НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ
Завідувач кафедри
Комп'ютерних систем, мереж та кібербезпеки

Касаткін Д.Ю., к. пед.н., доц.
Підпис ПІБ, вчене звання і ступінь
«___»_____2025 р.

КВАЛІФІКАЦІЙНА БАКАЛАВРСЬКА РОБОТА

На тему: Розробка сервісів аналізу активності користувачів
Спеціальність 123 «Комп'ютерна інженерія»

Гарант освітньої програми
к.фіз.-мат.н., доцент _____ Євгеній НІКІТЕНКО
(науковий ступінь та вчене звання) (підпис) (ПІБ)

Керівник випускної бакалаврської роботи
д.т.н., професор _____ Олексій КОВАЛЕНКО
(науковий ступінь та вчене звання) (підпис) (ПІБ)

Виконав
(підпис) (ПІБ студента)

Київ – 2025

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАНН УКРАЇНИ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

«ЗАТВЕРДЖУЮ»
завідувач кафедри
комп'ютерних систем, мереж та кібербезпеки

/ Касаткін Д.Ю., к.пед.н., доц. /
підпис ПІБ, вчене звання і ступінь

р.

З А В Д А Н Н Я
ДО ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ БАКАЛАВРСЬКОЇ СТУДЕНТА
Коваль Андрій Іванович
(прізвище, ім'я, по батькові)

Спеціальність (напрям підготовки) 123 Комп'ютерна інженерія
Тема випускної бакалаврської роботи Розробка сервісів аналізу активності користувачів
керівник проекту (роботи) Коваленко О.Є., д.т.н., проф.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджено наказом ректора НУБіП України від «16» грудня 2024 р. № 2250 «С»
Термін подання завершеної роботи на кафедру 28.05.2025
(рік, місяць, число)

Вихідні дані до випускної бакалаврської роботи _____

Перелік питань, які потрібно розробити: Аналіз вимог до комп'ютерної системи спостереження, аналіз предметної області, проєктування, реалізація, тестування системи

Перелік графічних документів (за потреби)_____

Дата видачі завдання "17" грудня 2024 р
Керівник випускної бакалаврської роботи _____ / Коваленко О.Є.
(підпис) (прізвище та ініціали)
Завдання прийняв до виконання _____ / Коваль А.І
(підпис) (прізвище та ініціали студента)

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Аналіз вимог до системи	07.03.2025 р.	Виконано
2	Проєктування системи	21.03.2025 р.	Виконано
3	Реалізація системи	12.04.2025 р.	Виконано
4	Тестування розробленої системи	27.04.2025 р.	Виконано
5	Оформлення пояснювальної записки	07.05.2025 р.	Виконано
6	Оформлення графічного матеріалу	15.05.2025 р.	Виконано

Студент_____ / Коваль А.І. /

(підпис) (ініціали та прізвище)

Керівник проекту (роботи)_____ / Коваленко О.Є. /

(підпис) (ініціали та прізвище)

АНОТАЦІЯ

Розробка сервісів аналізу активності користувачів

Кваліфікаційна робота за спеціальністю 123 - «Комп'ютерна інженерія» –

Кваліфікаційна робота присвячена розробленню веб-додатку рекомендацій на основі поведінкових даних користувачів з використанням основних методів фільтрації рекомендаційних систем.

В роботі були проаналізовані системи сервісів аналізу активності користувачів , їх види, основні методи фільтрації. Також був проведений аналіз відмінностей рекомендаційних систем, їх переваги та недоліки. Далі приведена структура додатку, вибір технологій та обґрунтування їх вибору. Після, описується реалізація цієї системи.

Ключові слова: рекомендаційні системи, веб-додаток, JavaScript, Node JS, методи фільтрації, розробка, методи, Python, Express JS.

Зміст

ВСТУП 7

Розділ 1 ТЕОРЕТИЧНИЙ 8

1.1 Види систем рекомендацій 9

1.1.1 Колаборативна фільтрація: 10

1.1.3 Гібридні системи 13

1.2 Відмінності рекомендаційних систем 15

Висновки до розділу 1 18

РОЗДІЛ 2 АРХІТЕКТУРНІ РІШЕННЯ РОЗРОБКИ ДОДАТКІВ ТА АЛГОРИТМІЧНЕ ЗАБЕЗПЕЧЕННЯ 19

2.1 Фронтенд частина 19

2.2 Бекенд частина 21

2.3 МЕТОДИКИ ТА АЛГОРИТМИ ДЛЯ РЕАЛІЗАЦІЇ ЗАСТОСУНКУ 22

2.3.1 Тематична модель (Topic Modeling) 22

2.3.5 Косинусна схожість між користувачами в колаборативній фільтрації: 27

2.3.5 Косинусна схожість між користувачем і товаром в контент-орієнтованій фільтрації .. 29

Висновки до розділу 2 30

РОЗДІЛ 3 РОЗРОБКА ВЕБ-ДОДАТКУ 31

3.1 Мови програмування та технології для реалізації 31

3.1.1 Javascript 31

3.1.2 TypeScript 31

3.1.3 Node.js 32

3.1.4 Express Js 33

3.1.5 Python 34

3.2 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ВЕБ-ДОДАТКУ 35

3.2.1 Перевірка колаборативної фільтрації 40

3.2.2 Перевірка фільтрації на основі контенту 42

3.2.3 Перевірка фільтрації гібридної фільтрації 45

Висновки до розділу 3 46

ВИСНОВКИ 47

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 49

Додаток А – програмний код 51

ВСТУП

Актуальність теми. В сучасному світі, де технології та Інтернет постійно розвиваються, веб-додатки займають важливе місце у повсякденному житті, особливо в аспектах, як особистісного зростання, так і зручності у щоденних справах. Онлайн магазини, як приклад, економлять час, дозволяючи робити покупки не виходячи з дому. Розвиток веб-технологій надає компаніям великі переваги, дозволяючи їм займати провідні позиції на ринку завдяки створенню якісних веб-ресурсів.

Швидке завантаження веб-сторінок є критично важливим для успіху сайту, адже користувачі звикли до миттєвого доступу до інформації. Тому оптимізація веб-сторінок стає пріоритетом. Сучасні веб-розробки та стратегії конкуренції, включаючи алгоритми рекомендацій товарів, є важливими для забезпечення унікального користувацького досвіду. Ці системи аналізують взаємодію користувачів з платформою, пропонуючи товари, які відповідають їхнім індивідуальним інтересам.

Професійно створені веб-сайти здатні миттєво збирати та обробляти дані, дозволяючи онлайн-платформам швидко адаптуватися до змін у поведінці користувачів. Вони впроваджують технології штучного інтелекту та машинного навчання для оптимізації процесів рекомендацій, що підвищує точність та результативність рекомендацій.

Цей підхід не тільки стимулює ріст продажів через персоналізовані пропозиції, але й зміцнює відданість клієнтів, спонукаючи їх до повторних відвідин платформи. Така стратегія підвищує конкурентоспроможність, пропонуючи унікальний індивідуальний досвід кожному користувачу.

Мета роботи. У рамках кваліфікаційної роботи, фокус ставиться на створенні сучасного веб-додатку з використанням передових

фронтендтехнологій та методика проектування, а також використання основних методів фільтрації для рекомендації товарів, які відповідають перевагам користувача.

Завдання роботи. Проект спрямований на створення удосконаленої версії алгоритму рекомендації, заснованого на сингулярному розкладі матриць, та конструювання передового веб-додатка із застосуванням Angular фреймворк.

Це має підсилити комерційний потенціал магазину.

Об’єкт та предмет дослідження. Об'єктом дослідження є веб-додаток електронної комерції. Предметом є розробка алгоритму рекомендацій, спрямованого на підвищення ефективності та задоволеності користувачів.

Методи та засоби дослідження. Аналіз актуальних літературних джерел та дослідженні передових практик у галузі веб-розробки. Основна увага приділяється вивченню архітектури сучасних веб-додатків, з особливим акцентом на розробку клієнтської частини та ефективність моделей даних. Це включає детальне дослідження логіки створення інтерфейсів користувача, а також аналіз способів збору, оброблення та використання даних для оптимізації функціональності веб-додатків.

Наукова новизна та практичне значення отриманих результатів.

Наукова новизна виявляється у розробці вдосконаленого алгоритму рекомендацій, що враховує сучасні тенденції веб-додатків та користувацьких даних. Практичне значення полягає у підвищенні ефективності онлайнмагазинів та забезпеченні більш персоналізованого досвіду для користувачів, що може бути використане в різних бізнес-сферах.

Результати магістерської роботи були апробовані в наукової статті: Розробка веб-сервісу рекомендацій на основі аналізу поведінкових даних користувачів / Резанова, В. Г., Курочка В.О /

https://er.knutd.edu.ua/handle/123456789/24826

РОЗДІЛ 1 ТЕОРЕТИЧНИЙ

Розвиток рекомендаційних систем має довгу історію, яка почалася ще до появи Інтернету, але значно активізувалася з розвитком цифрових технологій. Давайте розглянемо ключові моменти у розвитку цих систем.

Спочатку рекомендаційні системи зустрічалися в бібліотеках, де бібліотекарі пропонували книги читачам на основі їхніх інтересів. Ці ранні системи використовували базові алгоритми, що орієнтувалися на категорії та жанри книг.

У 1990-х, з розширенням Інтернету, почався аналіз контенту для рекомендацій. Системи почали аналізувати характеристики об'єктів, як-от ключові слова та авторів, для виявлення та порівняння схожих об'єктів.

У кінці 1990-х з'явилися алгоритми колаборативної фільтрації. Вони аналізували взаємодію користувачів з об'єктами, такими як оцінки чи покупки, для виявлення подібностей між користувачами або об'єктами і рекомендували нові об'єкти, які сподобалися схожим користувачам.

З розвитком Інтернету та соціальних мереж, рекомендаційні системи стали більш індивідуалізованими. Вони використовують дані про інтереси, поведінку та демографічні особливості користувачів для підвищення точності рекомендацій.

Сучасні системи часто використовують гібридні підходи, комбінуючи колаборативну фільтрацію, контент-фільтрацію та інші методи, для забезпечення більш точних та різноманітних рекомендацій.

1.1 ВИДИ СИСТЕМ РЕКОМЕНДАЦІЙ

У сфері електронної комерції існують різні методи рекомендацій товарів, кожен з яких спрямований на створення індивідуалізованого досвіду для користувачів. Давайте детально розглянемо основні типи цих алгоритмів:

1.1.1 Колаборативна фільтрація:

Цей алгоритм працює шляхом збору та аналізу інформації про переваги користувачів. Він виявляє схожості між користувачами або між продуктами на основі їхніх покупок або оцінок. На підставі цього, система може рекомендувати продукти, які користувалися популярністю серед схожих користувачів або які є схожими на ті, які користувач вже оцінив позитивно. Принцип цієї фільтрації можна побачити на рисунку 1.1

Рис. 1.1 Принцип Колаборативної фільтрації

Ці алгоритми ґрунтуються на принципі "подібність приваблює" [4]. Вони дозволяють виявити схожі переваги між користувачами або між продуктами для створення рекомендацій. Важливо враховувати потенційні проблеми, такі як

"холодний старт" (коли система має обмежену інформацію про нових користувачів або продукти) і масштабованість при великій кількості даних.

Серед алгоритмів колаборативної фільтрації можна виділити:

1) На основі користувача колаборативна фільтрація:

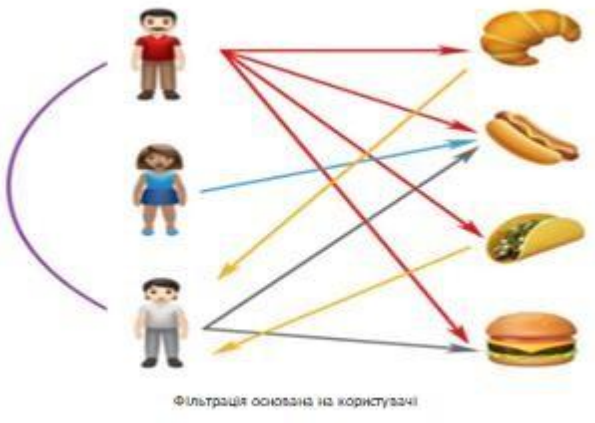

Рисунок 1.2 Колаборативна фільтрація на основі користувача

Цей підхід використовує аналіз схожості між користувачами для створення рекомендацій. Система аналізує взаємодію користувачів із продуктами (наприклад, оцінки) і шукає користувачів зі схожими смаками або перевагами. Потім вона ідентифікує продукти, які сподобалися цим схожим користувачам, але ще не були оцінені цільовим користувачем, і рекомендує їх. Принцип цієї методики можна побачити на рисунку 1.2

2) На основі товарів колаборативна фільтрація

Цей підхід зосереджений на аналізі схожості між самими об'єктами (товарами). Система досліджує, як користувачі взаємодіють з продуктами, і на основі цього визначає схожість між товарами, засновану на перевагах користувачів.

Головна відмінність цього методу від на основі користувачів колаборативної фільтрації полягає в акценті на даних про товари, а не користувачів. Система використовує ці дані для рекомендації інших продуктів, які схожі на ті, які вже сподобалися користувачу.

Цей метод дозволяє системі рекомендацій запропонувати продукти, які можуть бути цікаві користувачу, засновуючись на його попередніх виборах та вподобаннях, що допомагає підвищити якість та точність рекомендацій. Візуальне представлення цього принципу можна побачити на рисунку 1.3.

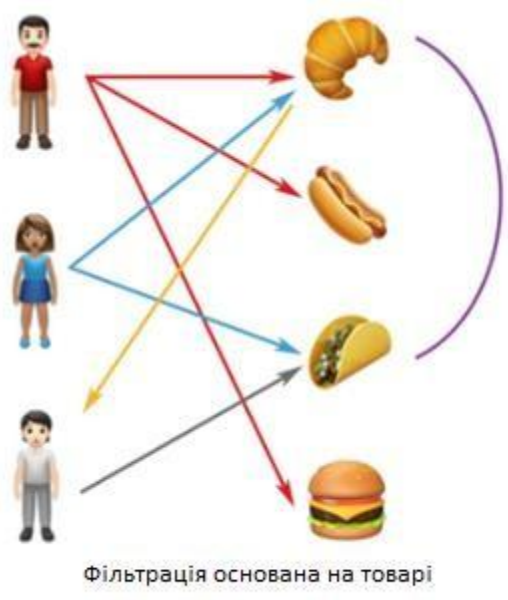

Рисунок 1.3 Колаборативна фільтрація на основі товарів

3) Гібридна колаборативна фільтрація

Гібридний метод колаборативної фільтрації інтегрує в собі кілька підходів до аналізу даних, що стосуються переваг користувачів і характеристик предметів. Він гармонійно об'єднує елементи користувацько-орієнтованої та об'єктно-орієнтованої фільтрації, доповнюючи їх іншими техніками підбору рекомендацій. Така багатогранна система, представлена на рис. 1.4, використовує вагові коефіцієнти для калібрування рекомендацій, спрямованих на забезпечення високої точності та індивідуалізації пропозицій користувачам.

Рисунок 1.4 – Принцип роботи гібридної фільтрації

1.1.2 Заснований на контенті (Content-based)

Підхід, що базується на аналізі вмісту товарів, використовує детальний розгляд атрибутів продуктів для створення рекомендацій. Через ретельний аналіз описів продуктів, ключових слів та інших важливих характеристик, алгоритм виявляє подібності між різними предметами. Заснований на виявленій схожості, алгоритм надає користувачам пропозиції товарів, що відповідають раніше виявленим уподобанням. Ця методологія фокусується на

атрибутах об'єктів продажу, таких як ключові слова, описи, жанри, та інші метадані, щоб запропонувати рекомендації. Цей механізм створює індивідуальний профіль для кожного користувача, використовуючи дані про їх попередні вподобання та взаємодії з продуктами, і на цій основі пропонує подібні товари. Графічне представлення цієї концепції можна побачити на рис 1.5.

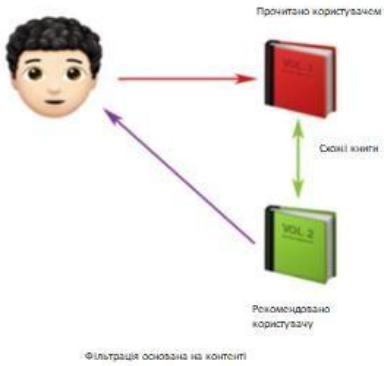

Рисунок 1.5 – Принцип роботи заснований на контенті

1.1.3 Гібридні системи

Гібридизація систем рекомендацій означає інтеграцію різноманітних методологій та алгоритмів для створення більш досконалих та індивідуалізованих пропозицій для користувачів. Використовуючи силу комбінованих стратегій, такі системи здобувають широке визнання у сфері цифрової комерції та інтернет-сервісів.

Сутність гібридних систем полягає в тому, що вони аплікують мікс різних моделей рекомендацій — від колаборативної фільтрації до контент-базованих алгоритмів, а також розглядають популярність, демографічні дані та інші критерії. Це дозволяє системі адаптуватися до різноманітних характеристик та уподобань користувачів.

Гібридні системи рекомендацій здійснюють інтеграцію різноманітних джерел та технік для надання точних індивідуалізованих пропозицій. Вони використовують синергію між спільними та унікальними властивостями різних алгоритмів, наприклад, агрегуючи інсайти з колаборативної фільтрації та контентно-орієнтованих рекомендацій.

Інноваційний підхід гібридних систем полягає у поєднанні прогнозів з множини алгоритмів, утворюючи комплексні рекомендації. Моделі машинного навчання можуть вносити свій вклад у прогнозування вподобань, які потім ансамблюються з традиційними методами.

Контекстуальна інформація, така як місцезнаходження користувача, час доби, а також поведінкові дані, інтегруються для тонкої настройки рекомендацій, забезпечуючи релевантність у конкретних ситуаціях. Використання цих даних дозволяє системі динамічно адаптуватися до змін у потребах та перевагах користувачів.

Ключовою перевагою гібридних систем є їх універсальність і здатність до надання високоякісних рекомендацій, які враховують як історію взаємодій, так і актуальні контекстуальні сигнали. Однак, вони вимагають ретельного аналізу та налаштування, а також мають високі вимоги до ресурсів для розробки та обслуговування.

Загалом, гібридні системи рекомендацій пропонують глибоку адаптацію до індивідуальних потреб користувачів, ефективно використовуючи дані для підвищення задоволеності клієнтів та комерційної ефективності. Розробка таких систем є складним процесом, але вони мають потенціал забезпечити значну конкурентну перевагу на ринку.

1.2 Відмінності рекомендаційних систем

Порівняючи різні системи рекомендацій, можна виділити унікальні особливості, переваги та недоліки кожної з них. 1) Колаборативна фільтрація Переваги:

Відкритість рекомендацій: Ця система здатна рекомендувати товари, які не мають детального опису чи контенту, виходячи з схожості між користувачами або товарами, базуючись на їхній історії взаємодії.

Виявлення складних зв'язків: Здатна ідентифікувати складні патерни взаємодій між користувачами та товарами, що можуть залишитися непоміченими для систем, заснованих на контенті. Використання існуючих даних: Колаборативна фільтрація оперує наявними даними про взаємодії користувачів з товарами, тому не потребує додаткової інформації про характеристики або контент об'єктів.

Недоліки колаборативної фільтрації:

Проблема "холодного старту": Може виникнути проблема при наданні рекомендацій новим користувачам або для нових товарів через відсутність історії взаємодії.

Проблема розрідженості матриці: У великих системах матриця взаємодій може бути неповною, що ускладнює аналіз та знижує точність рекомендацій.

Ризик обмеження рекомендацій: Система може схилятися до рекомендації вузького спектра товарів, базуючись лише на попередніх взаємодіях користувачів, не враховуючи нові можливості чи інтереси.

2) Плюси та мінуси алгоритмів на основі контенту

Плюси алгоритмів, заснованих на контенті:

- Урахування особистих переваг: Ці алгоритми здатні адаптуватися до унікальних вподобань користувачів, аналізуючи атрибути та характеристики об'єктів.
- Мінімізація проблеми холодного старту: Здатні надавати рекомендації новим користувачам або для нових об'єктів, оскільки вони фокусуються на характеристиках самих об'єктів.
- Ефективність у системах з обмеженими даними: Особливо корисні в системах, де кількість даних обмежена.

Мінуси алгоритмів, заснованих на контенті:

- Обмеження виявлення складних зв'язків: Можуть не виявити складніші взаємозв'язки між користувачами та об'єктами, оскільки вони зосереджені лише на характеристиках об'єктів.
- Проблема врахування нових тенденцій: Можуть бути менш адаптивними до змін в уподобаннях користувачів або нових тенденцій, оскільки вони орієнтовані на статичні характеристики об'єктів.

4) Плюси та мінуси гібридних систем

Переваги гібридних систем рекомендацій товарів:

- Краща точність рекомендацій: Гібридні системи, поєднуючи різні методи, забезпечують більш точні рекомендації, використовуючи дані з колаборативної фільтрації, контенту, демографії тощо.
- Широкий спектр рекомендацій: Можуть пропонувати рекомендації, засновані не лише на схожості користувачів або

об'єктів, а й на інших факторах, розширюючи можливості для врахування різних аспектів та потреб користувачів.

Недоліки гібридних систем рекомендацій товарів:

- Складність інтеграції: Комбінування різних підходів може бути складним та вимагати значних ресурсів для налаштування та підтримки.
- Вибір та налаштування алгоритмів: Потребує витонченого підбору та налаштування алгоритмів для оптимальної ефективності, що може бути складним завданням.
- Обмеження в широті та глибині використання даних: Хоча гібридні системи поєднують різні методи, вони все одно можуть мати обмеження у використанні великої кількості даних та їх різноманітності.
- Проблема холодного старту у гібридних системах рекомендацій: Гібридні системи можуть зіткнутися з проблемою недостатності даних про нових користувачів або продукти для забезпечення точних рекомендацій. Для вирішення цієї проблеми можна використовувати альтернативні стратегії, такі як рекомендації на основі популярності або контексту.
- Вимоги до обчислювальних ресурсів: Гібридні системи потребують значних обчислювальних ресурсів через використання комбінованих алгоритмів та методів. Обчислювальна складність може стати викликом у випадку обробки великих обсягів даних або в умовах обмеженого доступу до ресурсів.

Загальний аналіз гібридних систем: Незважаючи на зазначені вище недоліки, гібридні системи рекомендацій мають значні переваги, які часто переважають

їх обмеження. Вони забезпечують точніші та більш персоналізовані рекомендації, а також ефективно враховують різноманітні аспекти поведінки та потреб користувачів. Хоча розробка та підтримка таких систем може вимагати значних зусиль та ресурсів, вони здатні принести значну цінність, покращуючи загальний досвід користувачів.

Висновки до розділу

Колаборативна фільтрація відрізняється здатністю до рекомендацій, що не вимагають детальної інформації про об'єкти, але стикається з проблемами холодного старту та обмеженістю в охопленні різноманітних об'єктів. Це робить її ефективною для систем із великою кількістю історичних даних, але менш відповідною для нових або менш популярних об'єктів.

Алгоритми, засновані на контенті, надають перевагу індивідуальним вподобанням користувачів, але можуть бути обмежені у виявленні складних взаємозв'язків між об'єктами. Вони ефективні у ситуаціях, де доступні детальні описи об'єктів, але можуть бути недостатні для врахування змін у перевагах користувачів.

Гібридні системи об'єднують кращі аспекти різних підходів, пропонуючи точні та персоналізовані рекомендації, але водночас потребують складнішої розробки та підтримки. Вони здатні адаптуватися до різних контекстів та враховувати різноманітні потреби користувачів.

У цілому, кожен із розглянутих підходів має свої переваги та обмеження, що робить їх відповідними для різних сценаріїв використання. Вибір конкретного підходу чи комбінації підходів залежить від специфіки задачі, доступних даних та обчислювальних ресурсів.

2 МЕТОДИ ТА ЗАСОБИ ДОСЛІДЖЕННЯ АКТИВНОСТІ КОРИСТУВАЧІВ ВЕБСАЙТІВ

2.1. Методи моніторингу вебактивності користувача

Журнали сервера. Один з найпростіших методів моніторингу активності користувачів на вебсайті – це журнали сервера. Цей метод записує інформацію про запити користувачів, таку як IP-адреса, тип браузера та відвідані сторінки. Основна перевага використання логів сервера полягає в тому, що вони надають історичний запис активності користувача, який може бути корисним для усунення несправностей і виявлення моделей поведінки. Однак, журнали сервера можуть бути складними для аналізу і не надавати детальної інформації про поведінку користувача [4].

Відстеження файлів cookie. Ще одним поширеним методом моніторингу активності користувачів є використання файлів cookie для відстеження. Вебсайти можуть використовувати файли cookie для відстеження активності користувачів протягом декількох сеансів, що може надати інформацію про те, як часто користувачі відвідують сайт і які сторінки є найбільш популярними. Основна перевага використання файлів cookie полягає в тому, що вони можуть надати більш детальну картину поведінки користувача, ніж журнали сервера. Однак файли cookie також можуть викликати занепокоєння щодо конфіденційності та можуть не відповідати певним нормам [4].

Інструменти вебаналітики. Використання інструментів вебаналітики, таких як Google Analytics, може бути потужним способом моніторингу активності користувачів на вебсайті. Ці інструменти можуть надавати детальну інформацію про поведінку користувачів, таку як кількість відвідувачів, їхнє місцезнаходження та сторінки, які вони відвідують. Основна перевага

інструментів вебаналітики полягає в тому, що вони можуть надавати дієву інформацію, яка може допомогти власникам вебсайтів покращити ефективність їхніх сайтів. Однак ці інструменти можуть бути складними в налаштуванні і вимагати певного рівня технічних знань.

Теплові карти та запис сеансів (hotjar). Теплові карти та інструменти запису сеансів можна використовувати, щоб зрозуміти, як користувачі взаємодіють з вебсайтом. Ці інструменти можуть показати, де користувачі натискають, прокручують і наводять курсор на вебсайт, що може допомогти виявити проблеми з юзабіліті і больові точки. Основна перевага теплових карт і запису сеансів полягає в тому, що вони можуть надати візуальне уявлення про поведінку користувачів, яке може бути легше зрозуміти, ніж необроблені дані. Однак ці інструменти також можуть бути ресурсомісткими і вимагати додаткового обладнання та програмного забезпечення [4].

A/B-тестування. A/B-тестування можна використовувати для моніторингу активності користувачів, порівнюючи поведінку користувачів на різних версіях вебсайту. Це може допомогти власникам вебсайтів зрозуміти, які елементи дизайну є найбільш ефективними для залучення користувачів. Основна перевага A/B-тестування полягає в тому, що воно може надати прямі докази того, які зміни призводять до покращення продуктивності. Однак A/B-тестування вимагає значного обсягу трафіку і може зайняти багато часу для налаштування та аналізу [4].

Відстеження подій на стороні сервера. Відстеження подій на стороні сервера - це процес відстеження взаємодій користувачів і надсилання їх на сервер для аналізу. Цей метод може бути більш точним і надійним, ніж відстеження подій на стороні клієнта. Основна перевага відстеження подій на стороні сервера

полягає в тому, що воно може надавати більш точні дані і захищати конфіденційну інформацію від потрапляння до клієнта. Однак відстеження подій на стороні сервера вимагає складнішого налаштування і може мати більшу затримку [4].

2.2. Метрики аналізу вебактивності користувача

Метрика кількості унікальних відвідувачів вашого сайту – це метрика, яка використовується для вимірювання загальної кількості унікальних відвідувачів вашого сайту за певний період часу. Ця метрика важлива для розуміння загальної кількості користувачів, які взаємодіють з вашим сайтом. Крім того, кількість унікальних відвідувачів вашого сайту може використовуватися для вимірювання ефективності маркетингових кампаній і загальної продуктивності вашого сайту [4].

Показник кількості переглядів – це показник, який використовується для вимірювання загальної кількості переглядів сторінки вебсайту за певний період часу. Крім того, кількість переглядів може бути використана для визначення популярного контенту на вебсайті та відстеження ефективності з плином часу [4].

Показники зворотних посилань (беклінки) – вимірюють кількість і якість вхідних посилань на сайт або вебсторінку, а також джерела посилань. Зворотні посилання важливі для SEO, оскільки вони вказують пошуковим системам на те, що вебсайт є релевантним, надійним і авторитетним. Загальні показники зворотних посилань включають загальну кількість зворотних посилань, кількість унікальних доменів, що посилаються на вебсторінку, кількість посилаючих доменів, кількість зовнішніх і внутрішніх посилань, показник авторитетності сторінки та показник авторитетності домену. Крім того, зворотні посилання

можуть бути використані для відстеження ефективності кампаній з нарощування посилань. [4].

Показники джерел трафіку – вимірюють походження відвідувачів вебсайту, таких як пошукові системи, реферальні сайти, соціальні мережі та прямі відвідування. Відстеження джерел трафіку може допомогти вам визначити найбільш ефективні маркетингові канали, оптимізувати ваші кампанії та приймати обґрунтовані рішення про те, на чому зосередити свої зусилля. Загальні показники джерел трафіку включають кількість відвідувань з кожного джерела, відсоток загального трафіку з кожного джерела та середній час перебування на сайті для кожного джерела [4].

Показник "Нові відвідувачі проти постійних відвідувачів" використовується для вимірювання кількості відвідувачів, які вперше відвідали вебсайт, порівняно з кількістю відвідувачів, які повернулися на нього. Цей показник важливий для розуміння ефективності маркетингових кампаній та загальної ефективності вебсайту. Співвідношення нових відвідувачів до відвідувачів, що повертаються, може бути використано для вимірювання успіху вебсайту в утриманні відвідувачів. Крім того, цей показник може бути використаний для визначення сфер, що потребують покращення, та оптимізації вебсайту для підвищення його ефективності [4].

Показник користувачів, які діляться вашим контентом з іншими, вимірює кількість користувачів, які поділилися вашим контентом у соціальних мережах, на вебсайтах та інших платформах. Цей показник важливий для розуміння успішності вашої стратегії контент-маркетингу, а також рівня взаємодії з вашим контентом. Крім того, цей показник можна використовувати для визначення

популярного контенту, відстеження ефективності з плином часу та вимірювання ефективності ваших контент-маркетингових кампаній.

Показник кількості коментарів вимірює загальну кількість коментарів, які були розміщені на вебсайті або вебсторінці [4].

Показник клікабельності вимірює, наскільки легко натискається посилання або кнопка. Цей показник важливий для розуміння користувацького досвіду на вашому вебсайті, а також загальної продуктивності вашого вебсайту. Крім того, цей показник може бути використаний для визначення областей для поліпшення та оптимізації вебсайту для кращої продуктивності. Показник клікабельності зазвичай вимірюється з урахуванням таких факторів, як розмір і колір посилання або кнопки, кількість пробілів навколо неї, текст, що використовується в посиланні або кнопці, і позиція посилання або кнопки [4].

Показник кількості лідів (користувачів) вимірює загальну кількість потенційних клієнтів, які були ідентифіковані для вебсайту або бізнесу. Цей показник важливий для розуміння ефективності маркетингових кампаній та загальної ефективності бізнесу. Крім того, цей показник може бути використаний для вимірювання успіху вебсайту в перетворенні відвідувачів на потенційних клієнтів, відстеження ефективності з плином часу та визначення сфер для вдосконалення [4].

Показник економічного ефекту (ROI) вимірює рентабельність інвестицій в бізнес або вебсайт. Цей показник важливий для розуміння фінансових результатів бізнесу, а також ефективності маркетингових кампаній. Крім того, цей показник може бути використаний для вимірювання успіху вебсайту в отриманні доходу, відстеження продуктивності в часі та визначення областей для поліпшення [4].

2.3. Аналіз поведінки користувачів сайту засобами та інструментами вебаналітики

Аналіз дій відвідувачів вебсайту або, як ще кажуть вебаналітики, аналіз поведінки відвідувачів – є ключовим елементом оцінки ефективності сайту і прогнозування його діяльності на перспективу. Ще зовсім недавно проблема аналізу дій відвідувачів на вебсайті асоціювалася насамперед із даними, що надаються сервісом Google Analytics [9].

Нині ще досить багато власників сайтів активно використовують цю програму для відстеження ефективності своїх вебресурсів. Але поступово на ринок вебаналітики виходять більш сучасні системи, що дають змогу не тільки збирати статистику відвідуваності сайтів, а й аналізувати поведінку відвідувачів.

Якщо розглянути цю проблематику більш детально, то стає очевидним, що дії відвідувачів вебсайту і поведінка відвідувачів сайту – категорії дещо відмінні одна від одної. Відвідувач сайту може не здійснювати якихось конкретних дій (перехід вебсторінками, введення запитів у пошукову систему сайту, кліки на сторонні посилання тощо). У такому разі Google Analytics не надасть жодних даних про те, що відвідувач робить на сайті. Але якщо він не робить ніяких дій, це ще не означає, що він не взаємодіє з сайтом. Адже відвідувачі, перебуваючи на сайті, можуть просто переглядати контент, здійснювати рухи мишкою, не роблячи кліків на посилання або інші елементи сайту. Тобто певний алгоритм поведінки відвідувачів сайту існує в будь-якому разі [9].

Якщо говорити про методику збору даних про поведінку відвідувачів на вебсайті, то ідеальним рішенням буде відеозапис руху миші, теплові карти кліків, скролінгу і фіксації уваги відвідувачів на вебсторінці.

Наприклад, відвідувачі можуть просто водити курсором мишки по сторінці, не виділяючи нічого і не здійснюючи кліків. Відеозапис руху мишки покаже всю траєкторію переміщень курсору на вебсторінці, а теплова карта фіксації уваги – зони, які проглядаються найбільш інтенсивно. Таким чином, можна з'ясувати, які зони на сайті проглядаються відвідувачами найчастіше, а які взагалі не проглядаються, визначити непопулярні сторінки сайту, тобто побачити реальний ступінь зацікавленості відвідувачів змістом сайту. Виходячи з отриманих даних, розробляється комплекс заходів щодо оптимізації вебсайту. В ідеалі, ця процедура проводиться шляхом багатопараметричної перевірки "проблемних" елементів сайту.

Наприклад, якщо відвідувачі ігнорують контент на одній зі сторінок вебсайту, то копірайтер готує два-три нових варіанти контенту, і кожен з них тестується на предмет читабельності відвідувачами сайту. Перевірка проводиться шляхом використання систем вебаналітики. Той варіант контенту, який найчастіше стали переглядати відвідувачі, затверджується як остаточний. Завжди варто пам'ятати, що аналіз діяльності сайту – це головний інструмент його прогресу [9].

На практиці процес відстеження поведінки відвідувачів на вебсайті за допомогою відеозапису руху курсору мишки є досить трудомістким, тому під час використання зазначеного інструменту необхідно враховувати деякі аспекти.

На сьогоднішній день відеозапис руху мишки відвідувача на вебсайті є інструментом, що показує найбільш деталізовану картину дій відвідувача на сайті. Крім нього майже аналогічну картину показує система «eye tracking». Ці два інструменти взаємодоповнюють один одного. Адже з одного боку відвідувач може дивитися на сайт, не рухаючи мишкою. З іншого – відвідувач не може

натиснути на посилання поглядом, тому відстеження траєкторії руху мишки – важливий аспект під час проведення аналізу сайту. Водночас, як засвідчила практика, аналіз руху мишки не тільки вельми корисний, а й трудомісткий процес.

Аналіз відеозапису руху курсору мишки проводять у два етапи: первинний аналіз і вторинний (деталізований) аналіз. Первинний аналіз рекомендується проводити власникам сайтів, які не мають багато вільного часу або не знайомі з нюансами вебаналітики. Вторинний або деталізований аналіз - це скоріше сфера діяльності професійних вебаналітиків, які мають необхідний досвід роботи з аналізу сайтів.

Жодна з наявних систем вебаналітики не скаже, які реально емоції відчуває відвідувач, який прийшов на ваш вебсайт. Це можна визначити лише побічно шляхом аналізу поведінки відвідувача на вебсайті. Однак навіть у такому разі немає гарантії, що вебаналітик не зробить помилки в процесі інтерпретації даних про дії відвідувачів на вебсайті [4].

2.4. Аналіз активності користувачів (підписників) вебресурсу

Відслідковування активності користувачів (підписників сайту) дозволяє відстежити активність підписника в кількості відправлених йому розсилок, прочитаних і непрочитаних, кількості переходів за посиланнями.

Після запуску розсилки ви зможете аналізувати і покращувати маркетингові рішення (компанії) сайту за допомогою даних активності користувачів.

Аналіз активності користувачів забезпечує:

1) розуміння шаблонів поведінки користувачів
2) зменшення рівня занепокоєння клієнтів
3) збільшення тривалості життя клієнтів;

4) запобігання вигоранню бази клієнтів;
5) ефективне використання акційних пропозицій.

За способом надсилання розсилки можна розділити на 2 види:

1) Масові розсилки:

Відправлення: одноразово під певну подію на групу контактів.

Приклад використання: надсилання інформації про розпродаж.

Оцінка результату: розсилка вважається завершеною, щойно її надіслано заданій кількості осіб. Рекомендується дивитися звіт щонайменше за три дні після відправлення розсилки, оскільки більшість активних користувачів до цього часу відкриють і прочитають лист.

2) Тригерні розсилки:

Відправлення: автоматичні розсилки, які надсилаються після того, як користувач виконає необхідну для запуску дію.

Приклад використання: нагадування про забуті в кошику товари.

Оцінка результату: звіт перебудовується після кожного запуску листа, відправлення здійснюється, поки ми не зупинимо розсилку самостійно. На момент оцінки розсилка є активною.

Етапи дослідження:

1. Підготовка даних:

Змоделюємо простий набір даних, який включає:

ContactID - id користувача

Date - час отримання листа

Response - мітка конверсії (0 - ні, 1 - так)

У розглянутому наборі даних будуть користувачі, які: зовсім не читають листи, читають рідко, читають майже все. Врахуємо також, що деякі з

користувачів читають листи з певною періодичністю, а деякі - від випадку до випадку.

Приклад вихідного (тестового) набору даних (було проаналізовано 10 000 розсилок) наведено в таблиці 2.1.

Таблиця 2.1 – Приклад вихідного набору даних

	ContactID	Date	Response
1	32	2022-05-29 22:22:49	0
2	2	2022-06-14 18:35:40	0
3	54	2022-07-10 17:31:10	1
4	12	2022-10-04 14:49:50	0
5	60	2022-03-17 03:59:05	0
6	82	2022-08-18 08:30:45	0
7	79	2022-07-24 07:39:52	1
8	23	2022-09-14 01:20:03	0
9	134	2022-10-15 00:25:07	1
10	120	2022-10-01 16:14:43	0
…	…	…	…
10000	131	2022-11-07 04:08:03	1

Порахуємо для кожного користувача, на момент отримання чергового листа в історії, такі показники:

1) кількість отриманих/прочитаних листів;
2) кількість отриманих/прочитаних листів у будні;
3) кількість отриманих/прочитаних листів у вихідні;

4) відсоток відкриття листів;

5) відсоток відкриття листів по буднях/вихідних;

6) дата останнього отримання листа;

7) дата останнього прочитання листа;

8) час від останнього отриманого листа;

9) час від останнього прочитання листа;

10) нормовані показники.

– середня кількість листів, які отримує користувач на тиждень; – середня кількість листів, що відкриваються користувачем на тиждень.

11) бінарні показники активності за обрані періоди (0 - ні, 1 - так):

– активні протягом останніх 5 днів;

– активні протягом останніх 10 днів; – активні протягом останніх 15 днів.

Під прочитаними листами розуміються такі, на які було отримано відповідь адміністратору сайту.

2. Побудова моделі:

Необхідно побудувати модель передбачення значення змінної відгуку у за заданого значення вхідних параметрів $x_i \in X$, $i = \{1,...,n\}$.

Побудована модель має відповісти на запитання: прочитає користувач черговий лист чи ні? Якщо прочитає - відправляємо лист. Якщо ні - очікуємо від моделі сигналу до відправлення.

Поставлене завдання зводиться до розв'язання задачі бінарної класифікації – змінна – атрибут класу приймає 2 значення 0 або 1 (де 0 - ні, не прочитає (відповість); 1 - так, прочитає).

Як класифікаційні атрибути використано сукупність перерахованих вище показників (п. 1).

Як метод класифікації використовували наївний байєсівський класифікатор.

Для побудови моделі було використано мову R:

```
#заванажуємо бібліотеку нелінійних класифікаторів
library(klaR)
#імпорт даних; змінна y – атрибут класу
input_data <- read.table(file = "d:/in.txt", sep = ' ')
naive_classif <- NaiveBayes(input_data$Species ~ ., data = input_data)
naive_classif
```

Результати класифікації:

```
$apriori
grouping
    No  Yes
 0.6666666  0.3333333
```

Як можна бачити:

– 2/3 прикладів класифікована як No (не прочитає листа); –

1/3 прикладів класифікована як Yes (прочитає лист).

3. Оцінка якості побудованої моделі.

Точність класифікації:

```
acc <- mean(pred == input_data$y)
paste("Accuracy=", round(100*acc, 2), "%", sep = "")
```

Виведення результату:

```
Accuracy= 96%
```

Побудовану модель можна використовувати для визначення цільової аудиторії користувачів вебресурсу з якими можна активно працювати та заохочувати до акційних пропозицій через засоби email розсилки, а також для прогнозування активності користувачів вебресурсу.

2.5. Висновки до розділу 2

Побудовано модель для визначення цільової аудиторії користувачів вебресурсу яких можна заохочувати до пропозицій через розсилки email.

Проаналізовано методи і засоби для збору даних про активність користувачів за допомогою інструментів вебаналітики, таких як Google Analytics, для відстеження переглядів сторінок, показників відмов та інших метрик.

Розглянуто метрики аналізу вебактивності користувачів. Виявлено, що такі показники, як кількість переглядів сторінок, час перебування на сайті та показник відмов, можуть надати цінну інформацію про поведінку користувачів на вебсайті. Визначено, що більш просунуті показники, такі як теплові карти і карти прокрутки, можуть забезпечити більш глибоке розуміння залучення користувачів і взаємодії з вебсайтом.

На основі проведеного аналізу визначено, що використання декількох метрик у поєднанні може забезпечити більш повне розуміння активності користувачів. Завдяки використанню цих різних методів отримано всебічне розуміння поведінки користувачів на вебсайті, і ця інформація була використана для покращення користувацького досвіду.

Проведено аналіз активності підписників вебресурсу. Побудовано класифікатор, що дозволяє виявляти підписників, що є активними відносно відповіді на e-mail розсилку. В якості класифікаційниї атрибутів використано такі показники, як: кількість отриманих/прочитаних листів; дата останнього отримання листа; дата останнього прочитання листа та інші. В якості атрибуту класу розглядалося 2 значення : 0 - не відповість на лист; 1 – прочитає листа.

РОЗДІЛ 3 РОЗРОБКА ВЕБ-ДОДАТКУ

3.1 Мови програмування та технології для реалізації

3.1.1 Javascript

Зважаючи на універсальність JavaScript, варто відзначити, що ця мова програмування є стандартом для розробки веб-додатків. Вона підтримується всіма сучасними браузерами, що робить її ідеальним вибором для створення програм, які працюють на будь-якому пристрої з доступом до мережі Інтернет.

JavaScript використовується як на клієнтській, так і на серверній стороні розробки. На клієнтській стороні вона дозволяє створювати інтерактивні елементи веб-сторінок та забезпечує можливість взаємодії з користувачем на браузері. На серверній стороні, завдяки платформі, такій як Node.js, JavaScript використовується для створення потужних та масштабованих серверних додатків.

Крім того, JavaScript має багату екосистему з численними бібліотеками та фреймворками. Ці інструменти допомагають веб-розробникам прискорити розробку та покращити функціональність веб-застосунків. Наприклад, фреймворки, такі як React і Vue.js, роблять можливим створення інтерактивних веб-додатків, а Angular надає інструменти для розробки складних веб-проектів.

Загалом, ці характеристики роблять JavaScript популярним вибором для веб-розробки і надають йому широкий спектр можливостей.

3.1.2 TypeScript

TypeScript: Статично типізована мова програмування

TypeScript - це мова програмування, яка базується на JavaScript і надає можливість визначати типи даних для змінних, параметрів функцій і об'єктів. Це допомагає зробити ваш код більш надійним та легше підтримуваним.

Основні переваги TypeScript:

Статична типізація: TypeScript дозволяє визначити типи для змінних, що допомагає виявляти помилки на етапі розробки і полегшує роботу з великими проектами.

Орієнтована на об'єкти розробка: TypeScript підтримує об'єктноорієнтований підхід до програмування, включаючи підтримку класів і інтерфейсів.

Розширена інфраструктура: Є багато інструментів і бібліотек, які спрощують розробку на TypeScript, включаючи популярні фреймворки, такі як Angular і React.

Використання TypeScript: TypeScript можна використовувати для розробки вебдодатків, мобільних додатків, бекенд-систем і багатьох інших сфер програмування.

З TypeScript ваш код стає більш структурованим і менше схильним до помилок, що робить його потужним інструментом для розробки програмного забезпечення.

3.1.3 Node.js

Node.js - це середовище виконання JavaScript, яке дозволяє виконувати JavaScript на серверній стороні. Основні характеристики та переваги Node.js включають:

Серверні застосунки: Node.js дозволяє створювати серверні застосунки з використанням JavaScript. Це дозволяє розробникам створювати високоефективні та масштабовані серверні додатки, які можуть обробляти багато запитів одночасно.

Асинхронність: Однією з ключових особливостей Node.js є асинхронність. Він використовує неблокуючий ввод/вивід, що дозволяє обробляти багато запитів без блокування виконання інших операцій. Це особливо корисно для обробки багатьох одночасних запитів.

Велика екосистема: Node.js має велику та активну екосистему бібліотек і модулів, які дозволяють розробникам легко розширювати функціональність своїх додатків. Це включає в себе бібліотеки для роботи з мережами, файловою системою, базами даних і багато інших.

Спільнота та підтримка: Node.js має активну спільноту розробників, яка надає підтримку та розвиває цю платформу. Є багато документації, учбових матеріалів та сторонніх розширень, які полегшують роботу з Node.js.

Швидкість виконання: Node.js базується на движку V8 від Google, який є вельми швидким і виконує JavaScript з високою швидкістю. Це робить Node.js відмінним вибором для обробки запитів у реальному часі та великих обчислювальних завдань.

Node.js використовується для створення різних типів застосунків, включаючи веб-сервери, API, чат-сервери, додатки для роботи з базами даних і багато інших. Він є популярним інструментом у веб-розробці та має широкий спектр застосувань.

3.1.4 Express Js

Express.js, або просто Express, є веб-фреймворком для Node.js, який спрощує створення веб-додатків та веб-серверів. Він надає ряд корисних функцій і інструментів для розробки веб-додатків, що допомагає розробникам швидко створювати потужні та ефективні застосунки. Основні характеристики і переваги Express.js включають:

Маршрутизація: Express дозволяє визначити маршрути для обробки HTTP-запитів. Ви можете визначити, які дії повинні бути виконані при отриманні конкретного запиту (GET, POST, PUT, DELETE і т. д.) і на який URL-шлях вони повинні відповідати.

Мідлвари: Express використовує концепцію мідлварів (middleware), що дозволяє обробляти запити перед тим, як вони досягнуть маршруту. Це дозволяє виконувати автентифікацію, авторизацію, перевірку валідності даних та інші операції перед обробкою запиту.

Шаблонізація: Express дозволяє використовувати шаблонізацію для створення динамічних веб-сторінок. Ви можете використовувати популярні двіжки шаблонів, такі як EJS або Handlebars, для відображення даних на сторінці.

Робота з HTTP-запитами та відповідями: Express надає зручний інтерфейс для роботи з HTTP-запитами та відповідями. Ви можете легко отримувати дані з запитів, надсилати HTTP-відповіді, встановлювати заголовки і багато іншого.

Розширюваність: Express є дуже розширюваним фреймворком. Ви можете додавати власні мідлвари, розширювати функціональність за допомогою сторонніх модулів і створювати власні розширення для веб-додатків.

Активна спільнота: Express має велику та активну спільноту розробників, що робить його добре підтримуваним і надійним фреймворком для веброзробки. Express.js є дуже популярним вибором для створення веб-серверів і веб-додатків на платформі Node.js. Він допомагає розробникам ефективно створювати API, веб-сайти, веб-додатки і багато іншого.

3.1.5 Python

Python - це високорівнева мова програмування, яка відома своєю простотою та читабельністю коду. Вона має широке застосування у різних галузях програмування і включає в себе наступні основні характеристики та переваги:

1. Читабельний синтаксис: Python відомий своїм читабельним і лаконічним синтаксисом, що робить його ідеальним для початківців та досвідчених розробників. Код на Python легко читається і розуміється людьми.
2. Крос-платформеність: Python є крос-платформеним, що означає, що ви можете виконувати код Python на різних операційних системах, включаючи Windows, macOS та Linux.
3. Багата стандартна бібліотека: Python має велику стандартну бібліотеку, яка включає в себе різноманітні модулі і функції для розв'язання різних завдань. Це дозволяє вам швидко створювати програми без необхідності писати код з нуля.
4. Застосування у різних галузях: Python використовується у багатьох галузях, включаючи веб-розробку, наукові дослідження, штучний інтелект, обробку даних, аналітику, робототехніку та інше. Він має багато бібліотек і фреймворків, що полегшують роботу в цих областях.
5. Активна спільнота: Python має велику та активну спільноту розробників, що надає підтримку, створює сторонні бібліотеки та фреймворки, і розвиває мову. Це допомагає вирішувати проблеми та підтримувати Python у актуальному стані.
6. Розширюваність: Python дозволяє вам використовувати модулі, написані на C/C++, що робить його розширюваним та дозволяє взаємодіяти з іншими мовами програмування.

Python є популярним вибором для різних завдань програмування, від розробки веб-додатків до наукових досліджень та штучного інтелекту. Його простота та потужність роблять його однією з найпопулярніших мов програмування у світі.

3.2 Реалізація та тестування веб-додатку

З реалізацію коду методів фільтрування можна ознайомитись в Додатку А.

Основні кроки для тестування веб-додатку Крок 1: Встановлення середовища NodeJS

Для початку роботи з програмою, переконайтеся, що у вас встановлене середовище NodeJS.

Крок 2: Відкриття проєкту у середовищі розробки

Відкрийте проєкт у вашому обраному середовищі розробки, наприклад, Visual Studio або будь-якому іншому, яке вам зручно використовувати.

Крок 3: Запуск веб-застосунку

Після відкриття проекту виконайте наступні дії:

1. В командному рядку (CLI) введіть команду **ng serve**.
2. Після виконання команди, веб-застосунок буде доступний за посиланням **http://localhost:4200** у вашому браузері.

Крок 4: Аутентикація

По замовчуванню, якщо ви не авторизовані, ви будете перенаправлені на аутентикаційну сторінку (home page). На цій сторінці вам буде запропоновано два варіанти:

- Увійти до сайту, якщо у вас вже є обліковий запис.

• Зареєструватись, якщо у вас немає облікового запису. рис 3.1 містить зображення автентифікаційної сторінки.

Рис.3.1 сторінка для входу або реєстрації

Це всі необхідні кроки для початку роботи з програмою. Дотримуйтеся інструкцій та насолоджуйтеся використанням веб-застосунку.

Якщо користувач натисне кнопку "Увійти" (Sign In), то з'явиться модальне вікно для введення особистих даних. Це вікно має наступний вміст:

1. Форма для заповнення особистих даних користувача: В цій формі користувач повинен ввести свою електронну пошту та пароль.

2. Кнопка "Закрити": Користувач може закрити модальне вікно, якщо він не бажає авторизуватися на даний момент.

Кнопка "Увійти": Кнопка для відправки запиту на сервер для автентифікації. Проте, ця кнопка буде заблокована до того моменту, поки користувач не введе усі обов'язкові поля для заповнення, тобто електронну пошту та пароль.

Після успішного запиту на сервер та автентифікації, користувач буде автоматично переправлений на головну сторінку веб-застосунку.

Для кращого розуміння, ось зображення модального вікна для автентифіка́ції:

Рис.3.2 модальне вікно для входу

Це модальне вікно дозволяє користувачеві зручно та безпечно ввести свої облікові дані для входу до системи.

Якщо користувач натисне кнопку "Зареєструватися", то з'явиться модальне вікно для введення особистих даних для реєстрації. Це вікно має наступний вміст:

1. Форма для заповнення особистих даних користувача: У цій формі користувач повинен ввести свою електронну пошту, ім'я, новий пароль та повторне введення паролю для підтвердження.
2. Кнопка "Закрити": Користувач може закрити модальне вікно, якщо він не бажає реєструватися на даний момент.

Кнопка "Зареєструватися": Кнопка для відправки запиту на сервер для реєстрації. Проте, ця кнопка буде заблокована до того моменту, поки користувач не введе усі обов'язкові поля для заповнення, тобто електронну пошту, ім'я та паролі.

Після успішного запиту на сервер та реєстрації, користувач буде автоматично перенаправлений на головну сторінку веб-застосунку.

Для кращого розуміння, ось зображення модального вікна для реєстрації:

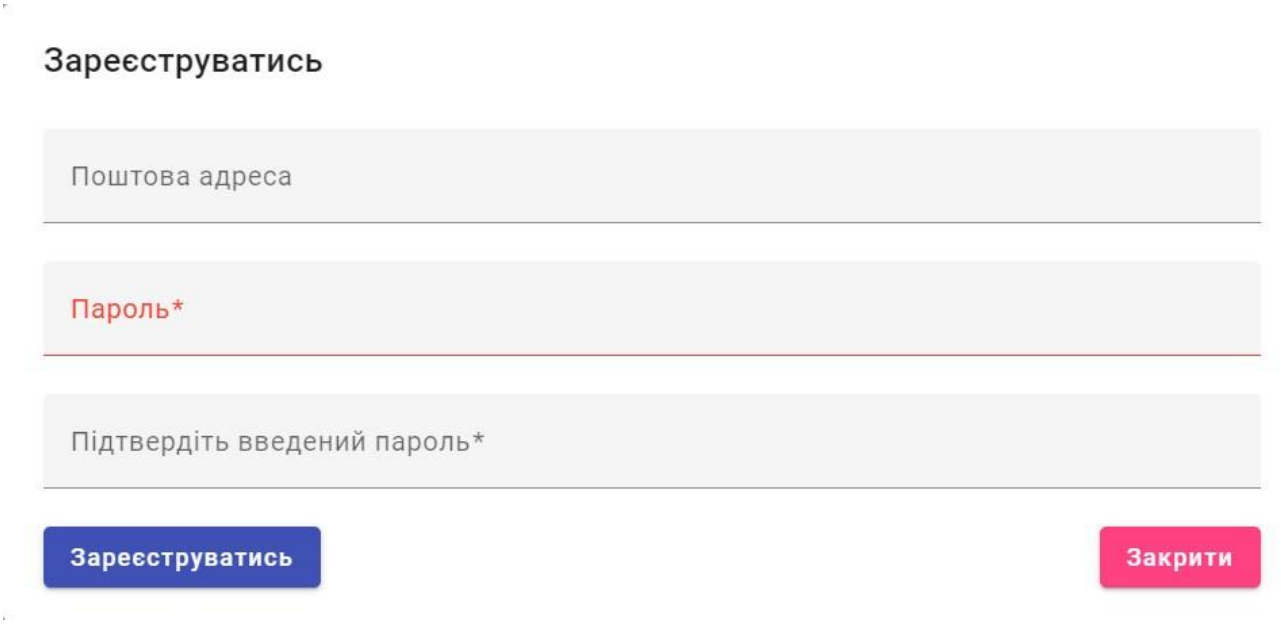

Рис.3.3 модальне вікно для реєстрації

Після того, як користувач натисне на відповідну кнопку (наприклад, "Увійти" або "Зареєструватися") і успішно виконає дію, його буде перенаправлено на головну сторінку веб-застосунку.

Головна сторінка містить наступний функціонал:

1. Список доступних товарів: На головній сторінці зображено перелік усіх доступних товарів. Кожен товар включає зображення товару, ціну та назву. 2. Кнопка для придбання: Кожен товар має кнопку, за допомогою якої користувач може придбати цей товар. При натисканні на цю кнопку і при умові успішної покупки, користувачу буде показане модальне вікно із пропозицією залишити особистий відгук про придбаний товар.

Модальне вікно для відгуків: Модальне вікно для відгуків з'являється після покупки товару і надає користувачу можливість залишити свій відгук про придбаний товар. Відгук може включати текстовий коментар та оцінку товару.

4. Кнопки для отримання рекомендацій з різних систем

Головна сторінка має наступний вигляд рис. 3.4

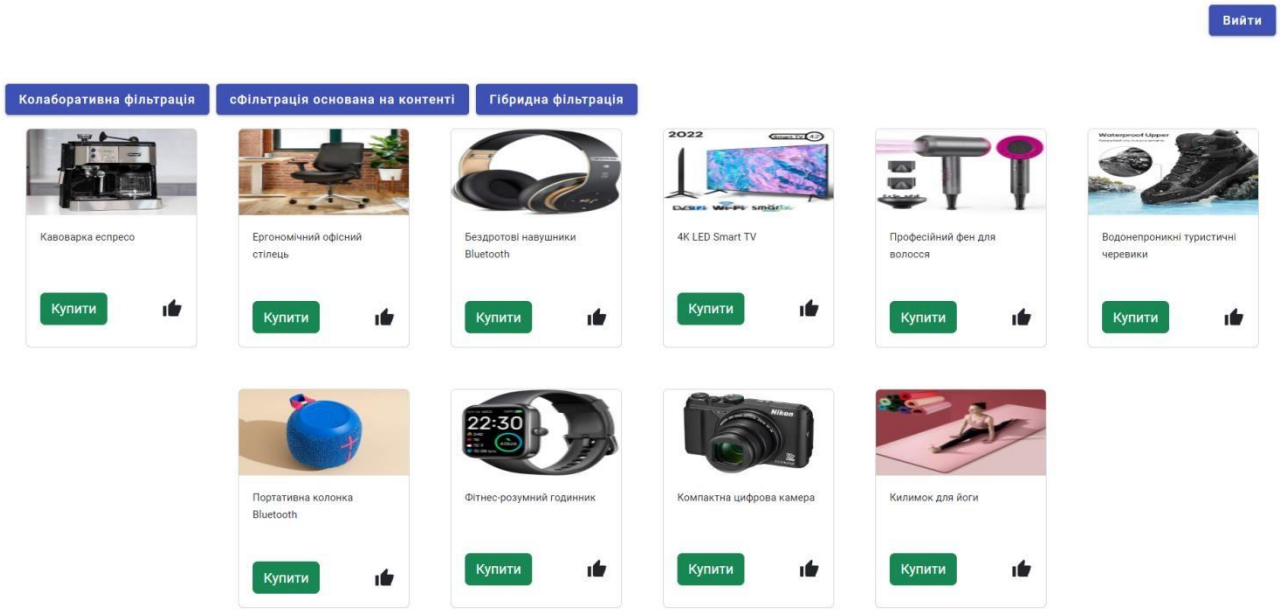

Рис.3.4 головна сторінка

При натисканні на кнопку "Придбати" після успішної покупки товару, з'являється модальне вікно для оцінки придбаного товару. Це вікно призначене для збирання даних, які можуть бути використані для подальшого аналізу та рекомендаційних систем.

Модальне вікно містить наступний функціонал:

1. Форма оцінки товару: В модальному вікні користувач може залишити оцінку придбаного товару. Це може бути числова оцінка або зіркова рейтингова система.

2. Кнопка "Зберегти": Після заповнення форми оцінки та коментаря користувач натискає кнопку "Зберегти", щоб зберегти свій відгук.

3. Кнопка "Скасувати": Кнопка "Скасувати" дозволяє користувачеві закрити модальне вікно без збереження відгуку.

Модальне вікно для оцінки придбаного товару має наступний вигляд рис.3.6 (це приклад):

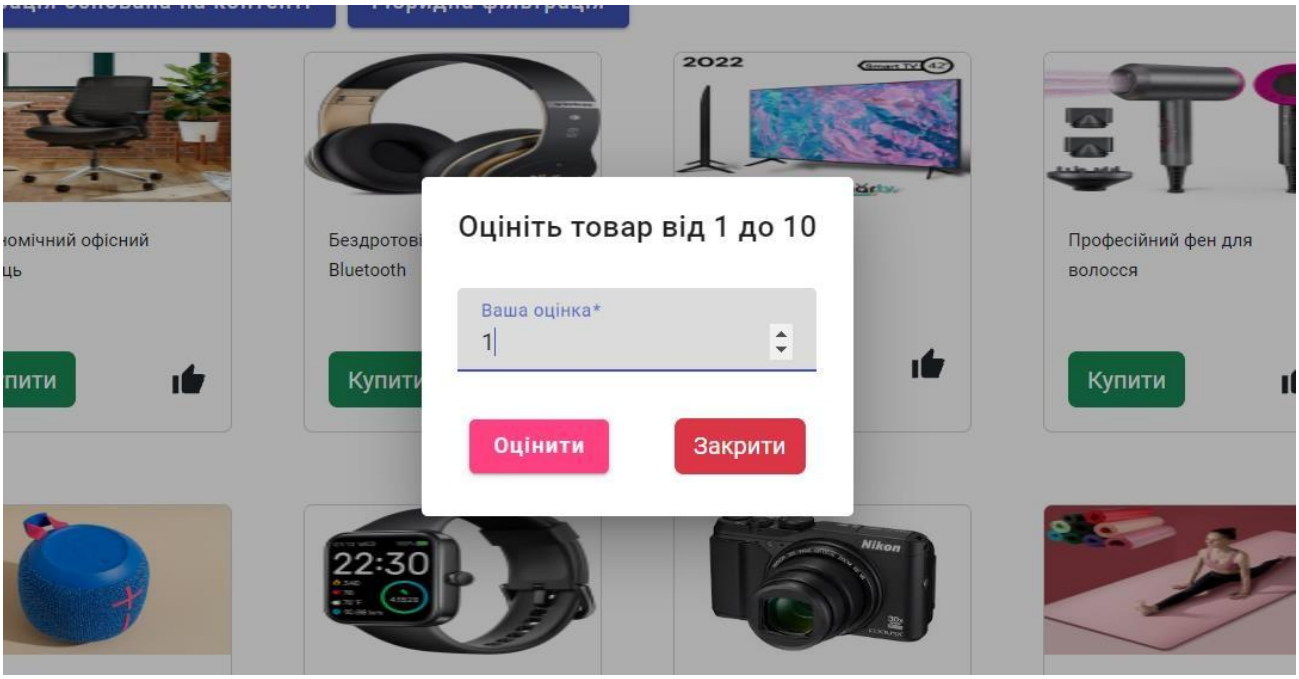

Рис.3.5 модальне вікно оцінки товару

Після успішного підтвердження відгук про товар – модальне вікно зачиняється.

3.2.1 Перевірка колаборативної фільтрації

Попередні дії ми виконали для користувача test7@gmail.com

Тепер ми можемо перевірити наскільки діє наш алгоритм для колаборативної фільтрації

Натискаємо кнопку колаборатива фільтрація, яка повинно викликати арі для запуску алгорітма.

На кінці я подаю векторне представлення матриці , для перевірки точності данних рис.3.7

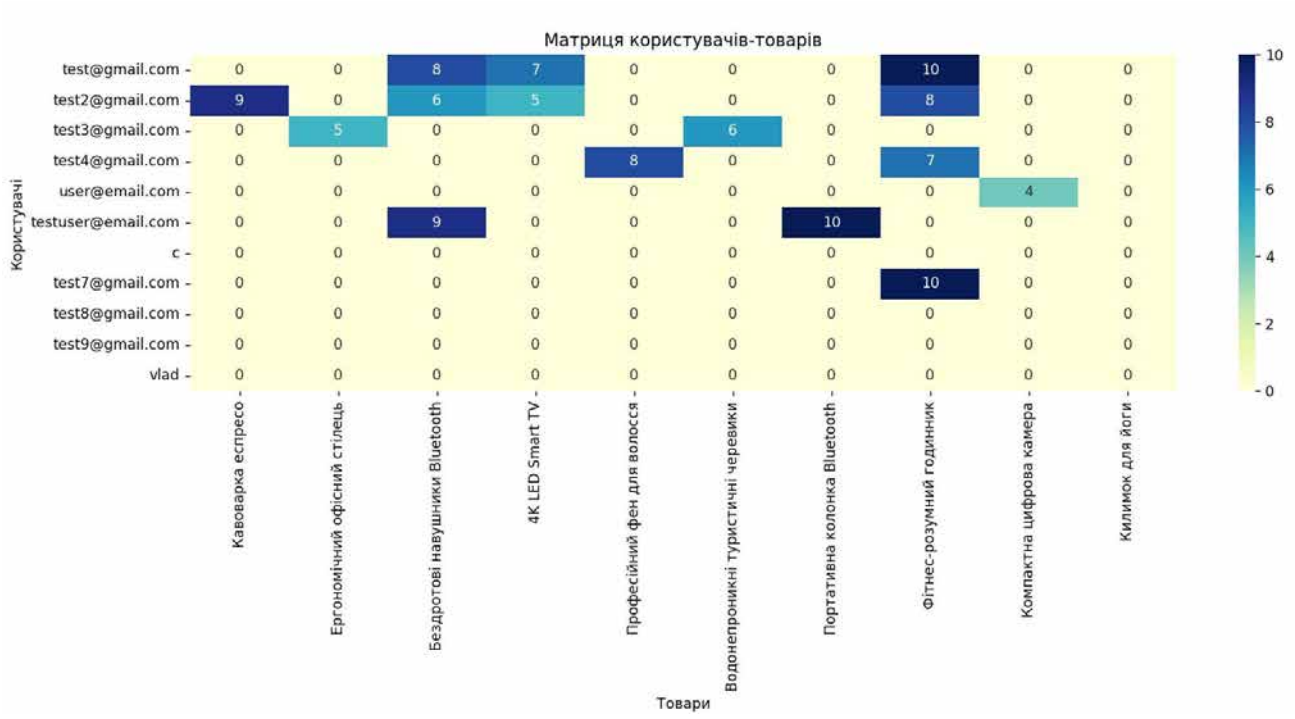

Рис.3.6 матриця даних колаборативного фільтрування

Процес колаборативної фільтрації:

1. **Збір оцінок**: Система збирає оцінки від різних користувачів для різних предметів (як вказано в матриці).
2. **Виявлення подібних користувачів**: Для кожного користувача система шукає інших користувачів зі схожими оцінками. Наприклад, якщо два користувачі високо оцінили одні й ті ж предмети, вони вважаються схожими.
3. **Визначення невідомих оцінок**: Використовуючи оцінки подібних користувачів, система передбачає, як високо користувач може оцінити предмет, для якого оцінка ще не була надана (поля з нулями в матриці).
4. **Рекомендація предметів**: На основі передбачених оцінок, система вибирає предмети з найвищими прогнозованими оцінками, щоб рекомендувати їх користувачеві.

Отже зважаючи на кроки та проаналізувавши рис. 3.7. можна прорахувати, щокористувач test@gmail.com та test7@gmail.com(на якому ми тестуємо) схожі, отже товари користувача test@gmail.com мають бути зарекомендовані Wireless

Bluetooth Headphones та 4K LED Smart TV. Після завершення виконання апі, ми відкриваємо модальне вікно, з рекомендуючими товарами рис.3.8

Рис.3.7 модальне вікно з рекомендаціями для колаборативної фільтрації

Можемо зробити висновок, що алгоритм виконується вірно.

3.2.2 Перевірка фільтрації на основі контенту

Модель товарів

Id	ідентифікатор товару
Name	назва
Category	категорії до яких входить товар
features	функціональність.
Image	посилання на картинку

Модель користуача

Id	ідентифікатор користувача
username	Ім’я користувача
password	Захешований пароль
preferences	Вподобання користувача
purchasedItems	Куплені товари
viewedItems	Товари які користувач дивився
ratingsGiven	Масив ідентифікаторів товарів, для яких користувач відгукнувся

З алгоритмом фільтрації на основі контенту можно ознайомитись у додадтку А.

Попередньо я вже обновив модель для testuser@email.com,

Додавши інформацію про його куплені, оцінені та переглянуті товари.

Виходячи з цього, в нього сформувались такі вподобання :

1)Електроніка

2) Фітнес

3) Кемпінг

Після заповнення даних, можемо починати тестування.

1) Натискаємо кнопку contentBasedFiltering.
2) Робимо запит на апі та очікуємо виконання.
3) Кроки виконання алгоритму:
    a. Аналіз характеристик предметів: Спочатку визначаються атрибути або характеристики - категорія, технічні специфікації тощо.

b. Створення профілю користувача: Для кожного користувача на основі його попередніх оцінок створюється профіль, який показує його переваги щодо цих атрибутів. Наприклад, якщо користувач високо оцінив " Wireless Bluetooth Headphones " і "4K LED Smart TV", то профіль може підкреслити високу цінність для користувача технологічної складності та якості продукту.

c. Порівняння предметів з профілем: Коли користувач шукає новий предмет, система порівнює його атрибути з профілем користувача. Якщо характеристики предмета відповідають профілю, цей предмет може бути рекомендований.

d. Рекомендація предметів: На основі цього порівняння система вибирає предмети, які найкраще відповідають індивідуальним перевагам користувача, та рекомендує їх

4) Після виконання отримуємо рекомендації а відкриваємо модальне вікно з ними рис.3.9

Товар	Категорія
Bluetooth Portable Speaker	Електроніка
Yoga Mat	Фітнес, Кемпінг
Fitness Smartwatch	Фітнес, Електроніка
Waterproof Hiking Boots	Кемпінг
Compact Digital Camera	Електроніка

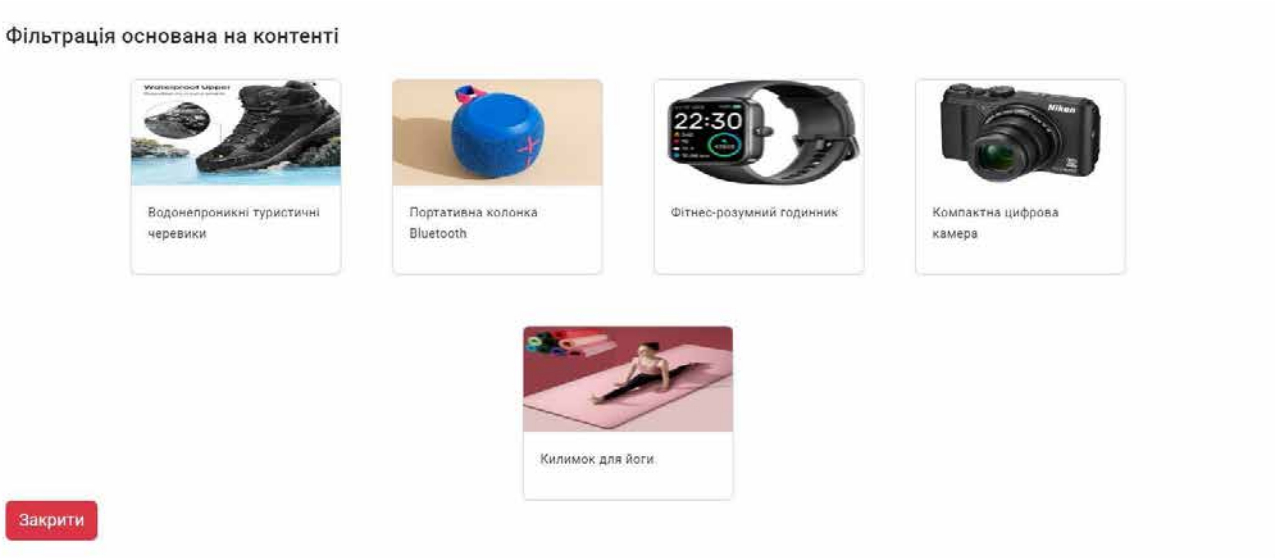

Рис.3.8 модальне вікно з рекомендаціями для фільтрації на основі контенту

Опираючись на побажання користувача та категоріями товарів які були рекомендовані, можемо зробити висновок, що алгоритм працює.

3.2.3 Перевірка фільтрації гібридної фільтрації

Гібридна фільтрація комбінує елементи колаборативної фільтрації та фільтрації на основі контенту для покращення рекомендаційних систем. Вона прагне уникнути обмежень, які мають ці два методи окремо, таких як проблема холодного старту в колаборативній фільтрації або обмежене різноманіття рекомендацій у фільтрації на основі контенту.

Кроки тестування:

1) Натискаємо кнопку hybridFiltering, та робимо запит на api.
2) Кроки виконання алгорітму:
    - Об'єднання інформації: Система використовує оцінки користувачів, щоб знайти подібності між ними (колаборативна фільтрація), а також

аналізує характеристики предметів, які вони оцінили (фільтрації на основі контенту).

- Створення профілю користувача: Профіль користувача формується на основі його власних оцінок та характеристик оцінених предметів, а також оцінок та вподобань подібних користувачів.
- Поєднання рекомендацій: Система враховує як схожість з іншими користувачами, так і переваги користувача до певних характеристик предметів для формування рекомендацій.
- Адаптація та оптимізація: Гібридна система може адаптуватися до змін у поведінці користувачів, динамічно оновлюючи рекомендації на основі нових даних.
- Видача рекомендацій: Користувачу надаються рекомендації, які базуються на комплексному аналізі його минулих оцінок та вподобань, а також на вподобаннях користувачів з схожими інтересами

3) Після виконання api, отримуємо рекомендації та відкриваємо модальне вікно для відображення їх(див. рис.3.9). Дивлячись на результати колаборативна фільтрації та фільтрації на основі контенту , можемо зрозуміти, що ми отримали уніфікований результат рекомендацій виконання двох попередніх алгоритмів.

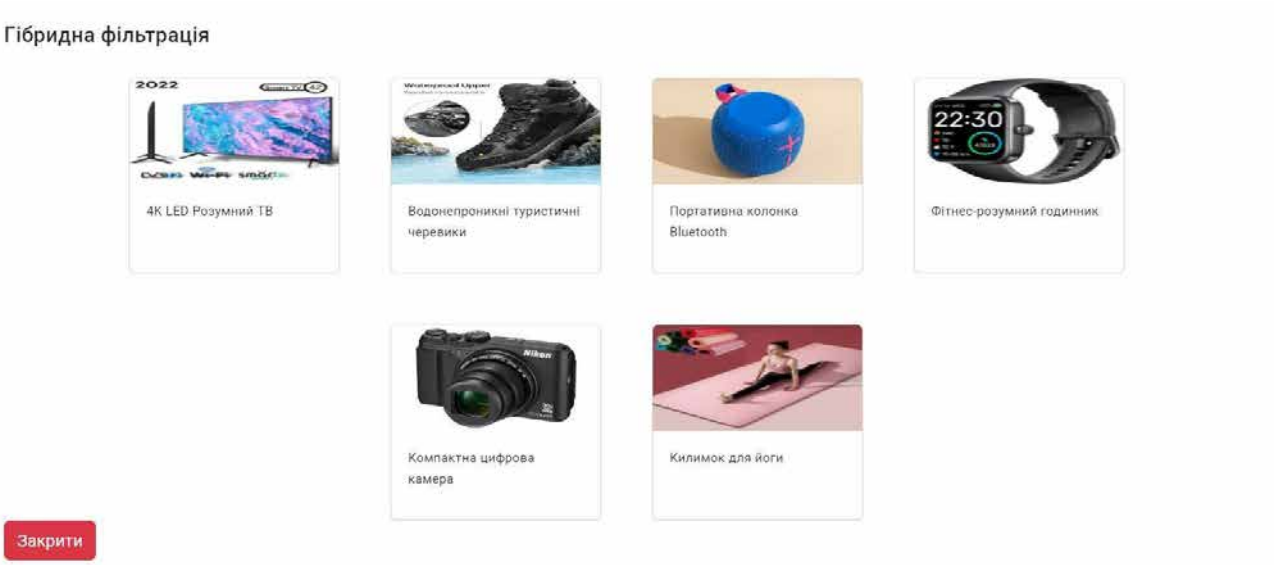

Рис. 3.9 модальне вікно з рекомендаціями для гібридної фільтрації

ВИСНОВКИ ДО РОЗДІЛУ 3

У цьому розділі висвітлюється процес реалізації та тестування вебдодатку з використанням трьох різних методів фільтрації для рекомендаційних систем: колаборативної фільтрації, фільтрації на основі контенту та гібридної фільтрації. Кожен метод пройшов кроки налаштування середовища, запуску веб-додатку, аутентикації користувача та роботи з модальними вікнами для входу, реєстрації та рекомендацій.

Колаборативна фільтрація використовувала оцінки користувачів для виявлення подібності між ними та рекомендувала продукти на основі цих подібностей. Фільтрація на основі контенту аналізувала характеристики товарів, які користувачі вже оцінили, та рекомендувала подібні товари, засновані на цих характеристиках. Гібридна фільтрація поєднувала обидва ці підходи для створення більш точних та персоналізованих рекомендацій.

Тестування показало, що кожен із методів ефективно рекомендував товари, які відповідали інтересам та вподобанням користувачів, що підтверджується

відображенням рекомендованих товарів у відповідних модальних вікнах. Отже, можна зробити висновок, що реалізовані алгоритми фільтрації ефективні та можуть бути використані для покращення досвіду користувачів у веб-додатку.

ВИСНОВКИ

У даній роботі було проведено дослідження та розроблено систему рекомендацій товарів, використовуючи різні алгоритми, зокрема колаборативну фільтрацію та алгоритм заснований на контенті. Розглянуті алгоритми були порівняні за їхньою ефективністю та перевагами.

У першому розділі проведено аналіз різних видів систем рекомендацій, включаючи колаборативну фільтрацію, алгоритми засновані на контенті та гібрідній фільтрації.В результаті порівняльного аналізу були виділені переваги та недоліки кожного з цих підходів.

У другому розділі була представлена структура розробленої системи, включаючи вибір мови програмування, середовища розробки та використання сторонніх бібліотек. Описано архітектуру фронтед та бекенд застосунку та використані сучасні технології розробки.

У третьому розділі було надано математичну постановку задач із детальним поясненням математичного підґрунтя стосовно реалізації алгоритму.

У четвертому розділі проведено експериментальне дослідження та перевірку правильності роботи алгоритмів та застосунку. Демонструється коректна робота логіки програми.

Результатом роботи є працездатна система рекомендацій товарів, яка використовує алгоритми машинного навчання та може бути використана для побудови веб-застосунку. Система дозволяє користувачам отримувати персоналізовані рекомендації щодо товарів у магазині. Розроблене рішення є актуальним та має потенціал для подальшого розвитку та масштабування.

Таке поєднання сучасних методів веб-розробки з математичним підґрунтям дозволяє створити високоефективну систему рекомендацій для електронного магазину та забезпечити покращення користувацького досвіду та збільшення обсягів продажів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1) Річчі, Ф., Рокач, Л., і Шапіра, Б. Посібник з систем рекомендацій. Спрінгер., 2015 р., 1003 с.
2) Белл, Р. М., Корен, Й., та Волінський, К. Моделювання взаємозв'язків на різних рівнях для підвищення точності великих систем рекомендацій. KDD '07, 2007, с.104
3) Лопс, П., де Гемміс, М., та Семераро, Г. Системи рекомендацій на основі контенту: стан мистецтва та тенденції. Посібник з систем рекомендацій, Спрінгер., 2011, 1003.с
4) Аггарвал, К. К.. Системи рекомендацій: Підручник. Спрінгер., 2016, 498 с.
5) Корен, Й., Белл, Р., та Волінський, К. , Техніки матричної факторизації для систем рекомендацій. Комп'ютер, 2007 47с..
6) Лінден, Г., Сміт, Б., та Йорк, Дж. (2003). Рекомендації Amazon.com: Колаборативне фільтрування від предмету до предмету. IEEE Інтернетобчислення, 2003 р., 80 с..
7) Сарвар, Б., Каріпіс, Г., Констан, Дж., та Рідл, Дж. Алгоритми рекомендаційних систем на основі колаборативного фільтрування предметів. WWW '01., 2001 р., 295 с.
8) Бобаділла, Дж., Ортега, Ф., Ернандо, А., та Гутьєррес, А. Огляд систем рекомендацій. Системи на основі знань, 2013 р., 46 с.
9) Шафер, Дж. Б., Констан, Дж. А., та Рідл, Дж. (2001). Рекомендаційні застосування в електронній комерції. Майнінг та відкриття знань, 2001 р., 5 с.
10) Бурк, Р Гібридні рекомендаційні системи: Огляд та експерименти. Моделювання користувачів та адаптована взаємодія, 2002 р. 12с.
11) Адомавічус, Г., та Тузілін, А. До наступного покоління рекомендаційних систем: огляд сучасного стану мистецтва та можливі розширення. IEEE Транзакції знань та інженерії даних, 2005, 17с.

12) Рекомендаційна система [Електронний ресурс] - Режим доступу Wikipedia
13) Surprise - бібліотека для побудови та аналізу систем рекомендацій, що працюють з явними даними рейтингів. [Електронний ресурс] Режим доступу https://github.com/NicolasHug/Surprise
14) Scikit-learn - машина для навчання з великим набором алгоритмів, включаючи рекомендаційні системи. [Електронний ресурс] Режим доступу https://github.com/scikit-learn/scikit-learn
15) Pandas - бібліотека для аналізу даних, яка пропонує зручні структури даних та операції для маніпулювання числовими таблицями та часовими рядами. [Електронний ресурс] Режим доступу https://github.com/pandas-dev/pandas
16) Numpy - основна бібліотека для наукових обчислень у Python, що підтримує великі багатовимірні масиви та матриці. [Електронний ресурс] Режим доступу https://github.com/numpy/numpy
19. Можливості Figma. Сервіс для розробки веб-дизайну. hyperhost: вебсайт. URL: https://hyperhost.ua/info/uk/mozhlivosti-figma-servis-dlya-rozrobki-vebdizaynu (дата звернення 20.01.2022)
17) 20. What is wireframing?. experienceux: вебсайт. URL: https://www.experienceux.co.uk/faqs/what-is-wireframing/(дата звернення Документація Angular [Електронний ресурс] - Режим доступу https://angular.dev/
18) Документація [Електронний ресурс] - Режим доступу Express JS

Додаток А

Програмний код

1) Колаборативна фільтрація:

```
2) import json
3) import sys
4) import seaborn as sns
5) import matplotlib.pyplot as plt
6) from sklearn.metrics.pairwise import cosine_similarity
7) from scipy import sparse
8) import numpy as np
9) 10)
11) def collaborative_filtering(user_id, users, items, ratings, N=1): 12)
item_ids = [item['_id'] for item in items]
13)     # Витягуємо ID користувачів зі списку користувачів
14)     user_ids = [user['_id'] for user in users]
15)     item_names = [item['name'] for item in items]
16)     # Витягуємо імена користувачів зі списку користувачів
17)     user_names = [user['username'] for user in users] 18)
19)             try:
20)             # Створення матриці користувач-продукт
21)             matrix = []
22)             for user in users:
23)             if '_id' not in user:
24)             raise ValueError(f"Відсутній '_id' у користувача: {user}")
25)             row = []
26)             for item in items:
27)             if '_id' not in item:
28)             raise ValueError(f"Відсутній '_id' у товара: {item}")
29)             # Шукаємо рейтинг, який відповідає даному користувачу та продукту
30)             rating = next((r['rating'] for r in ratings if str(r['userId']) == str(user['_id']) and str(r['itemId']) ==
                str(item['_id'])), 0)
31)             row.append(rating)
32)             matrix.append(row)
33)
34)     # Перетворення матриці на розріджену для підвищення ефективності
35)     matrix_sparse = sparse.csr_matrix(matrix)
36)
37)     # Розрахунок косинусної подібності між користувачами
38)     similarities = cosine_similarity(matrix_sparse) 39)
40)     # Визначення індексу даного користувача
41)     user_index = next(i for i, user in enumerate(users) if str(user['_id']) == user_id) 42)
43)     # Визначення індексів N найбільш схожих користувачів
44)     most_similar_user_indices = similarities[user_index].argsort()[-(N+1):-1] 45)
46)     # Визначення товарів, які оцінили ці користувачі, але не оцінив поточний користувач
47)     unrated_items = [(i, item['_id']) for i, item in enumerate(items) if matrix[user_index][i] == 0 and
        any(matrix[other_user_index][i] > 0 for other_user_index in most_similar_user_indices)]
48)     # Отримання ID п'яти найкращих товарів
```

```
49)         recommendations = [str(item_id) for _, item_id in unrated_items[:5]]
50)
51)         # Виведення рекомендацій
52)         print(json.dumps(recommendations))
53)
54)        # Після отримання рекомендацій конвертуємо матрицю для візуалізації 55)
ratings_matrix = np.array(matrix)
```

2) Фільтрація на основі контенту

```
3) import sys
4) import json
5) from sklearn.feature_extraction.text import TfidfVectorizer
6) from sklearn.metrics.pairwise import cosine_similarity


7) def generate_user_profile(user, items, ratings):
8) # Об'єднання вподобань користувача, переглянутих, куплених та вподобаних товарів 9) interests = user.get('preferences', [])
10) interests += [item['name'] for item in items if item['_id'] in user.get('viewedItems', []) + user.get('purchasedItems', []) + user.get('likedItems', [])]
11) return ' '.join(interests)


12) def content_based_filtering(user_id, users, items, ratings): 13) user = next((u for u in users if u['_id'] == user_id), None) 14) if not user:
15) raise ValueError(f"Користувача з ID {user_id} не знайдено")


16) user_profile = generate_user_profile(user, items, ratings)


17) # Векторизація
18) tfidf_vectorizer = TfidfVectorizer(stop_words='english')
19) item_tfidf_matrix = tfidf_vectorizer.fit_transform([item['description'] for item in items]) 20) user_tfidf = tfidf_vectorizer.transform([user_profile])


21) # Розрахунок схожості
22) cosine_similarities = cosine_similarity(user_tfidf, item_tfidf_matrix).flatten()
23) recommended_item_indices = cosine_similarities.argsort()[-5:][::-1]
24) recommended_items = [items[index]['_id'] for index in recommended_item_indices]


25) return recommended_items


26) if __name__ == '__main__': 27) user_id = sys.argv[1]
28) users = json.loads(sys.argv[2])
29) items = json.loads(sys.argv[3])
30) ratings = json.loads(sys.argv[4])


31) try:
```

```
32) recommendations = content_based_filtering(user_id, users, items, ratings) 33) print(json.dumps(recommendations)) 34) except Exception as e:
35) print(f"Помилка: {str(e)}", file=sys.stderr)
36) sys.exit(1)
37)
```

3) Гібридна фільтарція

```
4) export const hybridFiltering = async (req, res) => {
5) try {
6) const userId = req.user._id;
7) const users = await User.find({});
8) const items = await Item.find({});
9) const ratings = await Rating.find({});
10)
11) // Запуск скрипта для колаборативної фільтрації
12) const collaborativePython = spawn('python', ['-u', './python/collaborative_filtering.py', userId, JSON.stringify(users), JSON.stringify(items), JSON.stringify(ratings)]);
13) let collaborativeData = '';
14)
15) // Запуск скрипта для контент-орієнтованої фільтрації
16) const contentBasedPython = spawn('python', ['-u', './python/content_based.py', userId, JSON.stringify(users), JSON.stringify(items), JSON.stringify(ratings)]);
17) let contentBasedData = '';
18)
19) collaborativePython.stdout.on('data', (data) => {
20) collaborativeData += data.toString();
21) });
22)
23) contentBasedPython.stdout.on('data', (data) => {
24) contentBasedData += data.toString();
25) });
26)
27) // Обробка завершення обох скриптів
28) Promise.all([
29) new Promise((resolve, reject) => {
30) collaborativePython.on('close', (code) => {
31) if (code !== 0) reject('Помилка у скрипті колаборативної фільтрації'); 32) else resolve();
33) });
34) }),
35) new Promise((resolve, reject) => {
36) contentBasedPython.on('close', (code) => {
37) if (code !== 0) reject('Помилка у скрипті контент-орієнтованої фільтрації'); 38) else resolve();
39) });
40) })
41) ]).then(() => {
42) // Комбінування результатів обох методів
43) const collaborativeRecommendations = JSON.parse(collaborativeData);
44) const contentBasedRecommendations = JSON.parse(contentBasedData); 45)
46) // Приклад комбінації: об'єднання та видалення дублікатів
47) const combinedRecommendations = [...new Set([...collaborativeRecommendations, ...contentBasedRecommendations])];
```

```
48)
49)     res.status(201).json(combinedRecommendations);
50)     }).catch((error) => {
51)     console.error(error);
52)     res.status(500).send('Помилка при обробці гібридних рекомендацій.');
53)     });
54)     } catch (error) {
55)     console.error('Помилка сервера:', error);
56)     res.status(500).send('Внутрішня помилка сервера.');
57)     }
58)     };
```