алгоритм зіставлення: алгоритм зіставлення використовується для порівняння характеристик елементів із профілем користувача, щоб визначити релевантність кожного елемента для користувача. Загальні міри подібності включають косинусну подібність для текстових ознак або евклідову відстань для числових ознак [10].

Переваги:

– персоналізація: фільтрація на основі вмісту надає персоналізовані рекомендації на основі конкретних уподобань окремих користувачів;
– прозорість: рекомендації часто можна пояснити, оскільки вони є похідними від особливостей елементів і історії взаємодії користувача;
– пом'якшення проблеми холодного запуску: фільтрування на основі контенту менш чутливе до проблеми холодного запуску, коли нові елементи або користувачі мають обмежену історію взаємодії.

Недоліки:

– обмежена прозорість: фільтрування на основі вмісту може заважати познайомити користувачів із абсолютно новими чи неочікуваними елементами, оскільки рекомендації базуються на попередніх уподобаннях;

– розробка функцій: ефективна фільтрація на основі вмісту базується на добре розроблених функціях, які можуть потребувати досвіду домену та ретельного проектування функцій.

Вибір між контентною та колаборативною фільтрацією часто залежить від характеру проблеми з рекомендаціями, доступності даних і бажаного рівня персоналізації та прозорості, основні ідеї методів представлені на рисунку 2.2.

Рисунок 2.2 – Порівняння методів реалізації рекомендаційних систем

Незважаючи на різні алгоритми, засновані на різних даних, рекомендаційні системи допомагають полегшити проблему інформаційної перевантаження, дозволяє користувачам мати доступ до продуктів та послуг, які не завжди доступні користувачам у системі.

Рекомендаційний процес ділиться кілька основних етапів.

1. Збір інформації

Система збирає відповідну інформацію про користувачів для створення бази даних та формування рекомендацій. Ця база даних може містити докладну

інформацію про поведінку користувачів, дані споживчого кошика та багато іншого. Для підвищення точності системі потрібно додаткова інформація, специфіка якої залежить від методу функціонування рекомендаційних систем. Ця додаткова інформація повинна характеризувати модель користувача, оскільки ефективність будь-якої системи рекомендацій значною мірою залежить від її здатності відображати інтереси та переваги користувачів. Точні моделі необхідні надання своєчасних і точних рекомендацій з використанням різних методів прогнозування. [8].

2. Етап зворотнього зв'язку

Існує три види зворотнього зв'язку:

- явний;
- неявний;
- гібридний.

Явний зворотний зв'язок пропонує користувачеві через системний інтерфейс оцінити елемент, тим самим скласти рейтинг, щоб побудувати та покращити модель. Однак у такої моделі існує вагомий недолік - людський фактор.

Більшість користувачів не хочуть витрачати такий ресурс як час. Але навіть при врахуванні цього недоліку, як і раніше, вважається, що даний вид зворотного зв'язку є найбільш надійним, тому що тоді система не передбачає отримання переваг з дій, а також забезпечує прозорість процесу рекомендацій [11]. Уподобання користувача встановлюються за допомогою неявного зворотного зв'язку, який включає моніторинг різних дій користувача, таких як історія покупок, час, проведений на сайті, та посилання, за якими переходить користувач, серед інших показників. Ця форма взаємодії не тільки полегшує роботу користувача за рахунок використання вичерпної інформації про поведінку користувачів, але також вважається більш надійною. Навпаки, явний зворотний зв'язок вважається схильним до людського чинника, що потенційно може призвести до необ'єктивних та упереджених оцінок [11].

Переваги явного та неявного зворотного зв'язку об'єднані в гібридному вигляді. Гібридний зворотний зв'язок мінімізує недоліки двох видів зворотного зв'язку та отримує найбільш ефективну систему. Це досягається при використанні неявних даних як перевірки явної оцінки або допуску користувача давати явну зворотну зв'язок лише тоді, коли він самостійно вирішує висловити інтерес.

3. Етап навчання

Етап навчання застосовується у системах алгоритмів рекомендацій для фільтрації інформації, що була отримана на етапі збору. Цей процес фільтрації зібраної інформації потребує навчальної вибірки. У контексті машинного навчання використовують вхідні дані, такі як оцінки користувачів продуктів. Крім того, параметрами моделі, що вимагають навчання, є фактори, що визначають взаємодію користувачів і продуктів.

4. Етап рекомендації

Цей етап є останнім кроком і вирішення основних завдань. Використовуючи оброблені дані, система може пропонувати користувачеві певні елементи (продукти). Враховуючи, що для створення та оцінки функціональності рекомендаційної системи ми використовуємо продуктовий магазин, важливо наголосити на актуальності фільтрації даних про користувачів та придбаних ними продуктів, у тому числі поточного користувача.

2.2 Основні алгоритми рекомендаційних систем

2.2.1 Кореляція Пірсона

Кореляція Пірсона — це статистичний показник, який зазвичай використовується в системах рекомендацій для кількісної оцінки подібності між уподобаннями користувачів. Цей метод оцінює, наскільки добре оцінки або взаємодії одного користувача збігаються з оцінками іншого, надаючи цінний

показник схожості користувачів, який відіграє вирішальну роль у алгоритмах спільної фільтрації [12]. Для векторів користувачів u і v формула коефіцієнта кореляції набуває вигляду:

$$sim(u,v) = \frac{\sum_{i \in I} (r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in I} (r_{u,i} - \bar{r}_u)^2} \sqrt{\sum_{i \in I} (r_{v,i} - \bar{r}_v)^2}}$$

2.2.2 Алгоритми кластеризації

Алгоритми кластеризації – виявляють структуру в рядах випадкових(без розпізнавальних міток) даних. Такі алгоритми засновані на виявлення подібності між об'єктами (наприклад, між покупцями інтернет магазину) за допомогою обчислення їх відстані від інших елементів у просторі ознак (feature space) (кількість та якість товарів, що купуються в одній транзакції). Кількість незалежних ознак визначає розмірність простору ознак. Якщо елементи подібні один до одного, то вони поєднуються в кластер.

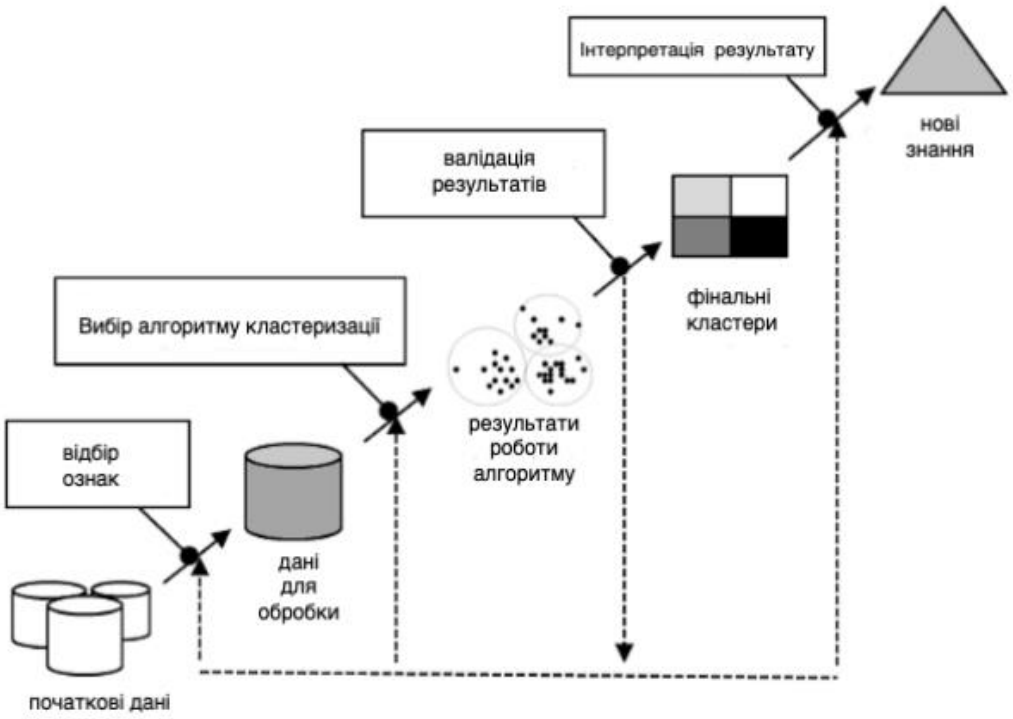


Рисунок 2.3 – Ілюстрація процесу кластеризації

Оскільки алгоритми кластеризації оперують безліччю параметрів, найчастіше функціонують у просторах високої розмірності та мають обробляти галасливі, неповні і вибіркові дані, їх ефективність може істотно відрізнятися залежно від конкретного додатка і типу даних. На практиці при виборі підходу до кластеризації для певного набору даних або завдання виникають складності [4].

Найпопулярнішим існуючим алгоритмом кластеризації є алгоритм k-means, який класифікує об’єкти на k кластерів. Спочатку сутності довільно призначаються до цих кластерів. Згодом центр маси визначається як функція його складових елементів. Потім оцінюється відстань кожного елемента кластера від центру його відповідного кластера. Якщо елемент виявляється ближче до іншого кластера, він переходить у цей кластер. Після перевірки всіх відстаней для всіх членів центри кластерів обчислюються заново. Як тільки досягається стійкий стан (за результату перевірки елементи не були переміщені) набір вважається кластеризованим належним чином, а потім алгоритм закінчує свою роботу. Обчислення відстані між двома об'єктами може бути важким для візуалізації. Для вирішення цієї проблеми слід розглядати кожен елемент кластера як багатовимірний масив і обчислювати йому т.зв. евклідова відстань [13].

Також існують інші різновиди кластеризації, такі як:

- теорія адаптивного резонансу (Adaptive Resonance Theory);
- нечітка кластеризація методом С-середніх (Fuzzy C-means);
- ймовірнісна кластеризація за допомогою ЕМ-алгоритму (Expectation-Maximization) і т.д.

2.2.3 Алгоритм найближчого сусідства

Метод сусідства — це клас алгоритмів, які використовуються в системах рекомендацій для створення персоналізованих пропозицій, враховуючи вподобання користувачів або предметів, які знаходяться в безпосередній близькості від цільового користувача або елемента. Ці методи використовують концепцію подібності чи відстані, щоб ідентифікувати сусідство користувачів або предметів, роблячи рекомендації на основі переваг цієї місцевої спільноти.

Метод k-найближчих сусідів — метричний алгоритм для автоматичної класифікації об'єктів. Основним принципом методу найближчих сусідів є те, що об'єкт приписується класу, найпоширенішому серед його сусідів. Сусіди беруться, виходячи з множини об'єктів, класи яких уже відомі, і, виходячи з ключового для даного методу значення k, вираховується найчисленніший серед них клас. Кожен об'єкт має кінцеву кількість атрибутів (розмірностей)

Найближчі сусіди користувач-користувач. Формула подібності користувача (наприклад, косинусної подібності): $sim(u,v) = \frac{\sum_i r_{ui} r_{vi}}{\sqrt{\sum_i r_{ui}^2}\sqrt{\sum_i r_{vi}^2}}$.

Етапи алгоритму:

1) обчислення подібності між цільовим користувачем і всіма іншими користувачами;
2) обрання k користувачів з найбільшою схожістю;
3) об’єднання переваг цих сусідів, задля рекомендації елементів цільовому користувачеві.

Найближчі сусіди предмет-предмет. Рекомендація елементів, схожих на ті, з якими взаємодіяв користувач, на основі вподобань користувачів, які взаємодіяли з подібними предметами. Формула подібності користувача (наприклад, косинусної подібності): $sim(i,j) = \frac{\sum_u r_{ui} r_{uj}}{\sqrt{\sum_u r_{ui}^2}\sqrt{\sum_u r_{uj}^2}}$.

Етапи алгоритму:

1) обчислення подібності між цільовим елементом і всіма іншими

елементами;

2) обрання k елементів з найбільшою схожістю;
3) рекомендація цих елементів користувачеві на основі їх взаємодії з цільовим елементом.

Алгоритм зважених найближчих сусідів — це варіант традиційного підходу найближчих сусідів, який вводить вагові коефіцієнти для уподобань сусідів, надаючи більше значення певним сусідам над іншими під час надання рекомендацій. Цей алгоритм покращує персоналізацію пропозицій, враховуючи не лише схожість між користувачами чи елементами, але й релевантність або надійність уподобань кожного сусіда.

Етапи алгоритму:

1) обчислити подібність: Обчисліть подібність між цільовим користувачем (або елементом) і всіма іншими користувачами (або елементами) у системі. Загальні показники подібності включають косинусну подібність, кореляцію Пірсона або подібність Жаккара;
2) виберіть сусідів:Виберіть k користувачів або предметів з найбільшою схожістю з цільовим. Ці користувачі або елементи утворюють околиці;
3) схема зважування: Призначте ваги кожному сусідові на основі їх схожості з ціллю. Вагові ваги можуть бути розраховані, наприклад, як величина, обернена відстані або необроблена оцінка подібності;
4) сукупні параметри: Поєднайте переваги сусідів, враховуючи їхню вагу. Часто використовується зважена сума або середнє зважене уподобань сусідів;
5) створити рекомендації: Рекомендуйте товари цільовому користувачеві на основі сукупних уподобань. Пропонуються елементи з вищими сукупними балами [14].

2.3 Математичний опис алгоритму кластеризації користувачів

2.3.1 Кластерний аналіз

Кластерний аналіз включає класифікацію об’єктів даних виключно на основі інформації, наявної в даних, яка описує ці об’єкти та їхні зв’язки. Мета полягає в тому, щоб сформувати групи, де об’єкти в кожній групі виявляють схожість один з одним, але відрізняються від об’єктів в інших групах[15]. Якість кластеризації покращується коли подібність між об’єктами всередині групи збільшується, що призводить до більшої відмінності між групами. Часто визначення кластера не є чітким. Для кращого розуміння складності визначення складу кластерів, представлено малюнок 2.4, який ілюструє 20 точок і три різні методи їх розподілу на кластери.

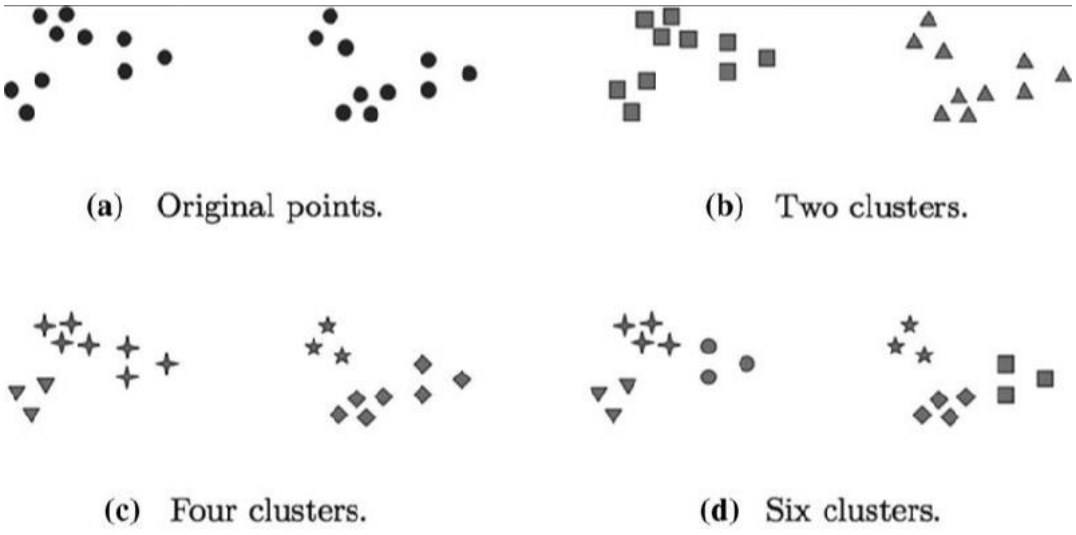


Рисунок 2.4 - Варіанти кластеризації одного набору точок

Існують різні типи кластеризації. Розглянемо основні з них:

- ієрархічна і неієрархічна;
- перетинаюча, непертинаюча і нечітка;
- повна і часткова.

Найбільш спірний аспект обертається навколо різниці між різними типами кластеризації, зокрема, чи є набори кластерів вкладеними чи навпаки,

використовуючи більш традиційні терміни ієрархічного чи неієрархічного. Неієрархічна кластеризація передбачає прямий поділ набору об’єктів даних на окремі підмножини (кластери), причому кожен об’єкт даних належить лише одній підмножині. Кожне окреме розташування кластерів на малюнку 2.4 є неієрархічним. Навпаки, якщо ми дозволяємо кластерам мати підкластери, ми переходимо до ієрархічної кластеризації, де набори суміжних кластерів організовані в структуру дерева. У цій ієрархічній схемі кожен вузол (кластер) у дереві, за винятком листкових вузлів, представляє об’єднання своїх нащадків (підкластерів), а корінь дерева охоплює всі об’єкти. Варто відзначити, що листя дерева часто, хоча і не завжди, являють собою окремі кластери, що складаються з окремих об'єктів даних.

Кластери, зображені на малюнку 2.4, не перекриваються, тому що кожен об’єкт призначено виключно одному кластеру. Однак у сценаріях, коли точка може входити до кількох кластерів, доцільно вибрати методи кластеризації, які дозволяють перекриватися. У ширшому контексті кластеризація, що перекривається, використовується для підтвердження одночасної належності об’єкта до кількох груп або класів. Наприклад, людина може одночасно бути студентом і співробітником університету. Кластеризація, що перекривається, також застосовується, коли об’єкт знаходиться між двома або більше кластерами і може бути призначений будь-якому з них.

Нечітка кластеризація вводить концепцію, що кожен об’єкт належить кожному кластеру з вагою членства в діапазоні від 0 (вказує на відсутність належності) до 1 (вказує на абсолютну належність). Цей підхід розглядає кластери як нечіткі набори, де об’єкти мають ваги в діапазоні від 0 до 1. Обмеженням нечіткої кластеризації є те, що сума ваг елементів у кожному кластері має дорівнювати 1. Подібним чином методи імовірнісної кластеризації обчислюють ймовірність належності об’єкта до кожного кластеру, при цьому сума цих ймовірностей в цілому дорівнює 1. Незважаючи на призначення ваг членства або ймовірностей для кожного об’єкта нечітка або імовірнісна

кластеризація не створює справжньої багатокласової ситуації, як у прикладі студент-працівник. Натомість він добре підходить для запобігання конфліктам під час призначення об’єктів одному кластеру в безпосередній близькості до кількох [4].

На практиці нечітка або імовірнісна кластеризація часто перетворюється на непересічної кластеризації шляхом призначення кожного об’єкта кластеру з найвищою вагою членства. Повна кластеризація передбачає присвоєння кожному об’єкту кластеру, тоді як часткова кластеризація дозволяє деяким об’єктам залишатися невіднесеними до визначених груп, часто представляючи шум, викиди або нецікаві елементи фону.

Незважаючи на наявність різних нотацій для оцінки практичної корисності кластера, кінцевою метою кластеризації є ідентифікація значущих груп об’єктів, корисність яких визначається цілями аналізу даних. Візуальне представлення, що використовує двовимірні точки, зазвичай використовується для ілюстрації відмінностей між типами кластерів.

2.3.2 Класифікація алгоритмів кластеризації даних

Алгоритми кластеризації даних можна загалом класифікувати на кілька категорій на основі їх підходу та характеристик:

1. Алгоритми розділення:

- K-Means:т ділить дані на визначену кількість кластерів (k) шляхом мінімізації суми квадратів відстаней між точками даних і центроїдами призначених їм кластерів;
- K-Medoids: Подібно до K-Means, але використовує фактичні точки даних як представників кластерів (medoids) замість центроїдів.

2. Ієрархічні алгоритми:

- агломеративний: починається з окремих точок даних як кластерів і

ітеративно об’єднує їх на основі близькості, доки не буде сформовано єдиний кластер;

- розділ: починається з усіх точок даних в одному кластері та рекурсивно поділяє їх на менші кластери.

3. Алгоритми на основі щільності:

- DBSCAN (просторова кластеризація додатків із шумом на основі щільності): ідентифікує кластери на основі областей із більшою щільністю точок даних, розрізняючи основні точки, граничні точки та шум;
- OPTICS (впорядкування точок для визначення структури кластеризації): Подібно до DBSCAN, але забезпечує більш гнучку структуру кластеризації.

4. Алгоритми на основі сітки:

- STING (Statistical Information Grid): Розділяє простір даних на структуру сітки, і кластери формуються на основі статистичної інформації в кожній комірці сітки.

5. Алгоритми на основі моделі:

- алгоритм очікування-максимізації (EM): припускає, що дані генеруються сумішшю кількох розподілів ймовірностей, і він ітеративно оцінює параметри цих розподілів, щоб знайти кластери;
- змішані моделі Гауса (GMM): специфічний тип кластеризації на основі моделі, де передбачається, що точки даних у кластері відповідають розподілу Гауса.

6. Нечітка кластеризація:

- нечіткі C-середні: розширює K-середні, дозволяючи точкам даних належати до кількох кластерів із різним ступенем приналежності, представлених нечіткими значеннями.

7. Самоорганізуючі карти (SOM):

- використовує штучні нейронні мережі для відображення

високовимірних даних на низьковимірну сітку, зберігаючи топологічні зв’язки між точками даних;

- кластеризація на основі обмежень: Включає визначені користувачем обмеження під час процесу кластеризації, щоб керувати формуванням кластерів на основі знань домену;

Ці категорії забезпечують класифікацію високого рівня, і в кожній категорії є численні специфічні алгоритми з унікальними функціями та застосуваннями. Вибір алгоритму кластеризації залежить від характеру даних, мети та розроблюваного проекту.

K-Means отримав перевагу перед іншими методами для розроблюваної системи рекомендацій інтернет-магазину косметичних товарів завдяки своїй масштабованості та інтерпретованості, що робить його особливо ефективним для великих наборів даних та забезпечує прозоре розуміння поведінки користувачів. У порівнянні з альтернативними алгоритмами K-Means продемонстрував чудову обчислювальну ефективність, адаптованість до мінливих переваг і плавну інтеграцію з гібридними підходами. Хоча кожен алгоритм має свої переваги, K-Means виявився добре збалансованим вибором, що поєднує в собі простоту та ефективність сегментації користувачів та рекомендацій у динамічному контексті інтернет-магазину.

2.4 Формальна постановка задачі кластеризації

Нехай X={x1 , x2 ,… xn } – множина N об’єктів, заданих в n-вимірному векторному просторі ознак:

$$x^k = (x_1^k, x_2^k, \dots, x_n^k)^T \quad , k = 1 \dots N$$

Нехай Y = {1,2,3,…} - множина номерів міток кластерів і між об’єктами задана функція відстані. Найчастіше береться евклідова метрика:

$$p_2(\mathrm{x}, \mathrm{x}') = \sqrt{\sum_{j=1}^{n} (\mathrm{x}_j - \mathrm{x}_j')^2}$$

Потрібно розбити кінцеву вибірку об’єктів X на K неперетинних підмножин:

$$S_k, k = 1 \dots K \; ; \; \mathrm{x} = \bigcup_{k=1}^{K} S_k$$

З метою забезпечення того, щоб кожен кластер був групою об'єктів, близьких один до одного згідно з метрикою p, при цьому об'єкти з різних кластерів істотно відрізнялись. Алгоритм кластеризації – це функція X → Y, яка довільному об’єкту x ∈ X ставить у відповідність номер кластера y ∈ Y.

Множина Y часто відома заздалегідь, проте частіше ставиться задача визначити оптимальне число кластерів з точки зору того чи іншого критерію кластеризації.

2.5 Алгоритм K-means

Алгоритм K-means – це популярний алгоритм кластеризації, який використовується в машинному навчанні та аналізі даних. Його основна мета розділити набір даних на k кластерів, де кожна точка даних належить кластеру з найближчим середнім (рисунок 2.5). Алгоритм працює ітеративно, призначаючи точки даних кластерам на основі їх близькості до центроїдів кластера, перераховуючи центроїди та повторюючи ці кроки до збіжності. K-середні чутливі до початкового вибору центроїдів і можуть сходитися до локальних оптимумів, тому часто виконується кілька ініціалізацій. Він широко застосовується в різних сферах для таких завдань, як сегментація клієнтів,

стиснення зображень і розпізнавання образів[13].

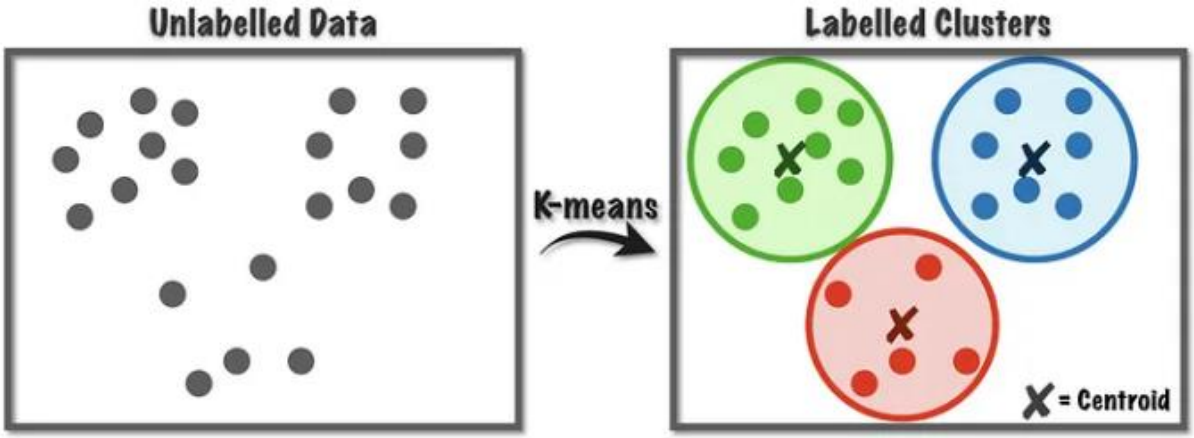


Рисунок 2.5 – Дані після обробки алгоритмом K-means

Даному алгоритму кластеризації потрібно знайти точки даних, характеристики яких подібні одна до одної, і, отже, ці точки належатимуть одному кластеру. Метод, за допомогою якого будь-який алгоритм кластеризації це робить, полягає в методі знаходження чогось, що називається «мірою відстані». Міра відстані, яка використовується в кластеризації K-середніх, називається мірою евклідової відстані, показано на рисунку 2.6 .

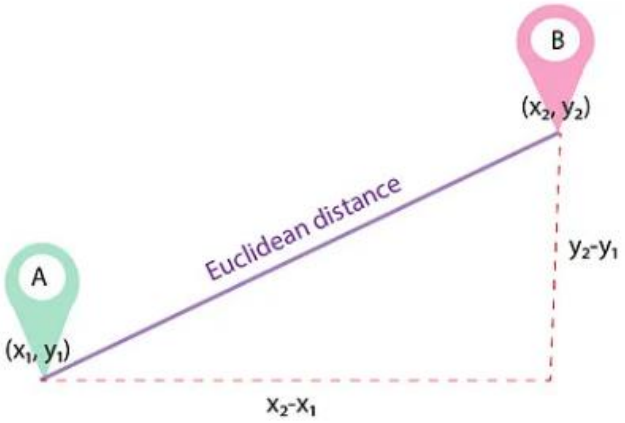


Рисунок 2.6 – Графічне зображення евклідової відстані

Метод k-середніх - це метод кластерного аналізу, мета якого є поділ m спостережень (з простору R^n) на k кластерів, при цьому кожне спостереження відноситься до того кластера, до центру (центроїду) якого воно найближче.

Як міру близькості використовується Евклідова відстань:

$$\rho(\mathrm{x},\mathrm{y}) = \left|\left|\mathrm{x} - \mathrm{y}\right|\right| = \sqrt{\sum_{p=1}^{n}(\mathrm{x}_p - y_p)}\,,\ \text{де}\ \mathrm{x},\mathrm{y} \in R^n\ .$$

Отже, розглянемо низку спостережень $(\mathrm{x}^{(1)},\mathrm{x}^{(2)},\dots,\mathrm{x}^{(m)}$, $\mathrm{x}^{(j)} \in \mathrm{R}^{(n)}$. Метод k-середніх поділяє m спостережень на k груп (або кластерів) ($\mathrm{k} \leq \mathrm{m}$) $S = S_1, S_2, \dots, S_k$, щоб мінімізувати сумарне квадратичне відхилення точок кластерів від центроїдів цих кластерів:

$$\min\left[\sum_{i=1}^{k}\sum_{\mathrm{x}}(j)_{eS_j}\left|\left|\mathrm{x}^{(j)} - \mu_i\right|\right|^2\right], \text{де}\ \mathrm{x}^{(j)} \in R^n, \mu_i \in R^n$$

μ_i - центроїд для кластера S_i.

Отже, якщо міра близькості до центроїду визначено, то розбиття об'єктів на кластери зводиться визначення центроїдів цих кластерів. Число кластерів k задається дослідником заздалегідь.

Розглянемо початковий набір k середніх (центроїдів) $\mu_1, \dots, \mu_k$ у кластерах $S_1, S_2, \dots, S_k$. На першому етапі центроїди кластерів вибираються випадково або за певним правилом (наприклад, вибрати центроїди, що максимізують початкові відстані між кластерами).

Відносимо спостереження до тих кластерів, чиє середнє (центроїд) до них найближче. Кожне спостереження належить лише одного кластеру, навіть якщо його можна віднести до двох і більше кластерів.

Потім центроїд кожного і кластера перераховується за наступним правилом:

$$\mu_i = \frac{1}{s_i}\sum_{\mathrm{x}^{(i)} es_i}\mathrm{x}^{(j)}$$

Таким чином, алгоритм k-середніх полягає у перерахуванні на кожному кроці центроїду для кожного кластера, отриманого на попередньому кроці.

Алгоритм зупиняється, коли значення μ_i не змінюються: $\mu_i^{mar\,t} = \mu_i^{mar\,t+1}$.

Алгоритм такий, що прагне звести до мінімуму середнє квадратне відхилення кореня в точках кожного скупчення. Основна ідея полягає в тому, що при кожній ітерації перераховується центр маси для кожного кластера, отриманого на попередньому кроці, потім вектори знову діляться на кластери, згідно з якими з нових центрів був найближчий в обраній метриці. Алгоритм закінчується, коли кластери не змінюються під час ітерації.

Вибір алгоритму k-means для рекомендаційної системи був зумовлений його масштабованістю, простотою та гнучкістю. Його ефективність у роботі з великими наборами даних, легкість інтерпретації, адаптованість до різних типів даних і взаємодоповнюваність з іншими методами рекомендацій роблять його всебічним вибором. Емпіричний успіх у різноманітних сферах і розумна швидкість конвергенції додатково підтверджують його придатність для надання персоналізованих рекомендацій ефективним і доступним способом.

На рисунку 2.7 показано послідовну блок-схему алгоритму k-means.

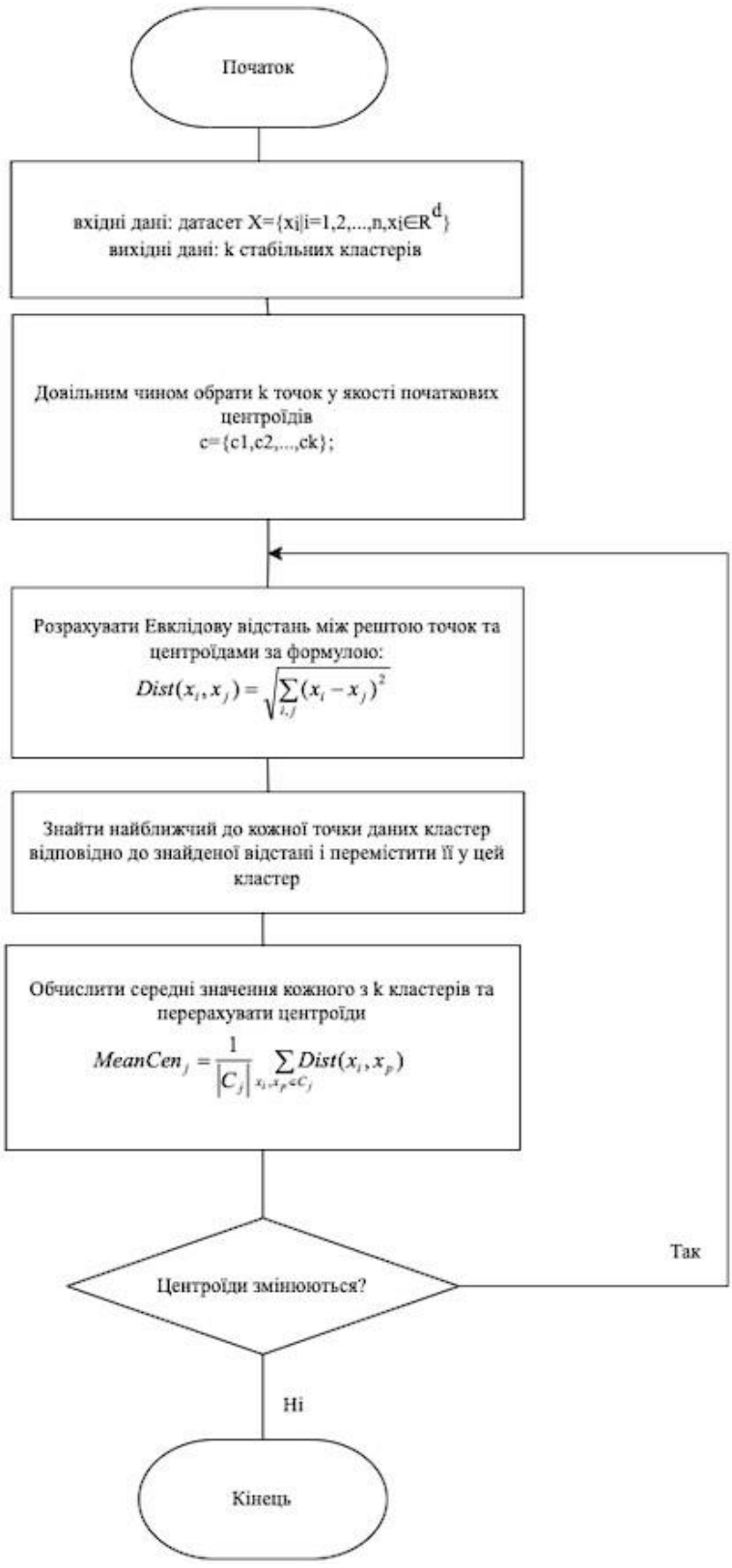


Рис. 2.7 – Блок-схема алгоритму K-means

3 ПРОГРАМНА РЕАЛІЗАЦІЯ КЛАСТЕРИЗАЦІЇ КОРИСТУВАЧІВ

3.1 Переваги машинного навчання для сегментації користувачів

Методології машинного навчання - відмінний інструмент для аналізу даних про клієнтів та пошуку ідей та закономірностей. Моделі штучного інтелекту є потужними інструментами для осіб, які ухвалюють рішення. Вони можуть точно ідентифікувати сегменти клієнтів, що набагато складніше зробити вручну чи за допомогою традиційних аналітичних методів.

Існує безліч алгоритмів машинного навчання, кожен із яких підходить для вирішення певного типу завдань. Навіщо треба запроваджувати машинне навчання для сегментації клієнтів?

Додатковий час

Виконання сегментації клієнтів вручну – трудомістке завдання. Процес ручного вивчення наборів даних та виявлення закономірностей може зайняти місяці чи навіть роки. Більше того, якщо підходити евристично, досягнута точність може не відповідати очікуваним перевагам.

Історично сегментація клієнтів вимагала ручного процесу, якому не вистачало точності. Вам доведеться вручну створювати та заповнювати різноманітні таблиці даних, досліджуючи дані подібно до детектива, що вивчає дзеркало. Сьогодні використання машинного навчання робить це завдання значно ефективнішим і простішим, дозволяючи вам приділяти час вирішенню складніших проблем, що потребують творчих рішень [16].

Легкість перенавчання

Сегментація клієнтів - це не проект за принципом "розроби один раз і користуйся назавжди". Дані постійно змінюються, тенденції коливаються, все продовжує змінюватись після розгортання вашої моделі. Зазвичай після розробки стає доступно більше зазначених даних, і це чудовий ресурс для покращення загальної продуктивності вашої моделі.

Є багато способів оновити моделі сегментації клієнтів, але два основних підходи:

- використовуйте стару модель як відправну точку та перенавчіть її;
- збережіть існуючу модель та поєднайте її вихідні дані з новою моделлю.

Оптимальне масштабування

Моделі машинного навчання, розгорнуті в робочому середовищі, забезпечують легкість масштабування завдяки використанню хмарної інфраструктури. Ці моделі виявляють достатню гнучкість для адаптації до майбутніх змін та зворотного зв'язку. Наприклад, представимо компанію, яка має зараз 10 000 клієнтів, і вони впровадили модель сегментації клієнтів. Якщо через рік кількість клієнтів збільшиться до 1 мільйона, то в ідеалі не потрібно створювати окремий проект для обробки цього обсягу даних, що збільшився. Моделі машинного навчання мають здатність обробки великих обсягів даних та ефективного масштабування виробництва.

Підвищена точність

Оптимальну кількість кластерів для даних про клієнтів легко визначити за допомогою методів машинного навчання, таких як метод ліктя. Застосування машинного навчання як забезпечує оптимальне число кластерів, а й істотно поліпшує продуктивність моделі.

3.2 Обґрунтування вибору середовища розробки, мови програмування

Середовище розробки, використане для цього проекту, складалося з Jupyter Notebook і Visual Studio Code. Jupyter Notebook служить інтерактивним інструментом, що дозволяє виконувати окремі блоки коду з негайною візуалізацією результату. Результати зберігаються разом із відповідним кодом у файлі проекту, що усуває необхідність повторно запускати ресурсомісткі блоки коду для перегляду результатів. Крім того, Jupyter полегшує створення комірок

із текстом із маркерами, пропонуючи чіткі засоби для опису дій у наступному блоці коду.

Процес розробки проводився на мові програмування Python 3. Розроблене програмне забезпечення вимагає включення сторонніх бібліотек. Подробиці цих бібліотек викладено в повному програмному коді, наданому в Додатку А.

Мова програмування Python

Python знаходить застосування у серверній веб-розробці, розробці програмного забезпечення, математиці та створенні системних сценаріїв. Він також широко використовується для швидкої розробки програм і служить мовою сценаріїв або склеювання для з'єднання існуючих компонентів. Це з його вбудованими структурами даних високого рівня, динамічної типізацією і динамічної прив'язкою. Python допомагає знизити витрати на обслуговування програм завдяки легкому для розуміння синтаксису та упору на читабельність.

Більше того, підтримка модулів та пакетів у Python спрощує створення модульних програм та полегшує повторне використання коду. Будучи мовою з відкритим вихідним кодом, що підтримується активною спільнотою, Python отримує вигоду від постійного внеску незалежних програмістів, що призводить до створення численних бібліотек та розширення функціональності [17].

Основні переваги Python:

- швидкість виконання програм, написаних на Python дуже висока: це пов'язано з тим, що основні бібліотеки Python написані на C++ і виконання завдань займає менше часу, ніж на іншими мовами високого рівня; – також існує можливість створення власних модулів для Python на C або C++;
- мова характеризується чітким і послідовним синтаксисом, продуманою модульністю і масштабованістю, завдяки чому початковий код написаних на Python програм можна легко прочитати;
- у стандартних бібліотеках Python можна знайти засоби для роботи з електронною поштою, протоколами Інтернету, FTP, HTTP, базами даних, тощо;

- скрипти, написані за допомогою Python виконуються на більшості сучасних ОС, така переносимість забезпечує Python популярність в різних областях програмування.

Мова програмування Julia

Розроблена, щоб забезпечити користувачам швидкість C/C++, зберігаючи при цьому зручність Python, Julia виділяється як високорівнева динамічна мова програмування. Ця характеристика дозволяє розробникам вирішувати проблеми з підвищеною швидкістю та ефективністю.

Julia особливо вміє справлятися зі складними обчислювальними завданнями, залучаючи перших користувачів переважно з таких галузей науки, як хімія, біологія та машинне навчання. Це вказує на те, що Джулія є універсальною та придатною для вирішення низки завдань, включаючи веб-розробку та розробку ігор. Джулія, яку широко вважають мовою наступного покоління для машинного навчання та науки про дані, отримує визнання в цих областях [18].

Мова програмування R

R служить і мовою програмування, і програмним середовищем, призначеним для статистичних обчислень, аналізу даних і графічного представлення даних. На її розвиток суттєво вплинули дві існуючі мови програмування: програмування S, що успадкувала семантику від Scheme.

R має значні можливості для проведення статистичного аналізу, що включає лінійну та нелінійну регресію, традиційні статистичні тести, аналіз часових рядів, кластерний аналіз тощо. Його універсальність ще більше покращується завдяки використанню додаткових функцій і пакетів, доступних на веб-сайті Comprehensive R Archive Network (CRAN). Хоча більшість стандартних функцій R написані мовою R, варто зазначити, що є можливість включити код, написаний на C, C++ або Fortran[19].

Мова програмування Matlab

MatLab виступає як інтерпретована мова програмування високого рівня, доповнена набором прикладних програм та інтегрованим середовищем розробки. Він призначений для проведення інженерно-математичних розрахунків, роботи з матрицями, базами даних і полегшення візуалізації [20].

MatLab включає в себе матричні структури даних, набір математичних функцій, об'єктно-орієнтовані функції та інтерфейси до програм, написаних іншими мовами програмування і т.д. В основному MatLab використовується в наукових дослідженнях і інженерних розробках.

Проаналізувавши наявні найбільш популярні мови програмування, що мють потрібний функціонал для реалізації завдання, була обрана мова програмування Python. Python — це мова програмування високого рівня, яка активно працює в сучасних сферах, таких як машинне навчання та великі дані. Він характеризується динамічною строгою типізацією та автоматичним керуванням пам’яттю, що забезпечує точніші обчислення в програмах, написаних на цій мові [17]. Python приділяє особливу увагу продуктивності розробника, читабельності коду та підтримці якості коду, вирішальним факторам у роботі з математичними методами. Він дотримується парадигми об'єктно-орієнтованого програмування. Крім того, синтаксис Python відомий своєю простотою, представляючи чистий і зрозумілий код, який легко читати та дозволяє швидко ідентифікувати помилки. Хоча Python спочатку був розроблений як освітня мова програмування, легкість вивчення та його широкі можливості не залишилися непоміченими в IT-спільноті. Як наслідок, Python швидко набув популярності і зараз робить значний крок вперед серед інших мов програмування.

3.3 Обґрунтування вибору бібліотек

Для реалізації алгоритмів та виконання поставленої задачі необхідно застосувати набір перевірених програмних бібліотек. Далі наведемо використані бібліотеки мови Python, функціональність яких задіяна в проєкті.

NumPy

NumPy, скорочення від Numerical Python, — це бібліотека, що містить багатовимірні масиви та набір функцій, призначених для роботи з цими масивами. Вона полегшує математичні та логічні операції, спеціально розроблені для операцій з масивами.

Використовуючи NumPy, розробник може виконувати такі операції:

- математичні та логічні операції над масивами;
- перетворення Фур'є та підпрограми для маніпуляції з формою;
- операції, пов'язані з лінійною алгеброю. NumPy має вбудовані функції для лінійної алгебри та генерації випадкових чисел.

NumPy, який часто використовується разом із такими пакетами, як SciPy (Scientific Python) і Matplotlib (графічна бібліотека), разом із цими компаньйонами є повною альтернативою MatLab — широко поширеній платформі для технічних обчислень. Проте Python, позиціонований як сучасна та більш всеосяжна мова програмування, став сучасною заміною MatLab [21].

Pandas

Pandas — це бібліотека Python для обробки та аналізу даних. Він представляє такі структури даних, як DataFrame і Series, надаючи інструменти для очищення, попередньої обробки, дослідження та аналізу даних. Він підтримує різні формати файлів, інтегрується з іншими бібліотеками та широко використовується в науці про дані та машинному навчанні завдяки своїй гнучкості та ефективності [22].

MatplotLib

Matplotlib — це комплексна бібліотека Python для створення статичних, анімованих та інтерактивних візуалізацій у різноманітних форматах. Він широко використовується для створення високоякісних графіків і діаграм у таких сферах, як наука про дані, наукові дослідження та інженерія. Matplotlib пропонує гнучкий і настроюваний інтерфейс, що дозволяє користувачам створювати різноманітні діаграми, включаючи лінійні діаграми, стовпчасті

діаграми, гістограми, діаграми розсіювання тощо. Він добре інтегрується з іншими бібліотеками Python, такими як NumPy і Pandas, що робить його популярним вибором для завдань візуалізації даних. За допомогою Matplotlib користувачі можуть керувати різними аспектами своїх візуалізацій, включаючи кольори, мітки та анотації, надаючи потужний інструмент для передачі інформації за допомогою графічних представлень [23].

Scikit-Learn

Scikit-Learn, також відомий як sklearn, є популярною бібліотекою машинного навчання на Python. Він надає простий і ефективний інструмент для аналізу даних і моделювання та побудований на NumPy, SciPy і Matplotlib. Scikit-Learn пропонує широкий набір алгоритмів машинного навчання для класифікації, регресії, кластеризації, зменшення розмірності тощо. Бібліотека розроблена з послідовним і зручним API, що полегшує користувачам впровадження та експериментування з різними алгоритмами. Крім того, Scikit-Learn надає інструменти для вибору моделі, оцінки та попередньої обробки даних, що сприяє її широкому використанню в спільноті машинного навчання як для початківців, так і для досвідчених практиків [24].

Мова програмування Python разом із бібліотеками NumPy, Pandas, Matplotlib і Scikit-Learn була обрана для проекту кластеризації користувачів системи з кількох вагомих причин. Python відомий своєю простотою, читабельністю та великою екосистемою бібліотек, що робить його чудовим вибором для різноманітних проектів. NumPy і Pandas надають потужні інструменти для обробки та аналізу даних, що має вирішальне значення для попередньої обробки та дослідження набору даних.

Matplotlib вибрано через його універсальність у створенні візуалізацій, допомагаючи в інтерпретації шаблонів даних і розуміння. Нарешті, Scikit-Learn пропонує повний набір алгоритмів та інструментів машинного навчання, що спрощує впровадження та оцінку моделей кластеризації для сегментації користувачів.

Поєднання цих бібліотек забезпечує злагоджений і ефективний робочий процес, полегшуючи завдання від попередньої обробки даних до розробки та оцінки моделі машинного навчання в єдиному середовищі Python.

3.4 Підготовка даних для кластеризації

Щоб створити систему рекомендацій, початковий крок включає формулювання та побудову керівних принципів і операційних правил. Модель, що перевіряється, спирається на дані клієнтів, зокрема на деталі про витрати кожного покупця, кількість куплених товарів та його лояльність.

Індивідуальні дані про покупця необхідні для визначення схожих, або подібних за поведінкою користувачів, систематизувавши їх та застосувавши алгоритм кластеризації, ми визначаємо до якого кластеру відноситься той чи інший користувач магазину.

Модель БД інтернет-магазину косметики показана на рисунку 3.1. Для реалізації та роботи рекомендаційної системи на основі кластеризації користувачів нам потрібні дані із User, Buy_products, product.

Для подальшої роботи, потрібно вилучити за допомогою запитів із таблиць «User» дані про ID клієнта магазину та дані про наявність або відсутність скарг на магазин, із «Products» ціни товарів , та із таблиці«Buy_products» кількість придбаних конкретним клієнтом товарів. Далі за допомогою збереженої процедури рахуємо суму витрат кожного користувача магазину і переносимо отримані дані в одну таблицю для опрацювання.

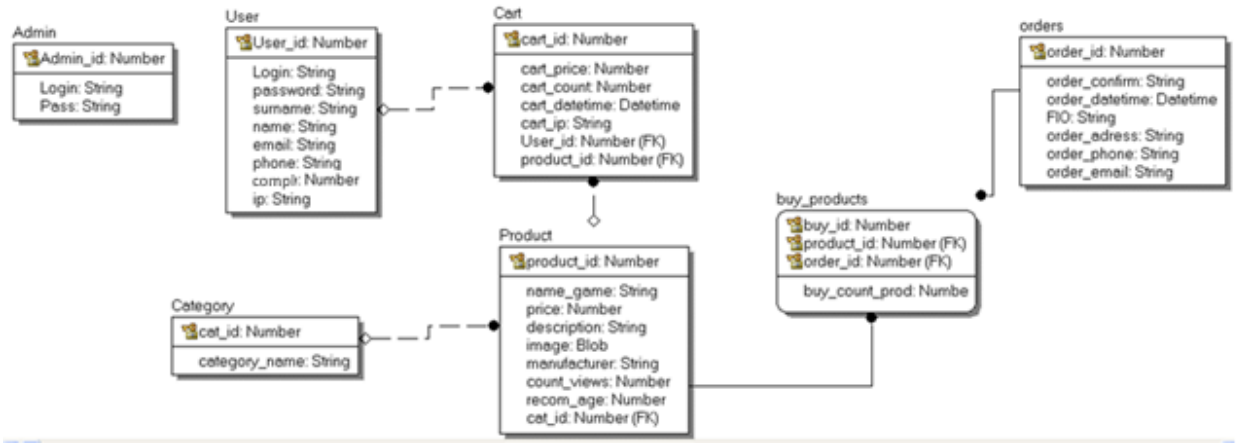


Рисунок 3.1 – Логічна модель даних

Сформовані дані перенесено в одну таблицю під назвою «customerdata» (рис. 3.2) та для коректності обробки алгоритмом кластеризації збережено даних, збережено у форматі csv.

```
customersdata.head()
```

	customer_id	products_purchased	complains	money_spent
0	649	1	0.0	260.0
1	1902	1	0.0	79.2
2	2155	3	0.0	234.2
3	2375	1	0.0	89.0
4	2407	2	0.0	103.0

Рисунок 3.2 – Дані з таблиці покупців

Останнім кроком для підготовки даних, є створення класів товарів для зіставлення із кластерами клієнтів та формуванні рекомендацій. Для цього треба вилучити із таблиці «Product» такі дані: назва, ціна, дата додавання, кількість проданих, знижки. На основі отриманих із БД даних створено відповідно до кількості кластерів, 5 класів продуктів представлених в магазині, а саме:

– високий ціновий сегмент (товари дорожчі за 1000 грн);

– новинки (товари додані в цьому місяці);
– хіт (товари з найбільшою кількістю замовлень);
– акційні (товари які мають знижку 50+%);
– низький ціновий сегмент (товари дешевше 200 грн).

3.5 Рекомендації на основі кластеризації

На відміну від алгоритмів навчання з учителем, кластеризація K-середніх є алгоритмом машинного навчання без вчителя. Цей алгоритм використовується, коли ми маємо немарковані дані. Немарковані дані означають вхідні дані без надання категорій чи груп. Наші дані щодо сегментації клієнтів для цієї проблеми такі.

Алгоритм виявляє групи (кластери) даних, де кількість кластерів представлено значенням K. Алгоритм діє ітеративно, присвоюючи кожні вхідні дані одному з кластерів K відповідно до наданими функціями. Все це робить k-середні цілком придатним для вирішення задачі сегментації клієнтів.

Даний набір точок даних згрупований за подібністю ознак. Результатом роботи алгоритму кластеризації K-середніх є:

- значення центроїдів для K-кластерів;
- мітки для кожної точки вхідних даних.

Наприкінці реалізації ми збираємося отримати вихідні дані, такі як група кластерів, а також те, який клієнт до якого кластера належить.

Перший крок це підключення необхідних бібліотеки Python, як показано у лістингу 3.1.

Лістинг 3.1 – ініціалізація бібліотек Python

```
import pandas as pd
import numpy as np
from sklearn.cluster import KMeans
import plotly.express as px
import plotly.graph_objects as go
```

```
import matplotlib.pyplot as plt
```

Ми імпортували бібліотеки pandas, NumPy sklearn, plotly і matplotlib. Pandas і NumPy використовуються для суперечок і маніпулювання даними, sklearn використовується для моделювання, а plotly разом з matplotlib використовуватимуться для побудови графіків і зображень.

Після імпорту бібліотеки нашим наступним кроком є завантаження даних у фрейм даних pandas (лістинг 3.2). Для цього ми будемо використовувати метод read_csv від pandas.

Лістинг 3.2 – імпорт файлу з даними

```
#Load customers data
customersdata = pd.read_csv("customers-data.csv")
```

Після завантаження даних нам потрібно визначити модель K-середніх. Це робиться за допомогою класу KMeans, який ми імпортували з sklearn, як показано в лістингу 3.3.

Лістинг 3.3 – визначення моделі K-середніх

```
kmeans_model = KMeans(init='k-means++',  max_iter=400, random_state=42)
```

Наступним кроком відбувається навчання моделі за допомогою даних що були отримані із таблиць БД. Це реалізовано за допомогою методу fit, як показано в лістингу 3.4. Модель дивиться на x та y і намагається знайти які-то взаємозв'язки або налаштувати коефіцієнти.

Тобто алгоритм отримує на вході дані про кількість куплених клієнтом товарів, наявність або відсутність скарг, та загальні витрати кожного клієнта. Далі порівнює ці значення та будує довільну кількість кластерів, де може міститись 1 або 2 елемента (клієнта), що є досить непродуктивним.

Лістинг 3.4 – метод fit

```
# Train the model
kmeans_model.fit(customersdata[['products_purchased','complains',
'money_spent']])
```

Тому пошук оптимальної кількості кластерів для даного набору даних є важливим для створення високопродуктивної моделі кластеризації k-середніх.

Наступним кроком знайдемо оптимальну кількість кластерів для заданого набору даних, а потім повторно навчимо модель кластеризації k-середніх із цими оптимальними значеннями k. Це дасть нашу остаточну модель.

Знаходження оптимальної кількості кластерів є одним із ключових завдань при реалізації алгоритму кластеризації k-середніх. Варто зазначити, що модель кластеризації k-середніх може збігатися для будь-якого значення K, але в той же час не всі значення K дадуть найкращу модель.

Для деяких наборів даних візуалізація даних може допомогти зрозуміти оптимальну кількість кластерів, але це стосується не всіх наборів даних. У нас є кілька методів, таких як метод ліктя, метод статистики розриву та метод середнього силуету, щоб оцінити оптимальну кількість кластерів для даного набору даних, розглянемо їх по черзі:

- метод ліктя знаходить значення оптимальної кількості кластерів, використовуючи загальну суму квадратів у межах кластера. Це показує, наскільки згенеровані кластери віддалені один від одного. У цьому випадку алгоритм K-means оцінюється для кількох значень k, а всередині кластера сума квадратів значень обчислюється для кожного значення k. Після цього ми будуємо графік залежності K від суми квадратичних значень. Після аналізу цього графіка кількість кластерів вибирається таким чином, щоб додавання нового кластера не змінило суттєво значення суми квадратичних значень;
- метод середнього силуету є мірою того, наскільки добре кожна точка даних відповідає відповідному кластеру. Цей метод оцінює якість кластеризації. Як правило, висока середня ширина силуету означає кращий результат кластеризації;
- метод статистики розривів є мірою значення статистики розривів. Gap

статистика — це різниця між загальними внутрішньокластерними змінами для різних значень k порівняно з їх очікуваними значеннями. Цей розрахунок виконується з використанням нульового еталонного розподілу точок даних. Оптимальна кількість кластерів – це значення, яке максимізує значення статистики розривів.

В роботі буде використовувати метод ліктя. Алгоритм кластеризації K-середніх кластеризує дані шляхом розділення заданих точок даних на k груп однакових дисперсій. Це ефективно мінімізує параметр під назвою інерція. У цьому випадку інерція — це не що інше, як суми квадратів відстаней у кластері, тобто кластери матимуть досить близьку відстань до визначеного центру і це дасть точне розуміння їх різниці.

Коли ми використовуємо метод ліктя, ми поступово збільшуємо кількість кластерів від 2 до тих пір, поки не досягнемо такої кількості кластерів, де додавання додаткових кластерів не призведе до значного падіння значень інерції.

Етап із такою кількістю кластерів називається ліктем моделі кластеризації. Ми побачимо, що в нашому випадку K =5. Для реалізації методу ліктя спочатку створюється наведена в лістингу 3.5 функція під назвою “try_different_clusters”. Він приймає два значення як вхідні дані: K (кількість кластерів), дані (вхідні дані).

Лістинг 3.5 – метод ліктя

```
# Create the K means model for different values of K
def try_different_clusters(K, data):

    cluster_values = list(range(1, K+1))
    inertias=[]

    for c in cluster_values:
        model = KMeans(n_clusters = c,init='k-
means++',max_iter=400,random_state=42)
        model.fit(data)
        inertias.append(model.inertia_)

    return inertias
```

Далі ми передаємо значення K від 1 до 12, це може бути довільне число, і змінювати його можна до тих пір поки не буде явно відображено лікоть, та обчислюємо інерцію для кожного значення k. Також будуємо графік значення K (на осі абсцис) проти відповідних значень інерції на осі Y.

Лістинг 3.6 – виклик методу ліктя та побудова графіку

```
# Find output for k values between 1 to 12
outputs = try_different_clusters(12,
customersdata[['products_purchased','complains','money_spent']])
distances = pd.DataFrame({"clusters": list(range(1, 13)),"sum of squared
distances": outputs})

# Finding optimal number of clusters k
figure = go.Figure()
figure.add_trace(go.Scatter(x=distances["clusters"], y=distances["sum of
squared distances"]))

figure.update_layout(xaxis = dict(tick0 = 1,dtick = 1,tickmode = 'linear'),
                  xaxis_title="Number of clusters",
                  yaxis_title="Sum of squared distances",
                  title_text="Finding optimal number of clusters using elbow
method")
figure.show()
```

Лікоть коду знаходиться на K=5, що показано на рисунку 3.4, де видно що після значення 5 по осі у графік перестає сильно спадати. Тому згідно правилам алгоритму k-means було обрано кількість кластерів 5, тому що якщо ми збільшимо кількість кластерів до більш ніж 5, буде дуже мала зміна інерції або суми квадратів відстані.

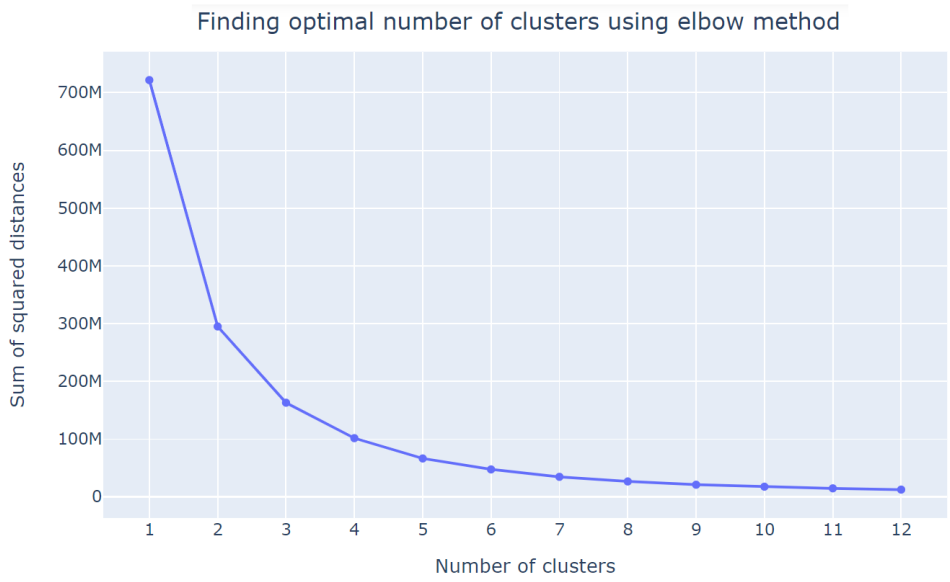


Рисунок 3.3 – візуалізація методу ліктя

Оптимальне значення K = 5. Стадія, на якій кількість кластерів є оптимальною, називається ліктем моделі кластеризації. Наприклад, на зображенні нижче лікоть знаходиться в п’яти кластерах (K =5). Додавання більше ніж 5 кластерів призведе до створення неефективної або менш продуктивної моделі кластеризації.

Останнім кроком для кластеризації користувачів, нам потрібно знову навчити модель кластеризації k-середніх із знайденою оптимальною кількістю кластерів (лістинг 3.7). Для цього використовуємо метод fit_predict, за допомогою якого обчислюються центри кластерів та спрогнозовується індекс кластера для кожного зразка.

Лістинг 3.7 – навчання моделі із відомою кількістю кластерів

```
# Re-Train K means model with k=5
kmeans_model_new = KMeans(n_clusters = 5,init='k-
means++',max_iter=400,random_state=42)

kmeans_model_new.fit_predict(customersdata[['products_purchased','complains'
,'money_spent']])
```

Завершальним кроком кластеризації буде реалізація коду за допомогою plotly express. Таким чином ми візуалізуємо кластери у трьох вимірах, утворені нашим алгоритмом k-середніх.

Додано новий стовпець під назвою «кластери» до наявного набору даних клієнтів. Цей стовпець відображає, який клієнт належить до якого кластера згідно із проведеною кластеризацією (лістинг 3.8). Використовується методи NumPy expm1. Функція NumPy expm1 повертає експоненціальне значення мінус один для кожного елемента, наданого всередині масиву NumPy як вивід. Тому метод np.expm1 приймає аргументи arr_name і out, а потім повертає масив як вихідні дані.

Лістинг 3.8 – модифікація таблиці

```
# Create data arrays
cluster_centers = kmeans_model_new.cluster_centers_
data = np.expm1(cluster_centers)
points = np.append(data, cluster_centers, axis=1)
points

# Add "clusters" to customers data
points = np.append(points, [[0], [1], [2], [3], [4]], axis=1)
customersdata["clusters"] = kmeans_model_new.labels_
```

Після виконання коду та додавання нової колонки до файлу з даними, отримуємо такий вигляд таблиці користувачів як показано на рисунку 3.4, де для кожного customer_id присвоєно номер кластера згідно з алгоритмом к-середніх.

```
customersdata.head()
```

	customer_id	products_purchased	complains	money_spent	clusters
0	649	1	0.0	260.0	4
1	1902	1	0.0	79.2	0
2	2155	3	0.0	234.2	4
3	2375	1	0.0	89.0	0
4	2407	2	0.0	103.0	0

Рисунок 3.4 – Оновлений файл з даними користувачів

Використавши код із лістингу 3.9, отримуємо візуалізацію п’яти створених кластерів. Тобто призначаємо кожній осі назву згідно з даними із таблиці клієнтів, а також нумеруємо отримані в процесі застосування алгоритму k-means кластери. Це робиться за допомогою plotly з експрес-бібліотекою.

Plotly — це бібліотека Python, яка використовується для побудови графіків, статистики, графіків і аналітики. Його можна використовувати разом із Python, R, Julia та іншими мовами програмування. Plotly — це безкоштовна бібліотека з відкритим кодом.

Plotly Express — це високорівневий інтерфейс над Plotly, який працює з декількома типами наборів даних і генерує високостилізовані графіки.

Лістинг 3.9 – візуалізація роботи програми

```
# visualize clusters
figure = px.scatter_3d(customersdata,
                    color='clusters',
                    x="products_purchased",
                    y="complains",
                    z="money_spent",
                    category_orders = {"clusters": ["0", "1", "2", "3",
"4"]}
                    )
figure.update_layout()
figure.show()
```

Виконавши код, отримуємо зображення в тривимірному просторі згідно заданих даних про користувачів системи.

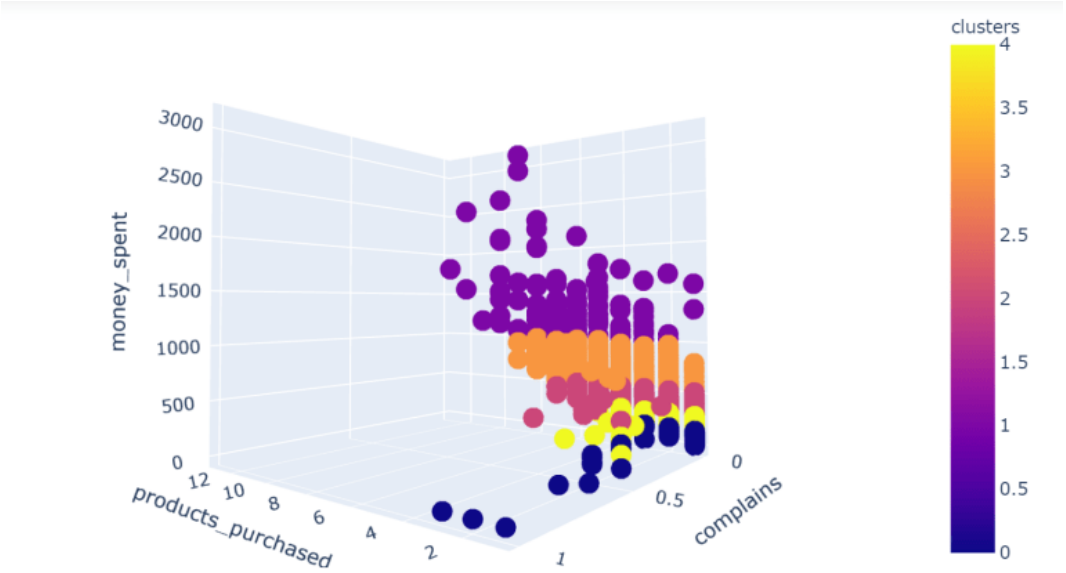


Рисунок 3.5 – Візуальне представлення кластерного аналізу

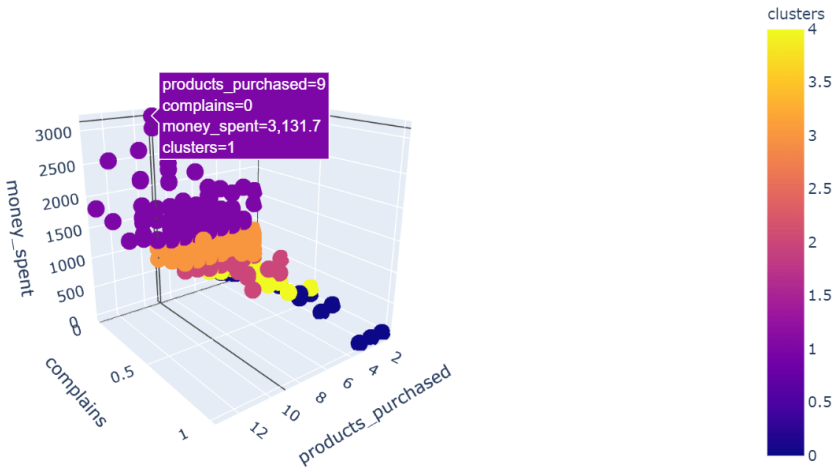


Рисунок 3.6 – Кластери користувачів системи

Нерозумно обслуговувати всіх клієнтів за допомогою однієї моделі продукту, електронної пошти, кампанії текстових повідомлень або реклами. Клієнти мають різні потреби. Універсальний підхід до бізнесу, як правило, призведе до меншої залученості, нижчих показників кліків і, зрештою, менших продажів. Кластеризація клієнтів є ліками від цієї проблеми.

Пошук оптимальної кількості унікальних груп клієнтів допоможе зрозуміти, чим відрізняються наші клієнти, і допоможе надати їм саме те, чого вони хочуть. Сегментація клієнтів покращує взаємодію з клієнтами та збільшує

дохід компанії. Ось чому сегментація є обов'язковою, якщо є мета перевершити своїх конкурентів і отримати більше клієнтів. Зробити це за допомогою машинного навчання, безперечно, є правильним шляхом.

За результатами проведеної роботи з кластеризації користувачів системи електронної комерції маємо такі висновки. Усі користувачі системи поділилися оптимально на 5 кластерів:

- 1 кластер (фіолетовий) – найбільш релевантні та лояльні клієнти магазину, що мають велику кількість куплених товарів та велику суму витрат в магазині, також не мають жодних претензій або скарг;

Для них було створено клас товарів «високого цінового сегменту» та рекомендації на головній сторінці видаються згідно цього класу. Система визначає до якого кластеру відноситься цей користувач. Для цього імпортовано в БД інтернет магазину в таблицю «User»стовбець «Cluster» із номером кластера відповідно до кожного користувача. Та відображає на головній сторінці відповідні товари (рис. 3.7).

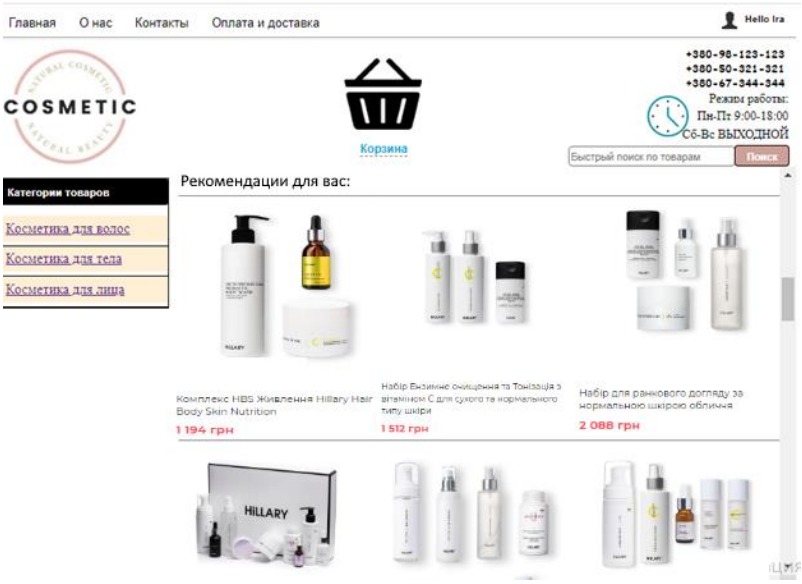


Рисунок 3.7 – Рекомендації для користувача із 1 кластеру

- 3 кластер (оранжевий) – клієнти із меншою кількістю придбаних товарів та середнім чеком в магазині, але все такі ж лояльні;

Для них підготовлено товари із класу «новинки», що додані на сайт в найближчим часом, для того щоб заохотити передивитись увесь асортимент нових товарів.

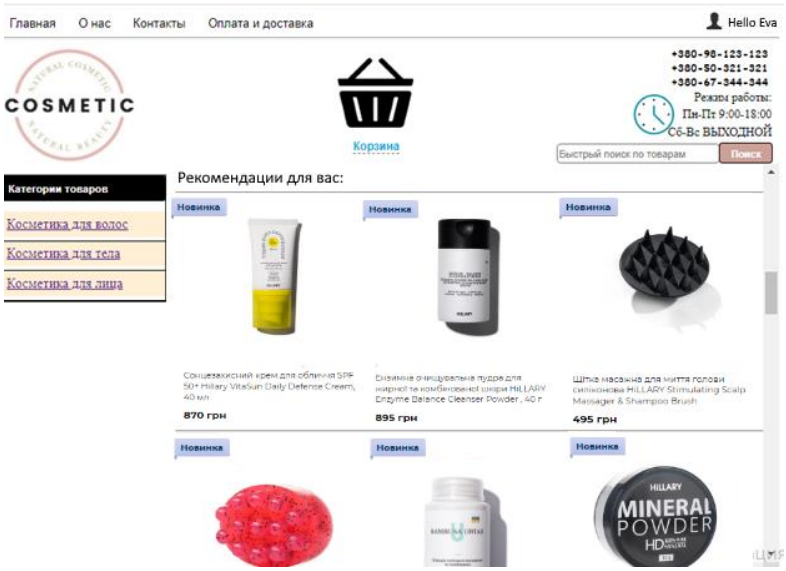


Рисунок 3.8 – Рекомендації для користувача із 3 кластеру

- 2 кластер (рожевий) – клієнти середнього звена, тобто невелика кількість покупок та малим чеком в магазині, нові клієнти;

Для них підготовлений клас хітових товарів, тобто тих що користуються найбільшим попитом серед клієнтів і є вірогідність що сподобаються та збільшать середню суму чеку даного клієнта.

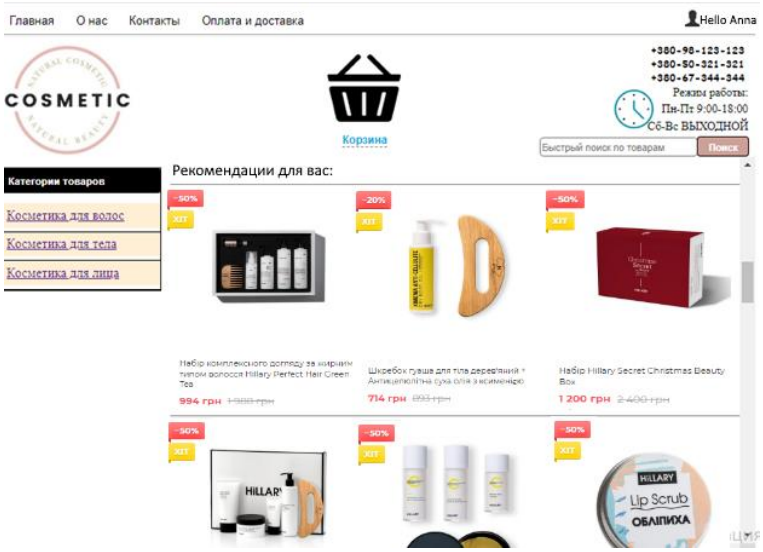


Рисунок 3.9 – Рекомендації для користувача із 2 кластеру

- 4 кластер (жовтий) – клієнти що мають малу кількість замовлень та

мінімальну суму покупок, але більш лояльні до магазину ніж кластер 0;

Для цих клієнтів підібрано товари із низькою вартістю, тобто надаємо можливість клієнту спробувати більшу кількість товарів та заохотити повернутись знову.

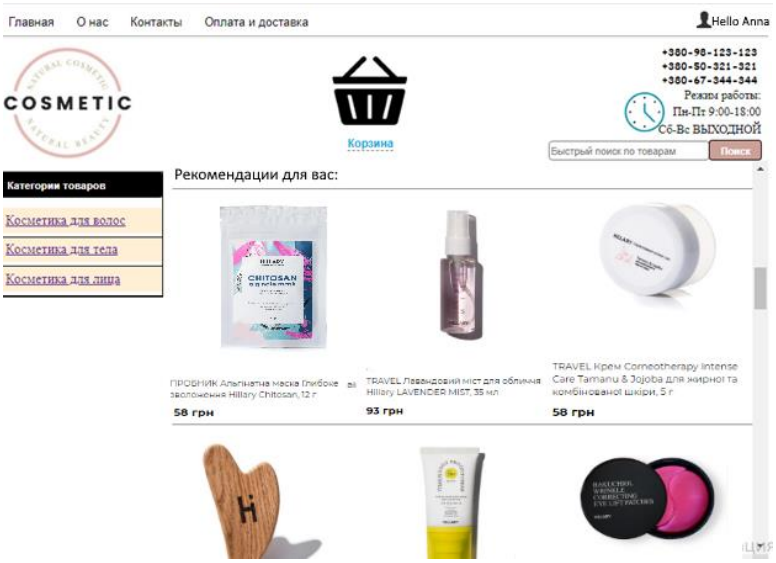


Рисунок 3.10 – Рекомендації для користувача із 4 кластеру

– 0 кластер (синій) – невдоволені клієнти.

Для них створено клас товарів із великою знижкою. Тобто рекомендуємо популярні товари із найбільшою знижкою серед інших магазинів. Тим самим спонукаючи їх на покупку та отримання гарного відгуку.

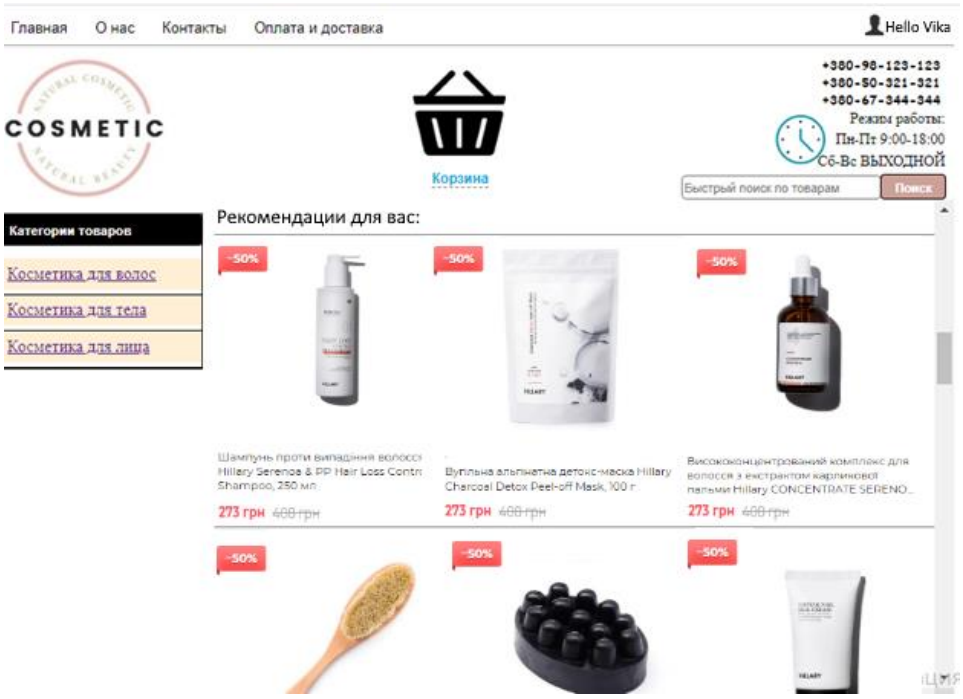


Рисунок 3.11 – Рекомендації для користувача із 0 кластеру

Завдяки отриманим кластерам, та проаналізувавши їх, була розроблена свого роду рекомендаційна система щодо товарів інтернет магазину косметики. Тобто для кожного кластеру створений відповідний список товарів для відображення на головній сторінці. В подальшому також можна розробити за цими кластерами розсилку на пошту, або купони безпосередньо на сайті. Але та система що наявна вже показала плюси її застосування, а саме, зменшилась кількість невдоволених клієнтів, та збільшився середній чек замовлень. Тобто можна зробити висновок що рекомендаційна система на основі кластеризації користувачів є вдалим рішенням для інтернет магазину косметичних товарів, але потребує доопрацювання.

ВИСНОВКИ

В ході виконання кваліфікаційної роботи було розглянуто проблему кластеризації користувачів інтернет магазину,

. В ході інформаційного огляду було проаналізовано декілька популярних магазинів та їх 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 даних про користувацькі сесії формує данні для навчання, а саме кількість відвідин користувачем сторінок, що містять інформацію про товар певного бренду та категорії. Далі оптимізує значення кількості кластерів k за допомогою методів локтя та силуету, після чого за алгоритмом k-середніх присвоює значення кластера (сегмента) кожному користувачеві.

У ході виконання дослідження було досягнуто наступних результатів:

– розроблено алгоритм кластеризації користувачів системи та протестовано на наборі даних, отримано результат та проаналізовано з наданням подальших рекомендацій;

– виходячи з предметної області, було розроблено пробну версію рекомендаційної системи, здатної ефективно вирішувати питання функціонування інтернет магазину;

– після проведеного аналізу застосування такої рекомендаційної системи було виявлено як її позитивні аспекти, і недоліки. В результаті було сформульовано вимоги до створення подібних систем.

Така система допоможе клієнтам магазину скоротити час перебування на сайті та спростити підбір потрібних товарів із асортименту, також спростить роботу працівникам магазину та збільшить дохід. Така система знайде своїх користувачів и буде завжди затребувана, так як електронна комерція зараз на піці своєї популярності.

Одною з важливих сторін аналізу даних у сучасному світі відокремлюють методи навчання без вчителя. Такі методи знаходять приховані закономірності чи шаблони в даних без зовнішнього втручання. Хоча методів кластеризації існує доволі багато, вони постійно розвиваються, і досі існують певні проблеми щодо їх ефективності у деяких окремих випадках. Основна проблема полягає в роботі з багатовимірними даними та пов’язаними з ними проблемами масштабованості через швидке зростання складності операцій. Інша не менш значна проблема полягає в інтерпретації результатів кластеризації, які можуть бути предметом довільної інтерпретації.

Алгоритми кластеризації знаходять потенційне застосування в різних аналітичних діях людини, охоплюючи маркетингові дослідження, природничі науки, біологію, оцінку надійності боржників, класифікацію документів на основі конкретних критеріїв, прогнозування небезпечних ділянок місцевості під час геологічних досліджень тощо. Ефективність кожного методу кластеризації залежить від таких факторів, як природа, розподіл, тип і розмір вхідних даних, що призводить до явних переваг і недоліків, пов’язаних з кожним методом.

Публікація

«Розробка та дослідження рекомендаційної системи на основі кластеризації користувачів». VIII Всеукраїнська науково-практична конференція «ПЕРСПЕКТИВНІ НАПРЯМКИ СУЧАСНОЇ ЕЛЕКТРОНІКИ, ІНФОРМАЦІЙНИХ І КОМП'ЮТЕРНИХ СИСТЕМ» MEICS-2023 22-24 листопада 2023 р., м. Дніпро, Україна.[25]

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Методичні вказівки до організації виконання та захисту атестаційної роботи на здобуття другого (магістерського) рівня вищої освіти для студентів усіх форм навчання спеціальності 122 – «Комп'ютерні науки», освітньо-професійна програма «Інформаційні технології проектування» / Упорядники: І.В. Гребеннік, В.Г. Іванов, Б.О. Колесник, А.І. Коваленко, Ю.В. Міщеряков, І.А. Урняєва – Харків: ХНУРЕ, 2021. – 54 с.

2. Структура інтернет-магазину [Електронний ресурс] // Режим доступу: https://dut.edu.ua/uploads/l_467_77066770 (дата звернення: 20.11.2023).

3. Інтернет-магазин/онлайн покупки [Електронний ресурс] // Режим доступу: https://uk.wikipedia.org/wiki/onlinestore (дата звернення: 25.11.2023).

4. Clustering algorithms: A comparative approach [Електронний ресурс] // Режим доступу: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6333366/ (дата звернення: 28.11.2023).

5. Jannach D., Zanker M., Felfernig A. Friedrich G. Recommender Systems. An Introduction. New York: Cambridge University Press 32 Avenue of the Americas, 2011. 352 P (дата звернення: 28.11.2023).

6. Інтернет-магазин EVA.UA - Evарианти на EVA.UA [Електронний ресурс]// Режим доступу: (дата звернення: 10.12.2023).

7. MAKEUP Інтернет магазин косметики та парфюмерії [Електронний ресурс] // Режим доступу: https://makeup.com.ua/ua/ (дата звернення: 10.12.2023).

8. Schwind C. Learning with personalized recommender systems: a psychological view Comput Human Behav / C. Schwind , J. Buder // Computers in Human BehaviorComputers in Human Behavior (дата звернення: 15.12.2023).

9. Hu Y., Koren Y. and Volinsky C. (2008), "Collaborative filtering for implicit feedback datasets", Data Mining, ICDM'08. Eighth IEEE International Conference on. IEEE, pp. 263–272. (дата звернення: 10.12.2023).

10. Як працює система рекомендацій контенту? [Електронний ресурс] // Режим доступу: https://www.mgid.com/uk/blog/yak-praczyuye-sistema-rekomendaczij-kontentu (дата звернення: 15.12.2023).

11. 9 способів отримати зворотний зв'язок від клієнтів на сайті [Електронний ресурс] // Режим доступу: https://web-promo.ua/ua/blog/9-sposobov-poluchit-obratnuyu-svyaz-ot-klientov-na-sajte/ (дата звернення: 15.12.2023).

12. Коефіцієнт кореляції Пірсона (r-Пірсона) [Електронний ресурс] // Режим доступу: https://www.eztests.xyz/criteria/pearsonr/ (дата звернення: 18.12.2023).

13. K-means clustering [Електронний ресурс] // Режим доступу: https://en.wikipedia.org/wiki/K-means_clustering (дата звернення: 18.12.2023).

14. Короткий посібник із розуміння алгоритму KNN [Електронний ресурс] // Режим доступу: https://www.unite.ai/uk/a-quick-guide-to-knn-algorithm// (дата звернення: 20.12.2023).

15. Bala, R., Sikka, S. and Singh, J., (2014). A Comparative Analysis of Clustering Algorithms. International Journal of Computer Applications, India, 100(15), pp. 35-39 (дата звернення: 20.12.2023).

16. Machine Learning, ML - Машинне навчання [Електронний ресурс] // Режим доступу https://www.it.ua/knowledge-base/technology-innovation/machine-learning (дата звернення: 21.12.2023).

17. Python це – [Електронний ресурс] // Режим доступу: https://uk.wikipedia.org/wiki/Python (дата звернення: 25.12.2023).

18. The Julia Programming Language [Електронний ресурс] // Режим доступу: https://julialang.org/ (дата звернення: 25.12.2023).

19. R (programming language) [Електронний ресурс] // Режим доступу: https://en.wikipedia.org/wiki/R_(programming_language) (дата звернення: 25.12.2023).

20. Mathlab [Електронний ресурс] // Режим доступу: https://dashboard.mathlab.academy(дата звернення: 28.12.2023).

21. NumPy в Python [Електронний ресурс] // Режим доступу: https://habr.com/ru/articles/352678/ (дата звернення: 30.12.2023).

22. Pandas - Python Data Analysis Library [Електронний ресурс] // Режим доступу: https://pandas.pydata.org/ (дата звернення: 30.12.2023).

23. Matplotlib [Електронний ресурс] // Режим доступу: https://ru.wikipedia.org/wiki/ Matplotlib (дата звернення: 30.12.2023).

24. Scikit-learn [Електронний ресурс] // Режим доступу: https://en.wikipedia.org/wiki/Scikit-learn (дата звернення: 30.12.2023).

25. Перспективні напрямки сучасної електроніки, інформаційних і комп'ютерних систем (MEICS-2023). Тези доповідей на VIII Всеукраїнській науково-практичній конференції: 22-24 листопада 2023 р., м. Дніпро / Укладач Іванченко О. В. – Дніпро, Дніпровський національний університет імені Олеся Гончара, ПП «Ліра ЛТД», 2023. – 262 с. ISBN 978-966-981-829-4