

Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерних наук

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему Метод рекомендацій за аналізом попередньої купівельної поведінки клієнта засобами машинного навчання для вебсистеми ігрових консолей та аксесуарів

Галузь знань 12 – Інформаційні технології
Шифр і назва галузі знань
Спеціальність 122 – Комп'ютерні науки
Шифр і назва спеціальності
Освітня програма Комп'ютерні науки
Назва освітньої програми

Виконала: студентка групи КН-20-2  Стефанія БОБК
Група виконавця Підпис Ім'я, ПРІЗВИЩЕ
Керівник: зав. каф. КН, д.т.н., проф.  Олександр БАРМАК
Науковий ступінь, посада Підпис Ім'я, ПРІЗВИЩЕ
Нормоконтроль: к.т.н., доц. каф. КН  Руслан БАГРІЙ
Науковий ступінь, посада Підпис Ім'я, ПРІЗВИЩЕ

До захисту допускаю:
зав. кафедри КН, д.т.н., професор  Олександр БАРМАК
Підпис Ім'я, ПРІЗВИЩЕ
11 червня 2024 р.

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій

Кафедра комп'ютерних наук


Освітній ступінь бакалавр

Галузь знань 12 – Інформаційні технології

Спеціальність 122 – Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри комп'ютерних наук


(підпис)
д.т.н., професор Олександр БАРМАК
« 16 » 02. 2024 року

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА**

1. Тема кваліфікаційної роботи бакалавра: «Метод рекомендацій за аналізом попередньої купівельної поведінки клієнта засобами машинного навчання для вебсистеми ігрових консолей та аксесуарів»

2. Завдання видано студентці Стефанії БОВК
(ім'я, прізвище)

3. Керівник роботи зав. кафедри КН, д.т.н., професор Олександр БАРМАК
(посада, ім'я, прізвище)

4. Затверджено наказом університету від « 15 » 02. 2024 р. № 8

5. Дата видачі завдання студенту: « 16 » 02. 2024 р.

6. Зміст пояснювальної записки (перелік задач) та вихідні дані:

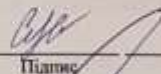
Мета роботи – покращення ефективності отримання персоналізованих рекомендацій за аналізом попередньої купівельної поведінки клієнта засобами машинного навчання для вебсистеми ігрових консолей та аксесуарів. При створенні методу враховуються такі параметри як: покупки клієнтів системи та всі товари, що наявні у базі для коректної роботи. Також слід забезпечити правильну роботу функцій додавання, редагування, видалення та перегляду товарів замовлень користувачів, функції додавання товарів у кошик та створення замовлення у роботі системи, одержання виводу роботи методу рекомендацій та таблиці з аналізом ефективності системи.

7. Календарний план виконання кваліфікаційної роботи бакалавра:

№	Назва етапів (розділів) кваліфікаційної роботи бакалавра	Термін виконання	Примітка
1	Вибір напрямку дослідження та узгодження тематики кваліфікаційної роботи бакалавра з керівником, складання календарного графіка виконання роботи	січень 2024	виконано
2	Ознайомлення з предметною областю, формулювання мети та задач дослідження, визначення об'єкта та предмета дослідження	січень 2024	виконано
3	Робота над розділом 1 Огляд підходів до персоналізованих рекомендаційних систем пропозицій покупцю	лютий 2024	виконано
4	Робота над розділом 2 – Метод рекомендацій товарів за аналізом попередньої купівельної поведінки	березень 2024	виконано
5	Робота над розділом 3 – Програмна реалізація вебсистеми ігрових консолей та аксесуарів	квітень 2024	виконано
6	Оформлення пояснювальної записки згідно вимог	травень 2024	виконано
7	Попередній захист кваліфікаційної роботи бакалавра	травень 2024	виконано
8	Захист кваліфікаційної роботи бакалавра	червень 2024	виконано

Виконавець: студентка групи КН-20-2

Група виконавця

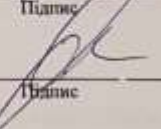

Підпис

Стефанія БОБК

Ім'я, ПРІЗВИЩЕ

Керівник: д.т.н., професор КН

Науковий ступінь, посада


Підпис

Олександр БАРМАК

Ім'я, ПРІЗВИЩЕ

Анотація

Тема кваліфікаційної роботи бакалавра: «Метод рекомендацій за аналізом попередньої купівельної поведінки клієнта засобами машинного навчання для вебсистеми ігрових консолей та аксесуарів»

Виконавець кваліфікаційної роботи бакалавра: студентка групи КН-20-2 Стефанія ВОВК

Керівник кваліфікаційної роботи бакалавра: зав. кафедри КН, д.т.н., професор Олександр БАРМАК

Кваліфікаційна робота бакалавра містить:

Пояснювальна записка				Кількість додатків
Сторінок	Рисунків	Таблиць	Джерел інформації	
72	23	6	30	4

Метою кваліфікаційної роботи бакалавра є покращення ефективності отримання персоналізованих рекомендацій за аналізом попередньої купівельної поведінки клієнта засобами машинного навчання для вебсистеми ігрових консолей та аксесуарів. Розробка методу рекомендацій потребує мови програмування Python з головною бібліотекою scikit-learn та інтеграції у вебсистему. Для розробки архітектури вебсистеми було використано мови програмування JavaScript з фреймворком Vue3.js та PHP з фреймворком Laravel 9, а також систему керування базами даних MySQL.

Розроблена система з імплементованим методом призначена для клієнтів та покупців веб-системи ігрових консолей та аксесуарів до них. Реалізований метод рекомендацій допомагає покращити дохід власника вебсистеми та пропонує користувачам системи персоналізовані рекомендації, враховуючи попередні замовлення.

Ключові слова: рекомендації, вебсистема, купівельна поведінка, замовлення, спільна фільтрація, фільтрація на основі контексту, ігрові консолі.

Виконавець: студентка групи КН-20-2

Група виконавця

Підпис

Стефанія ВОВК

Ім'я, ПРІЗВИЩЕ

Зміст

Перелік скорочень	3
Вступ.....	4
Розділ 1 Огляд підходів до персоналізованих рекомендаційних систем пропозицій покупцю	6
1.1 Огляд способів формування рекомендацій наступних покупок засобами машинного навчання.....	6
1.2 Огляд існуючих засобів реалізації рекомендацій наступних покупок.....	11
1.3 Огляд особливостей застосування рекомендацій наступних покупок для систем електронної торгівлі.....	14
1.4 Мета та завдання кваліфікаційної роботи	18
Розділ 2 Метод рекомендацій товарів за аналізом попередньої купівельної поведінки.....	19
2.1 Метод рекомендацій товарів за аналізом попередньої купівельної поведінки засобами машинного навчання.....	20
2.2 Набір даних для навчання моделі.....	26
2.3 Розробка архітектури вебсистеми ігрових консолей та аксесуарів	28
2.3.1 Проектування вебінтерфейсу інформаційної системи.....	28
2.3.2 Проектування структури бази даних	35
2.4 Особливості використання спеціалізованих програмних компонентів	38
2.5 Спосіб оцінювання ефективності вебсистеми з запропонованим методом.....	41
2.6 Висновки до 2 розділу	44
Розділ 3 Програмна реалізація вебсистеми ігрових консолей та аксесуарів	46
3.1 Структура та особливості реалізації методу рекомендацій за аналізом попередньої купівельної поведінки клієнта	46
3.2 Структура та особливості реалізації вебсистеми ігрових консолей та аксесуарів	51
3.3 Тестування інформаційної системи	56
3.4 Вимоги до публікації вебсистеми	64
3.5 Визначення ефективності вебсистеми з запропонованим методом	65
3.6 Висновки до розділу 3	67
Загальні висновки.....	69
Перелік посилань.....	71
Додатки	

Перелік скорочень

Скорочення, термін, позначення	Пояснення
SVD	Singular-Value Decomposition
PMF	Probabilistic Matrix Factorization
NMF	Non-negative matrix factorization
ALS	Alternating Least Squares
BPR	Bayesian Personalized Ranking
LMF	Logistic Matrix Factorization
KH	Комп'ютерні науки
TF-IDF	Term Frequency – Inverse Document Frequency
TF	Term Frequency
IDF	Inverse Document Frequency
PHP	Hypertext Preprocessor
MAE	Mean Absolute Error
RMSE	Root Mean Squared Error
NDCG	Normalized Discounted Cumulative Gain
MVC	Model-View-Controller
NMP	Node Package Manager

Вступ

Кваліфікаційна робота бакалавра присвячена покращенню ефективності отримання персоналізованих рекомендацій за аналізом попередньої купівельної поведінки клієнта засобами машинного навчання для вебсистеми ігрових консолей та аксесуарів шляхом розробки методу рекомендацій за аналізом попередньої купівельної поведінки клієнта засобами машинного навчання.

Актуальність. У сьогоденні користувачі вимагають персоналізованого підходу до будь-яких дій, починаючи з отримання якісного контенту у відеоіграх: від складності та геймплею до відчуття імпаكتу й емоцій, закінчуючи вибором товарів та послуг під час онлайн-покупок. Оскільки більшість людей сьогодні здійснюють покупки в інтернеті, щоб виділитися серед конкурентів та забезпечити собі найкращі умови, необхідно надавати персоналізовані рекомендації, що відповідають потребам та інтересам клієнтів. Тому, створення персоналізованих рекомендацій для користувачів є невід'ємною частиною кожного бізнесу, яка сприяє підвищенню конкурентоспроможності та задоволення потреб споживачів. Дослідження цієї теми стає актуальною частиною вебсистем, що спрямовані на розвиток та успішну діяльність в онлайн-середовищі покупок.

Об'єкт дослідження – процес отримання персоналізованих рекомендацій покупок за аналізом попередньої купівельної поведінки засобами машинного навчання.

Предмет дослідження – методи машинного навчання для отримання персоналізованих рекомендацій покупок за аналізом попередньої купівельної поведінки клієнта.

Мета кваліфікаційної роботи бакалавра – покращення ефективності отримання персоналізованих рекомендацій за аналізом попередньої купівельної поведінки клієнта засобами машинного навчання для вебсистеми ігрових консолей та аксесуарів.

Завдання кваліфікаційної роботи бакалавра – провести аналіз предметної області для задачі отримання персоналізованих рекомендацій за аналізом попередньої купівельної поведінки клієнта засобами машинного навчання для вебсистеми ігрових консолей та аксесуарів; виконати аналіз способів формування рекомендацій наступних покупок для користувача; виконати огляд та аналіз існуючих рішень щодо реалізації подібних задач; розробити метод рекомендацій за аналізом попередньої купівельної поведінки клієнта засобами машинного навчання; спроектувати структуру вебсистеми ігрових консолей та аксесуарів, а також базу даних для даної системи та методу; обрати сучасні засоби реалізації методу рекомендацій за аналізом попередньої купівельної поведінки клієнта засобами машинного навчання для вебсистеми ігрових консолей та аксесуарів; реалізувати вебсистему ігрових консолей та аксесуарів та запропонований метод; протестувати програмну реалізацію на коректність роботи всіх функцій; показати ефективність системи з запропонованим методом.

Розділ 1 Огляд підходів до персоналізованих рекомендаційних систем пропозицій покупцю

1.1 Огляд способів формування рекомендацій наступних покупок засобами машинного навчання

У сучасному стилі життя електронна комерція стала незамінною. Навігація величезною кількістю продуктів, доступних в Інтернеті, може бути надзвичайно важкою, що спонукає до необхідності ефективних методів пошуку. Тому у світі, де кількість товарів і послуг зростає експоненційно, забезпечення користувачів персоналізованими рекомендаціями стає ключовим завданням для багатьох компаній.

Наразі для просування свого продукту необхідно створювати рекламу продуктів, або просувати вже на інформаційній системі безпосередньо – товари. Для цього, необхідно заохочувати клієнтів якомога більше часу витратити на перегляд – перебування в застосунку, робити більше замовлень та оплачуваних покупок. Однак пошук товарів інколи може бути досить складним завданням, тому існують різні методи фільтрації, які спрощують цей процес, забезпечуючи користувачам отримання релевантних результатів, адаптованих до їхніх потреб [1]. Система рекомендацій – це тип техніки веброзвідки, яка може щоденно фільтрувати інформацію для користувачів.

З метою вдосконалення процесу пошуку найкращих товарів для клієнта використовуються різноманітні методи, але одним із найефективніших є застосування алгоритмів персоналізованих та неперсоналізованих рекомендацій засобами штучного інтелекту. Ці алгоритми часто враховують історію перегляду, історію покупок та тривалість часу, проведеного на сторінках різних продуктів [2].

Існує значна кількість різноманітних методів та алгоритмів, проте всі вони входять до трьох загальних підходів, а саме:

- фільтрування на основі вмісту [3];
- спільна фільтрація [4];

– гібридні моделі [6].

Аналіз або фільтрація на основі вмісту [3] передбачає рекомендацію продукту користувачам шляхом аналізу характеристик самих продуктів, не покладаючись на дані користувачів. Наприклад, якщо користувач переглянув вебсайт до ноутбуків, система рекомендацій порекомендує інші предмети за схожими категоріями чи типами, або інший ноутбук того ж виробника. Рекомендації, які отримують користувачі, враховують багато факторів, таких як колір, матеріал, категорія продукту та характеристики. Такий підхід може запропонувати багато різних алгоритмів для досягнення цілей, наприклад:

- TF-IDF [7];
- вбудування тесту [8];
- Cosine Similarity [9] тощо.

TF-IDF оцінює важливість кожного слова в тексті на основі кількості разів, коли воно з'являється в тексті, порівняно з кількістю разів, коли воно з'являється в інших статтях основної частини. Також цей метод враховує стоп-слова; або слова, що уже часто зустрічаються у конкретній мові чи то англійська, чи то українська; для виправлення таких слів, або ігнорування. Інверсна частота документа у цьому методі розраховується за такою формулою [7]:

$$IDF(t) = \log \frac{n}{df(t)} \quad (1.1)$$

Вбудування тесту [8] ще один з алгоритмів, який вимірює пов'язаність текстових рядків, дуже добре передає семантичне значення, допомагає з такими задачами як пошук тексту, кластеризація, блоки рекомендацій, класифікація, виявлення аномалій тощо.

Алгоритм Cosine Similarity один з алгоритмів, що може допомогти створити просту систему рекомендацій використовуючи подібність між елементами, по-простому, алгоритм обчислює кут між двома векторами, який визначає схожість між ними. Використовується для порівняння схожості між векторами, отриманими після векторизації тексту. Формула цього алгоритму виглядає так [9]:

$$\cos(\theta) = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}} \quad (1.2)$$

Досить часто у створенні рекомендацій таким підходом використовується одразу декілька алгоритмів, аби досягти більш якісних результатів.

Підхід спільна фільтрація [4] відрізняється від фільтрації на основі вмісту, тим, що збирає й аналізує інформацію про поведінку, дії та вподобання користувачів, щоб спрогнозувати, що їм може сподобатися, знаходячи схожість з іншими користувачами. Цей метод не залежить від здатності машини безпосередньо аналізувати вміст, що дозволяє рекомендувати складні елементи, як-от фільми, без необхідності розуміння самого вмісту. Передумова спільної фільтрації полягає в тому, що якщо користувачі мали схожі смаки в минулому, вони, швидше за все, матимуть подібні смаки в майбутньому. Підхід спільної фільтрації поділяється на два типи: на основі пам'яті та на основі моделі. При цьому, варіант на основі пам'яті включає два основних алгоритми [4]:

- 1) спільна фільтрація між користувачами;
- 2) спільна фільтрація між продуктами.

Алгоритм спільної фільтрації між користувачами аналізує схожість між користувачами для прогнозування продуктів, які можуть сподобатися даному користувачеві. Використовується відстань або коефіцієнт кореляції між рейтингами різних користувачів для визначення їхньої схожості.

Алгоритм спільна фільтрація між продуктами оцінює схожість між елементами, що базується на рейтингах, які користувачі цим елементам виставили. Якщо декілька користувачів високо оцінили певні елементи, ці елементи можуть рекомендуватися іншим користувачам, які оцінили один із цих елементів.

Тип фільтрації на основі моделі використовує статистичні алгоритми та методи машинного навчання для створення і вдосконалення рекомендацій. Основна ідея цього підходу полягає в тому, щоб вивчити загальну модель користувацьких уподобань на основі історії їхніх взаємодій з продуктами, що

дозволяє робити прогнози для нових даних. Цей тип може використовувати багато різних алгоритмів, як-от: кластеризація – K-means, матрична факторизація – SVD, PMF, NMF або глибоке навчання [5]. Ці алгоритми допомагають покращити точність рекомендацій, адаптуючи їх до унікальних потреб кожного користувача.

Найпростішим з раніше перерахованих алгоритмів є неконтрольований та ітераційний алгоритм, який використовується в аналізі даних та машинному навчанні кластеризації – k-means (k-середніх). Основна ідея полягає в тому, щоб розділити набір даних на K груп або кластерів, таким чином, щоб об'єкти всередині кожного кластера були максимально схожими між собою, а об'єкти з різних кластерів були максимально відмінними. Процес роботи методу включає такі кроки [10]:

- вибір випадкових точок з набору даних за початкові центроїди для кожного кластера;
- після кожна точка призначається до кластера, який має найближчий центроїд;
- потім відбувається призначення всіх точок до кластерів, центроїди кожного кластера перераховуються, вираховуючи середнє значення всіх точок у кожному кластері;
- цей алгоритм є ітераційним, тому повторення третього кроку відбувається до тих пір, поки центроїди стабілізуються і не змінюють своє положення;
- коли всі центроїди стали сталими робота алгоритму завершується, а кожна точка даних має бути призначена до кластера.

K Means є швидшим порівняно з іншими методами кластеризації. Це забезпечує міцний зв'язок між точками даних. Проте його не слід використовувати, якщо точки даних перекриваються.

Іншим складнішим, але не менш потужним є метод матричної факторизації – SVD [11]. Такий метод часто використовується для рекомендацій на основі факторів (якісних характеристик), які відображають взаємодію між користувачами та продуктами. Основна ідея полягає у тому, щоб розкласти

матрицю взаємодій між користувачами та продуктами на добуток трьох матриць: U , S і V . Формула роботи цього методу подана нижче:

$$M = U\Sigma V^t \quad (1.3)$$

де матриця M це початкова матриця, яка буде розкладатися; U – ліва сингулярна матриця; Σ - діагональна матриця, що містить власні значення та V , тобто права сингулярна матриця.

Основною перевагою SVD є те, що такий метод може працювати з розрідженими даними, тобто з даними, в яких багато значень відсутні. Також він може допомогти зменшити розмірність даних, використовуючи лише найважливіші фактори, що може поліпшити ефективність обробки та знизити ризик перенавчання моделі.

Метод глибокого навчання став одним із найпопулярніших в останній час. Головна ідея полягає в тому, щоб побудувати багатошарові нейронні мережі, які мають кілька прихованих шарів, спроможних автоматично вивчати корисні функції або представлення даних з вхідних даних. Цей метод підвищує точність рекомендацій і персоналізацію шляхом автоматичного вивчення шаблонів і представлень із великих наборів даних. Традиційні алгоритми рекомендацій, такі як спільне фільтрування та методи на основі вмісту, часто стикаються з проблемами під час захоплення складних шаблонів і прихованих особливостей у даних. Поглиблене навчання є кращим у цьому контексті [12].

Підхід гібридної моделі [6] поєднують переваги спільної фільтрації та фільтрації на основі вмісту, створюючи більшу стійкість та точну систему рекомендацій. Вони враховують як поведінку користувачів, так і характеристики продуктів, що дозволяють генерувати персоналізовані пропозиції.

Проаналізувавши наведені підходи, їх переваги та недоліки, для розробки методу рекомендацій обрано: використовувати підхід фільтрації на основі контексту з методами машинного навчання TF-IDF та косинусу подібності.

1.2 Огляд існуючих засобів реалізації рекомендацій наступних покупок

В епоху розквіту штучного інтелекту та машинного навчання, рекомендаційні системи стають не тільки важливим, але й невід'ємним елементом сучасного електронного бізнесу. Завдяки таким системам користувачі отримують персоналізовані рекомендації, що значно полегшує їхній процес вибору та збільшує задоволення від покупок. У цьому контексті велике значення мають інструменти для створення рекомендацій, серед яких виокремлюються різноманітні бібліотеки та фреймворки.

Існує безліч бібліотек, які можна використовувати для створення методів рекомендацій. Деякі з них спеціалізуються на певних типах рекомендацій, таких як фільмів або музики, тоді як інші надають загальні інструменти для створення рекомендацій з різноманітними видами даних. До найпопулярніших бібліотек у цій сфері входять:

- Surprise [14];
- LightFM [15];
- Implicit [16];
- TensorFlow Recommenders [17];
- OpenAI, а саме `embedding_utils` [18];
- Mahout [19];
- `scikit-learn` [20].

Surprise [14] належить до сімейства бібліотек Python, створених на основі `scikit-learn`, розроблених спеціально для побудови систем рекомендацій на основі явних рейтингових даних. Найчастіше використовується для створення та роботи з рекомендаціями спільної фільтрації. Належачи частиною екосистеми `scikit`, Surprise пропонує зручний досвід для розробників, які звикли працювати з бібліотеками на основі `scikit`, такими як `scikit-learn`. Крім того, Surprise пропонує низку алгоритмів, починаючи від методів на основі пам'яті, таких як `k-means`, до `SVD`, `NMF` та інших. Бібліотека полегшує роботу, пов'язану з обробкою даних,

надає інструменти для оцінки, аналізу та порівняння продуктивності алгоритмів та багато чого іншого.

Іншою досить популярною у створенні рекомендацій спільної фільтрації та гібридних моделей є бібліотека LightFM [15]. Вона була створена з метою розв'язування певних проблем, які виникають у традиційних рекомендаційних системах, зокрема при великій різноманітності та невеликій кількості даних про користувачів або товари. Також може працювати з різними типами вхідних даних, включаючи як чисельні рейтинги, так і імпліцитні зворотні зв'язки (наприклад, час перегляду сторінок або історію покупок), дає змогу включати метадані елемента та користувача в традиційні алгоритми факторизації матриці.

Ще одною бібліотекою для створення рекомендацій спільної фільтрації для неявних наборів даних є Implicit [16]. Замість явних рейтингів, таких як оцінки, імпліцитні дані можуть включати дії користувачів, такі як перегляди сторінок, покупки або історію прослуховування, що часто доступні в більших кількостях і можуть надавати додаткові інсайти про поведінку користувачів. Бібліотека пропонує реалізацію різноманітних широко використовуваних алгоритмів рекомендацій, розроблених для наборів даних із неявним зворотним зв'язком, таких як: ALS, BPR, LMF та багато інших. Бібліотека оптимізована для швидкої роботи на великих наборах даних і може ефективно використовуватися на мультитядерних системах.

TensorFlow Recommenders (TFRS) [17] — це високорівнева бібліотека від TensorFlow, призначена спеціально для створення складних рекомендаційних систем. Вона дозволяє легко побудувати, експериментувати та служити різним рекомендаційним моделям, інтегрованих із сучасними можливостями машинного навчання, що надаються платформою TensorFlow. Бібліотека побудована на технології Keras і націлена на плавну криву навчання, але також дає гнучкість для створення складних моделей. Бібліотека дозволяє використовувати як прості моделі на основі векторів подібності, так і складніші архітектури з використанням глибокого навчання для врахування різних аспектів поведінки користувачів та характеристик товарів.

Також досить значною є бібліотека на основі платформи OpenAI, що використовує функцію вбудовування через бібліотеку `embedding_utils` [18], вона дозволяє генерувати векторні представлення текстів. Ці векторні представлення можуть використовуватися для різноманітних завдань машинного навчання й аналізу даних, включаючи пошук подібності, кластеризацію і класифікацію текстів, а також створення рекомендацій. Бібліотека містить методи кластеризації `k-means`, а також `t-sne` для зручного пошуку та групування набору даних.

Певною мірою, унікальною бібліотекою для роботи з рекомендаціями від екосистеми Apache Hadoop є Apache Mahout [19]. Ця бібліотека є відкритим проектом, створеним з метою надання вільно доступних імплементацій алгоритмів машинного навчання, особливо з фокусом на обробку великих даних. Основна спеціалізація Mahout пропонує різноманітні алгоритми кластеризації такі, як: `k-Means`, `Fuzzy k-Means`, та `Dirichlet process clustering` або класифікації – `Naive Bayes` і `Random Forests`. Ці методи можуть використовуватися для групування великих наборів даних у сегменти для подальшого аналізу. Бібліотека може інтегруватися з іншими системами Apache, такими як Pig та Hive, що дозволяє аналітикам легко взаємодіяти з даними в екосистемі Hadoop.

Найбільш популярною відкритою бібліотекою для машинного навчання є `scikit-learn` [20]. Вона пропонує широкий спектр алгоритмів для класифікації, регресії, кластеризації, підтримки векторів та інші, що дозволяє розв'язувати різноманітні завдання аналізу даних. Бібліотека дозволяє візуалізувати дані за допомогою графіків; легко інтегрується з іншими популярними бібліотеками Python, такими як NumPy, SciPy та Pandas; надає інструменти для навчання з учителем, такі як методи опорних векторів (SVM), дерева рішень та навчання без учителя, такі як `k-Means` для кластеризації та PCA для зменшення розмірності.

Після ретельного аналізу доступних інструментів для рекомендацій наступних покупок було вирішено використовувати бібліотеку `scikit-learn`. Ця бібліотека виявилася найзручнішою та найбільш відповідною для розробки методу рекомендацій товарів за попередніми замовленнями клієнта. Scikit-learn не лише спрощуватиме процес розробки, але і має широкі можливості для інтеграції

з іншими бібліотеками Python. Крім того, вона надає доступ до таких алгоритмів машинного навчання, як TF-IDF та косинусна подібність, які допомагають у побудові ефективних систем рекомендацій.

1.3 Огляд особливостей застосування рекомендацій наступних покупок для систем електронної торгівлі

В сучасному світі електронна торгівля стала не лише важливим елементом бізнесу, але й невід'ємною складовою повсякденного споживацького досвіду. Щоб забезпечити задоволення клієнтів і збільшити обороти, виробники та роздрібні компанії вкладають значні зусилля в удосконалення систем рекомендацій наступних покупок. Ці системи, в основному побудовані на аналізі купівельних звичок та поведінки користувачів, стають ключовим інструментом для підвищення продажів та покращення користувацького досвіду.

В цей час системи рекомендацій широко застосовуються у практично кожному інтернет-магазині. Серед найбільших у світі гравців у цій галузі можна відзначити Amazon, eBay, Aliexpress, Zappos і Таобао. Українські покупці також активно користуються системами рекомендацій, а найвідоміші українські інтернет-магазини, які пропонують подібні сервіси, включають Rozetka, Epicentr K, Foxtrot та інші.

Amazon [21] є дуже відомим інтернет-магазином, що продає безліч товарів з різних категорій, а також хмарні послуги. У 1994 році цей магазин починав як невеликий онлайн-книжковий магазинчик, але зараз швидко розширив свій асортимент товарів і послуг. Amazon також активно використовує штучний інтелект для вдосконалення своїх послуг. Їх системи рекомендацій використовуються для підбору товарів, що найбільш відповідають індивідуальним потребам користувачів, підвищуючи зручність покупок та збільшуючи продажі. Завдяки великому асортименту – неперсоналізовані рекомендації для нових клієнтів продемонструють такі категорії: модні тенденції

які вам сподобаються, тут починається рай для гравців, оновіть свій простір тощо. Приклад рекомендацій подано на рис.1.1.

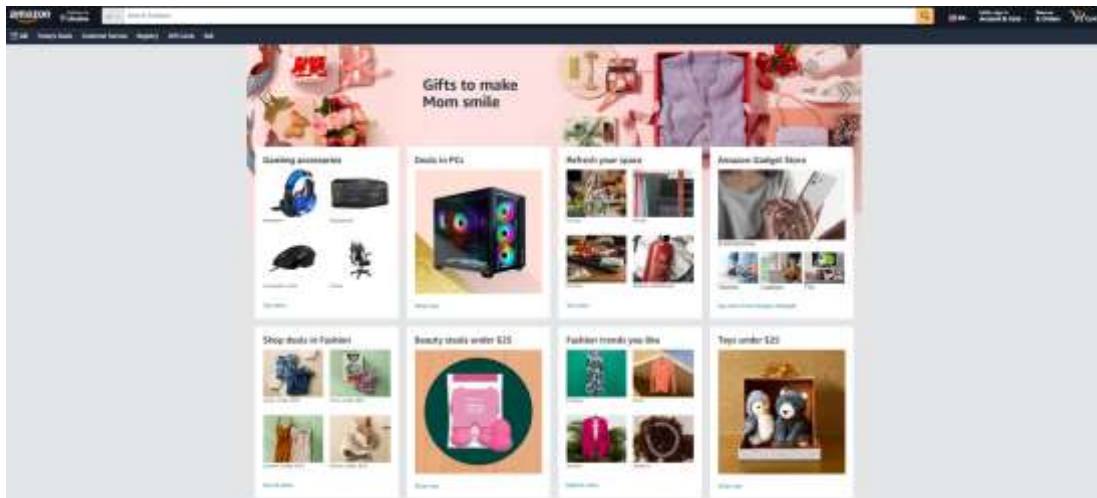


Рисунок 1.1 – Рекомендації у магазині Amazon [21]

Іншою популярною онлайн-торговельною платформою є китайський магазин Aliexpress. Основна особливість AliExpress [22] - це широкий асортимент товарів за доступними цінами. Там можна знайти практично будь-який товар, від електроніки і одягу до аксесуарів для автомобілів та товарів для побуту. Зверху головної сторінки представлено блок рекомендацій, що одразу привертає увагу, який охоплює такі рекомендації як: товари з безкоштовною доставкою, три речі з ціною від 1.79 доларів, кращі послуги та вибрані товари на вибір, більше речей що вам сподобаються, популярні товари тощо. Також частина рекомендацій схожих товарів до товару, що переглядається досить часто висвітлюється трохи нижче. Блок рекомендацій магазину представлено нижче на рис.1.2.

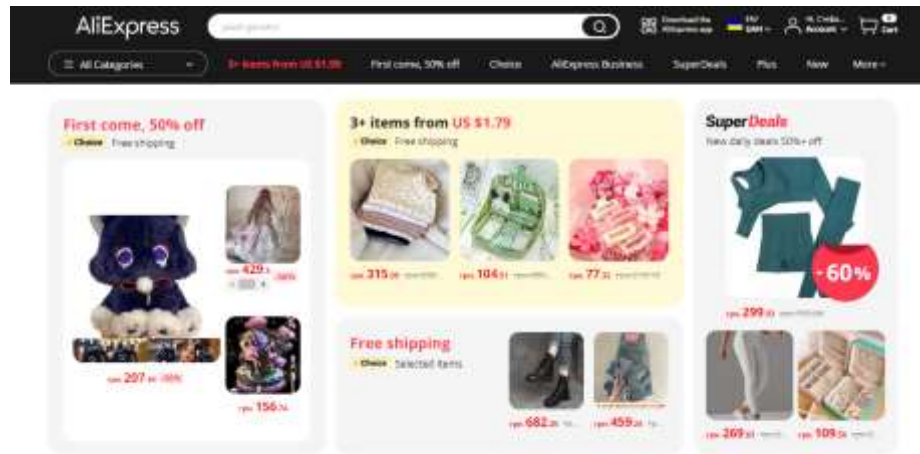


Рисунок 1.2 – Блок рекомендацій у магазині Aliexpress [22]

Відомим українським прикладом магазинів широкого асортименту є вебмагазин Rozetka. Заснований у 2005 році, Rozetka [23] спеціалізується на продажу електроніки, побутової техніки, комп'ютерної та мобільної техніки, товарів для дому, спорту, краси та багатьох інших категорій. Основною метою Rozetka є забезпечення клієнтам широкого асортименту високоякісних товарів та високого рівня обслуговування. Вони пропонують багато варіантів оплати та доставки, включаючи кур'єрську доставку, самовивіз з пунктів видачі та поштової відправки. Головна сторінка майже повністю переповнена різними типами рекомендацій: від рекомендації на основі переглядів до акційних пропозиції, від більше товарів для вибору до гарячих новинок. Приклад блоків рекомендацій як персоналізованих, так і неперсоналізованих подано нижче на рисунку 1.3.

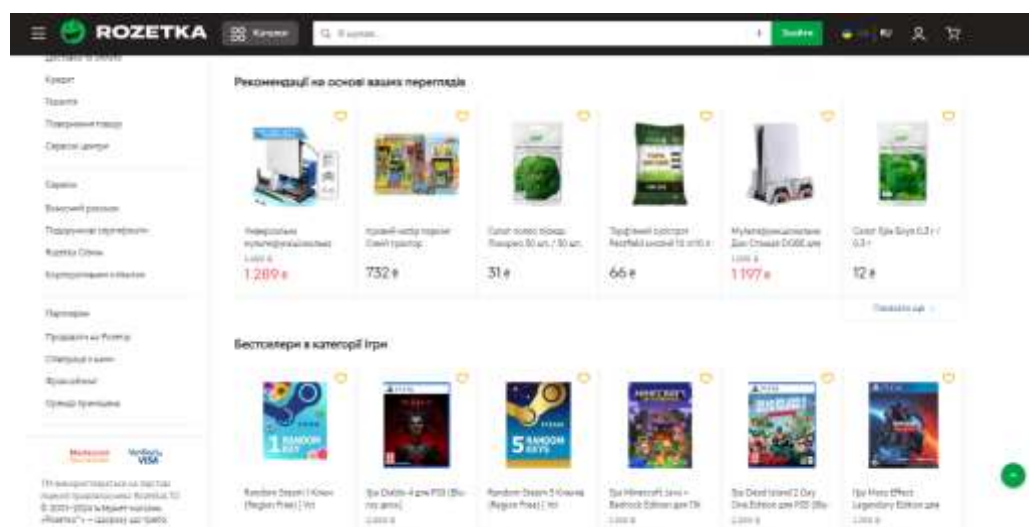


Рисунок 1.3 – Рекомендації, що подані у магазині Rozetka [23]

Іншим прикладом інтернет-магазину українського ринку є магазин Foxtrot. Foxtrot [24] – це один з найбільших роздрібних мереж українського ринку, що спеціалізується на продажі електроніки, побутової техніки, гаджетів та комп'ютерного обладнання. Цей інтернет-магазин пропонує зручну систему пошуку та фільтрації товарів, швидку доставку та різноманітні способи оплати, що дозволяє клієнтам зручно здійснювати покупки онлайн. Магазин відразу привернув увагу користувачів через різноманітні категорії: найкращі цінні пропозиції, тільки до 22 квітня, спеціально для вас, хіти продаж (рис.1.4).

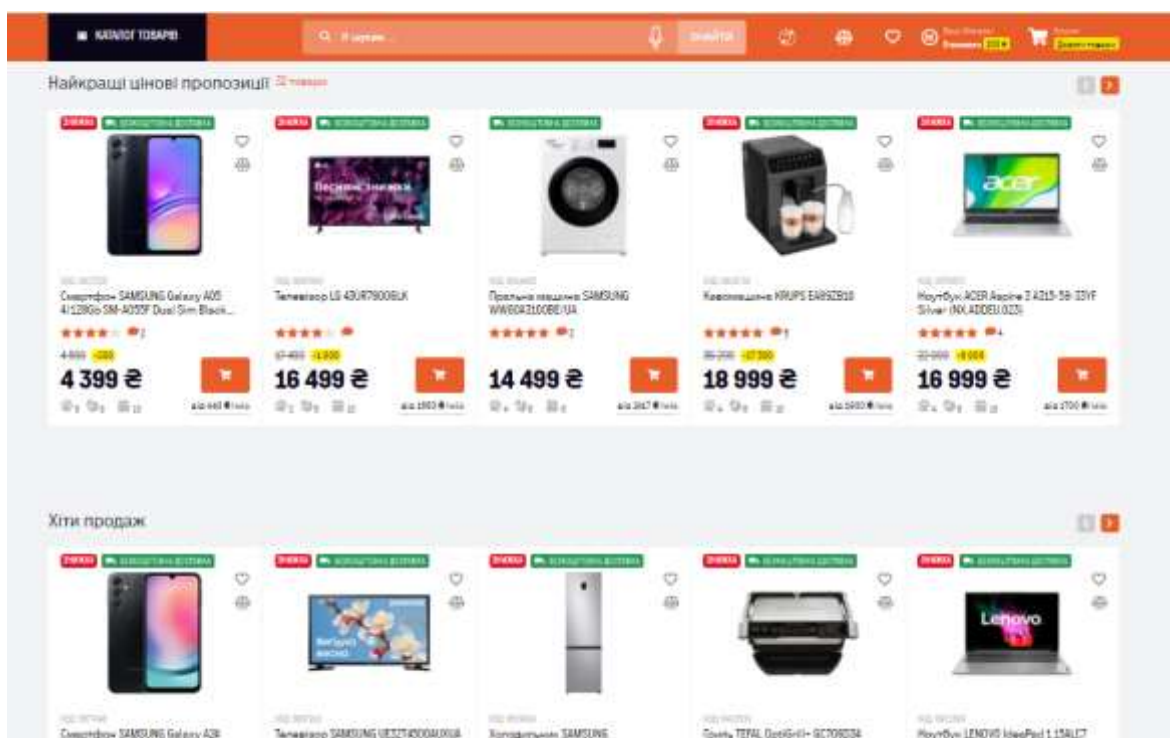


Рисунок 1.4 – Блоки рекомендацій магазину Foxtrot [24]

Отже, розглянувши особливості використання вбудованих рекомендацій у існуючих системах, вирішено створювати блок рекомендацій у верхній частині вебсистеми. Вбудований блок рекомендацій буде використовуватися для покращення доходу власника вебсистеми та значного підвищення конверсії, які сприятимуть зростанню кількості покупок та залученню користувачів до додаткових послуг чи продуктів.

1.4 Мета та завдання кваліфікаційної роботи

Проаналізувавши існуючі підходи до персоналізованого прогнозування наступних покупок, запропоновано мету кваліфікаційної роботи.

Отже, метою кваліфікаційної роботи бакалавра є покращення ефективності отримання персоналізованих рекомендацій за аналізом попередньої купівельної поведінки клієнта засобами машинного навчання для вебсистеми ігрових консолей та аксесуарів. Для досягнення поставленої мети потрібно реалізувати наступні завдання:

- провести аналіз предметної області для задачі отримання персоналізованих рекомендацій за аналізом попередньої купівельної поведінки клієнта засобами машинного навчання для вебсистеми ігрових консолей та аксесуарів;

- виконати аналіз способів формування рекомендацій наступних покупок для користувача;

- виконати огляд та аналіз існуючих рішень щодо реалізації подібних задач; розробити метод рекомендацій за аналізом попередньої купівельної поведінки клієнта засобами машинного навчання;

- спроектувати структуру вебсистеми ігрових консолей та аксесуарів, а також базу даних для даної системи та методу; обрати сучасні засоби реалізації методу рекомендацій за аналізом попередньої купівельної поведінки клієнта засобами машинного навчання для вебсистеми ігрових консолей та аксесуарів;

- реалізувати вебсистему ігрових консолей та аксесуарів та запропонований метод;

- протестувати програмну реалізацію на коректність роботи всіх функцій;

- показати ефективність системи з запропонованим методом.

Розділ 2 Метод рекомендацій товарів за аналізом попередньої купівельної поведінки

Оскільки системи рекомендацій у певній мірі є веброзвідкою, яка допомагає повсякчасно фільтрувати інформацію для користувачів, створення системи чи навіть методу пропозицій досить сильно допоможе покращити персоналізацію користувацького досвіду. Зі створенням рекомендацій товарів за аналізом попередньої купівельної поведінки може найкраще впоратися фільтрація на основі вмісту, адже вона зосереджується більше на клієнті, його попередніх замовленнях товарів та продуктах, ніж спільна фільтрація, яка створює рекомендації на основі схожості вподобань обраного клієнта з іншими.

Відправним пунктом для удосконалення стратегії рекомендацій товарів є аналіз попередньої купівельної поведінки клієнта. Цей метод дозволяє не лише спрогнозувати потенційні покупки, але й створює можливість персоналізованих рекомендацій, які точно відповідають індивідуальним смакам та потребам кожного клієнта. Аналізуючи відповідні категорії товарів, покупки, а також частоту та обсяг покупок, можна отримати унікальне уявлення про вподобання клієнта та вивести на передній план товари, які найбільш ймовірно привернуть їх увагу. Такий підхід не лише підвищує ефективність маркетингових кампаній, але й сприяє покращенню користувацького досвіду, забезпечуючи клієнтів із персоналізованим асортиментом товарів, що відповідають їхнім уподобанням та потребам.

Отже, метод рекомендацій товарів за аналізом попередньої купівельної поведінки є важливим інструментом для збільшення продажів, покращення користувацького досвіду та підвищення лояльності клієнтів. Цей підхід допоможе вебсистемі надати персоналізовані рекомендації для кожного клієнта та привернути увагу до товарів, які раніше могли залишитися непоміченими. Це сприятиме збільшенню загального обсягу продажів та підвищить ефективність маркетингових стратегій. Крім того, індивідуальний підхід у рекомендаціях може

стимулювати клієнтів до частіших та більших покупок, що сприятиме розвитку вебсайту.

Аби візуалізувати отримані рекомендації потрібно провести кілька додаткових кроків, а саме: спроектувати відповідну вебсистему з використанням сучасних технологій, яка буде вбудовувати метод рекомендацій та провести тести ефективності системи з методом та без. Створення вебсистеми може включати створення бази даних для зберігання інформації для коректної роботи вебсистеми та методу тощо. З різних тематик обрано створення вебсистеми ігрових консолей та їх аксесуарів. Окрім створення необхідної вебсистеми також важливо інтегрувати розроблений метод рекомендацій безпосередньо у вебсистему, зазвичай це вимагає змін у коді або використання спеціалізованих бібліотек та інструментів для реалізації алгоритму рекомендацій. Згідно з метою кваліфікаційної роботи важливо також провести тести для порівняння ефективності вебсистеми з методом рекомендацій та без нього. Зібрати дані про взаємодію користувачів з вебсистемою та проаналізувати їх, щоб визначити вплив методу на ключові метрики продажів та користувацького досвіду.

2.1 Метод рекомендацій товарів за аналізом попередньої купівельної поведінки засобами машинного навчання

Для створення методу рекомендацій товарів запропоновано використовувати фільтрацію на основі контексту. Метод створення пропозицій товарів за аналізом попередньої купівельної поведінки користувача передбачає два основних етапи, а саме: 1) навчання моделі та 2) тестування моделі рекомендацій. Перший етап як і другий працюють із завантаженням та обробкою набору даних, вилученням значень, що можуть містити NaN та від'ємні дані.

Вхідними даними для аналізу коректності роботи методу будуть розподілені на навчальні та тестувальні дані. Кожен наступний крок після завантаження та обробки даних, має складатися з роботи над даними, як метод TF-IDF та обчислення косинусної схожості. Відповідно до цього, робота з наборами

даних матиме декілька етапів, перш ніж можна всі значення подавати до безпосередньо методу рекомендацій. Для правильної обробки вхідні дані повинні містити декілька наборів, для найкращих результатів це повинні бути набори з попередніми замовленнями клієнта, набір даних з продуктами для пошуку схожості між певними даними товарів за допомогою лінійного ядра та можливо інші.

Таким чином, вхідні дані для навчання моделі та тестування моделі будуть розділені. Для відмінних результатів передбачається наявність достатньо великої кількості даних, що можна використовувати – через, що відбудеться покращення можливих отриманих результатів, тобто вихідних даних. Нижче на рисунку 2.1 подано схему роботи методу рекомендацій товарів за аналізом попередньої купівельної поведінки:

Вхідними даними методу рекомендацій товарів за аналізом попередньої купівельної поведінки є два масиви даних з такими необхідними значеннями як: попередні товари користувачів та товари, що використовуються для порівняння. Масиви даних розділюються на два набори, за одним з наборів відбувається навчання моделі, тоді як за другим безпосередньо відбудеться тестування, для подальшого виведення результатів на вебсистему. Дані, що тестуються, отримуються із бази даних системи.

Етап перевірки та обробки даних у роботі методу є дуже важливим аби отримати коректні результати без можливих помилок та перекручень. Важливо враховувати, що якість вхідних даних напряму впливає на ефективність та точність роботи методу. Під час цього етапу проводиться перевірка на наявність пропусків, аномалій та некоректних значень у даних. Додатково може здійснюватися попередня обробка даних, така як видалення дублікатів, нормалізація значень, виправлення помилок та інші операції, що допомагають підготувати дані для подальшого аналізу та використання у методі рекомендацій.

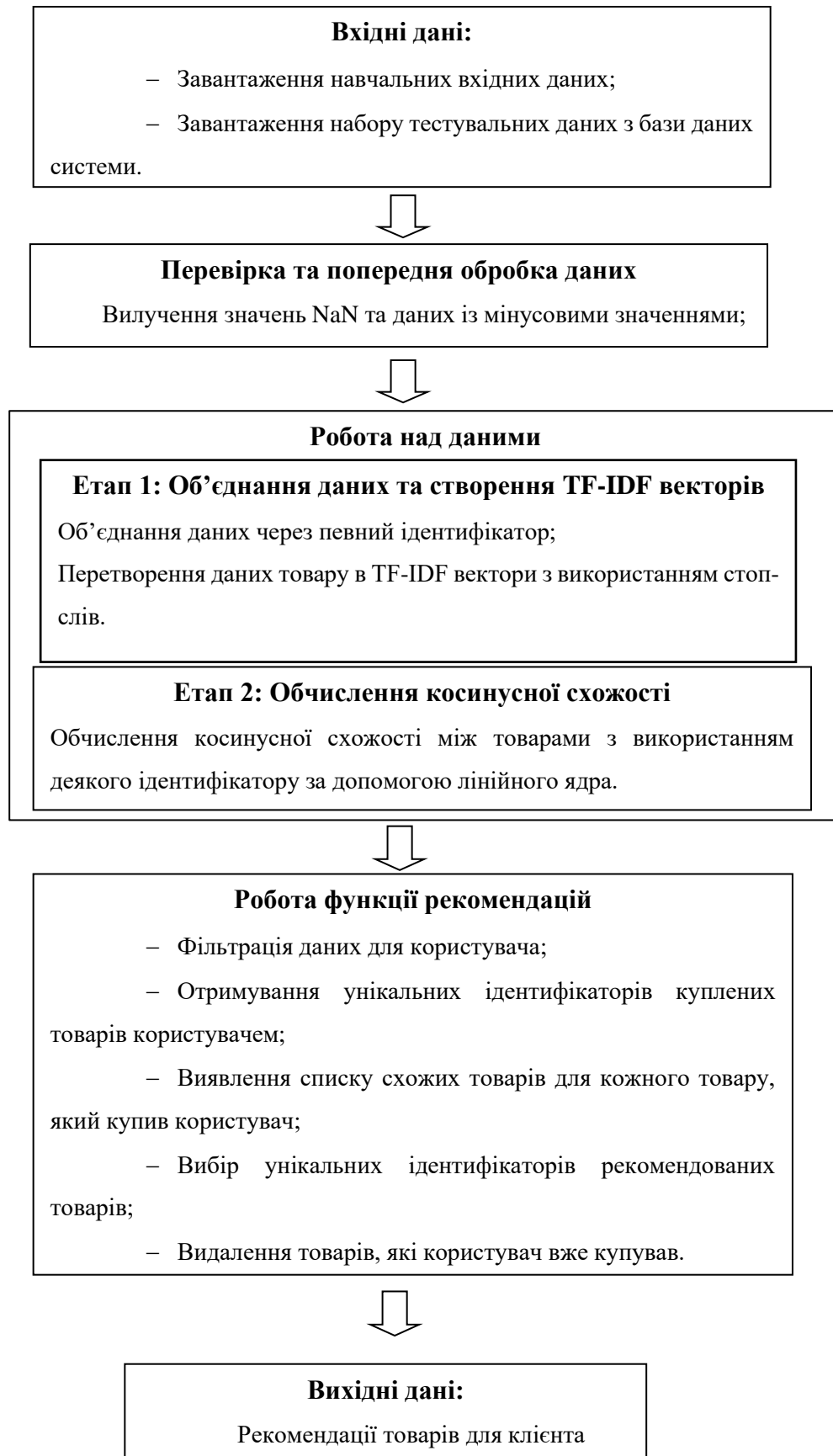


Рисунок 2.1 – Схема методу рекомендацій товарів за аналізом попередньої купівельної поведінки клієнта

Робота з підготовленими даними у першому етапі відбувається у два головних кроки, а саме: об'єднання даних двох наборів у один повний та комплексний набір, який включає інформацію як про замовлення, так і про товари та другий крок – створення векторів даних за допомогою методу машинного навчання TF-IDF.

Об'єднання даних замовлень та інформації про товари дозволяє аналізувати ці дані в комплексі та здійснювати різноманітні аналітичні операції. За допомогою ідентифікатора товару, об'єднання даних забезпечує зв'язок між замовленнями та конкретними товарами, які були придбані. Це дозволяє проводити аналіз популярності товарів, визначати зв'язки між різними товарами, розуміти купівельні звички користувачів та використовувати цю інформацію для створення ефективної системи рекомендацій.

Другий крок зі створення TF-IDF векторів є дуже важливим кроком у побудові системи рекомендацій на основі текстових даних, наприклад описи товарів. Аббревіатура TF-IDF складається з двох частин: TF та IDF, кожна частина цієї техніки виконує важливу роль у методі створення рекомендацій. Сама ж технологія може перетворювати текстові дані в числовий векторний формат, тоді як кожен опис товару стає вектором числових значень, що представляють важливість кожного слова у тексті. Головною функцією TF є те, що він враховує частоту кожного слова в документі. Формула роботи частоти слова або терміну обчислюється наступним чином:

$$tf(t, d) = \frac{f_{t,d}}{\sum t' \in df_{t',d}} \quad (2.1)$$

де $f_{t,d}$ це кількість разів, коли термін з'являвся у наборі даних та $\sum t' \in df_{t',d}$ загальна кількість термінів у наборі.

Слова, які часто зустрічаються в одному описі товару, але рідко в інших, отримують високу TF вагу, що вказує на їхню важливість для розрізнення між документами. Інша частина IDF відображає обернену частоту документа, що допомагає знизити вагу слів, які зустрічаються у багатьох документах, таких як "це", "то", "адже" тощо, щоб вони не впливали на розрахунок схожості між

документами. Формула роботи IDF, формула 1.1, була подана вище у першому розділі.

Після об'єднання наборів даних у один та створення TF-IDF векторів можна переходити до другого етапу роботи над даними – обчислити схожість між товарами, порівнюючи, наприклад за описами. Обчислення схожості може відбуватися за допомогою обрахування саме косинусної схожості з використанням лінійного ядра.

Після того, як описи товарів були представлені у векторному форматі, обчислюється косинусна схожість між векторами кожної пари товарів. Косинусна схожість є мірою кута між двома векторами в n-вимірному просторі. Такий підхід був названий на честь Евкліда, який був піонером геометрії [26]. Обидва вектори мають бути частиною одного простору внутрішнього добутку, тобто вони мають створювати скаляр через множення внутрішнього добутку. Подібність двох векторів вимірюється косинусом кута між ними. Зазвичай косинусна схожість більше за 0, що означає, що вектори спрямовані у схожі напрямки, тобто вони є схожими. Отримані значення косинусної схожості використовуються для визначення ступеня схожості між кожною парою товарів. Ця інформація надалі використовується вже безпосередньо у функції рекомендацій.

Після роботи над даними оброблені значення передаються у функцію, яка обробляє та видає рекомендації отримуючи індекс користувача. Функція працює в декілька етапів (рис.2.2). Перший етап отримує ідентифікатор користувача, для якого потрібно зробити рекомендації. Цей ідентифікатор використовується для отримання історії попередніх покупок цього користувача.

Після цього дані фільтруються для вибраного користувача, що означає, що з усіх попередніх покупок користувача вибираються тільки ті, які належать до товарів, які він вже придбав. Коли фільтрація товарів завершується починається етап – отримання унікальних ідентифікаторів товарів, які користувач вже придбав. Це потрібно для того, щоб надалі уникнути рекомендацій товарів, які вже є у власності у користувача. Для кожного товару, що придбав користувач,

знаходяться схожі товари на основі раніше отриманих значень обчисленої схожості між ними.

Для кожного товару, який придбав користувач, обираються найкраще 5, хоча можна отримати будь-яку кількість, але обирається кількість лише п'яти з найбільшою схожістю. При виборі перших п'яти рекомендацій зі списку схожих товарів видаляються дублікати та товари, які користувач вже придбав. Нарешті, функція повертає список рекомендованих товарів на основі аналізу попередніх покупок користувача та схожості з іншими товарами.

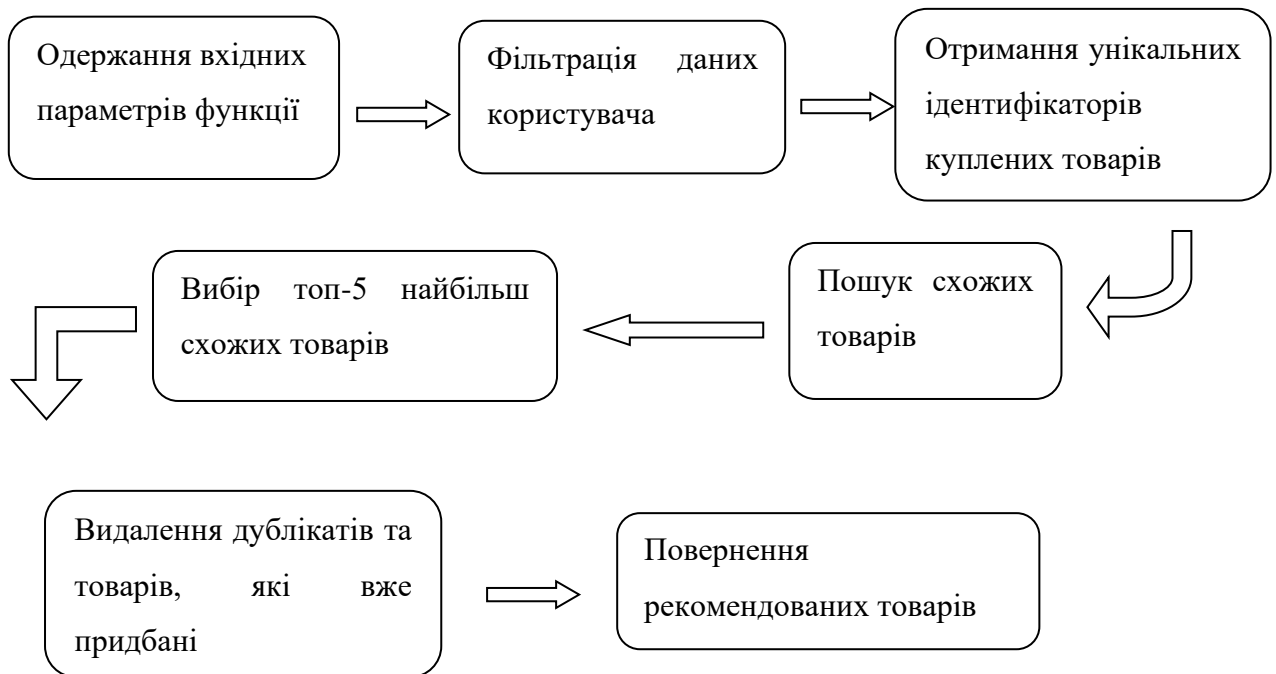


Рисунок 2.2 – Схема роботи функції рекомендацій

Таким чином, було розроблено метод рекомендацій товарів за аналізом попередньої купівельної поведінки клієнта засобами машинного навчання, який призначений для одержання вихідних даних пропозицій користувачу, які надалі можна вивести на клієнтську сторінку у вебсистемі ігрових консолей та аксесуарів.

2.2 Набір даних для навчання моделі

Набір даних для навчання моделі є важливою складовою у розробці та удосконаленні алгоритмів машинного навчання та штучного інтелекту, також він є основою, на якій модель буде навчатися шукати схожі продукти до товарів, що придбав користувач системи. Вибір та підготовка набору даних має вирішальне значення для досягнення успішних результатів в подальших дослідженнях та застосуваннях.

Обраний метод рекомендацій товарів за аналізом попередньої купівельної поведінки клієнта використовує досить специфічні дані для створення персоналізованих рекомендацій.

Враховуючи обмежену різноманітність наявних даних щодо продажу ігрових консолей та їх аксесуарів, було прийнято рішення створити два власних набори даних. Перший набір міститиме інформацію про замовлення клієнтів, який враховуватиме такі дані як:

- номер замовлення (`order_id`);
- ідентифікатор клієнта замовлення – `user_id`;
- ідентифікатор продукту – `product_id`;
- ціна за один продукт – `price`;
- кількість продуктів, що купуються – `count`;
- загальна ціна, за все замовлення – `total_price`;
- статус покупки, що враховує, чи доставлено товари, чи тільки оформлено замовлення – `status`;
- адреса доставки покупки – `address`.

Другий набір даних охоплюватиме всі продукти такі, як: ігри для консолей, ігрові консолі безпосередньо та аксесуари для них. Структуру набору подано нижче у таблиці 2.1:

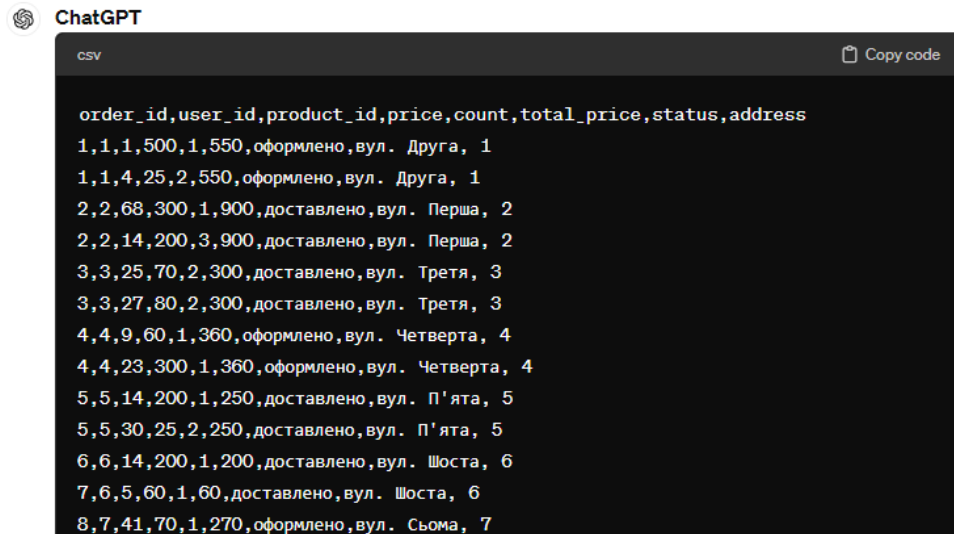
Таблиця 2.1 – Структура другого набору даних

Атрибут	Опис
---------	------

product_id	Числовий ідентифікатор товару
title	Назва товару
description	Короткий опис товару
content	Характеристики товару
preview_image	Перше та головне фото товару
price	Ціна за одиницю товару
count	Кількість товару, що є на складі
category_id	Числовий ідентифікатор категорії товару
tags	Назви тегів товару
colors	Назви кольорів, що є на товарі
created_at	Час створення товару
updated_at	Час оновлення інформації про товар

Дана структура наборів даних достатня для аналізу та розрахунку персоналізованих рекомендацій, а також дозволить забезпечити коректну роботу методу рекомендацій за аналізом попередньої купівельної поведінки.

Для заповнення достатньої кількості даних вирішено використовувати можливості штучного інтелекту чат-боту від компанії OpenAI – ChatGPT 3.5 [27]. Чат-бот досить добре розв’язав поставлену задачу (рис. 2.3) – створити набори даних на 70 різних товарів та більше ніж 30 замовлень (враховується, що один клієнт може купити від 1 до 3 видів товарів), але оскільки отримання справжніх даних: цін, характеристик, є затратним по ресурсах та часу процесом, більша частина поданої інформації у наборах є штучно створеною – вигаданою.



```

order_id,user_id,product_id,price,count,total_price,status,address
1,1,1,500,1,550,оформлено,вул. Друга, 1
1,1,4,25,2,550,оформлено,вул. Друга, 1
2,2,68,300,1,900,доставлено,вул. Перша, 2
2,2,14,200,3,900,доставлено,вул. Перша, 2
3,3,25,70,2,300,доставлено,вул. Третя, 3
3,3,27,80,2,300,доставлено,вул. Третя, 3
4,4,9,60,1,360,оформлено,вул. Четверта, 4
4,4,23,300,1,360,оформлено,вул. Четверта, 4
5,5,14,200,1,250,доставлено,вул. П'ята, 5
5,5,30,25,2,250,доставлено,вул. П'ята, 5
6,6,14,200,1,200,доставлено,вул. Шоста, 6
7,6,5,60,1,60,доставлено,вул. Шоста, 6
8,7,41,70,1,270,оформлено,вул. Сьома, 7

```

Рисунок 2.3 – Створення наборів даних чатом GPT [27]

Інколи чат може припуститися помилки – пише розділові знаки, для відділення даних не ті, що було запрошено, наприклад було запрошено відділяти всі категорії даних крапка з комою, чи він некоректно відмінює слова, але головну задачу він виконує більш менш коректно. Для навчання методу створено 70 різних товарів та 41 покупку клієнтами (більшість замовлень включають більше ніж один вид товару).

Таким чином, можна зробити висновок, що ретельно підготовлені набори даних дозволяють коректно працювати методу рекомендацій товарів, а також забезпечують точні та зрозумілі рекомендації, що відповідають потребам та уподобанням користувачів.

2.3 Розробка архітектури вебсистеми ігрових консолей та аксесуарів

2.3.1 Проєктування вебінтерфейсу інформаційної системи

Вебсистема ігрових консолей та аксесуарів є програмною реалізацією, яка буде використовувати метод рекомендацій товарів за аналізом попередньої купівельної поведінки клієнта, а також відображати функції, що визначені у меті та завданнях реалізації, а саме – автентифікацію та реєстрацію користувача,

відображення рекомендованих товарів користувачеві відповідно до його минулих замовлень, розділення системи на дві частини, клієнтської та адміністративної.

Схема структури вебсистеми ігрових консолей та аксесуарів складається з трьох основних частин, а саме: бази даних та трьох підсистем (рис.2.4).

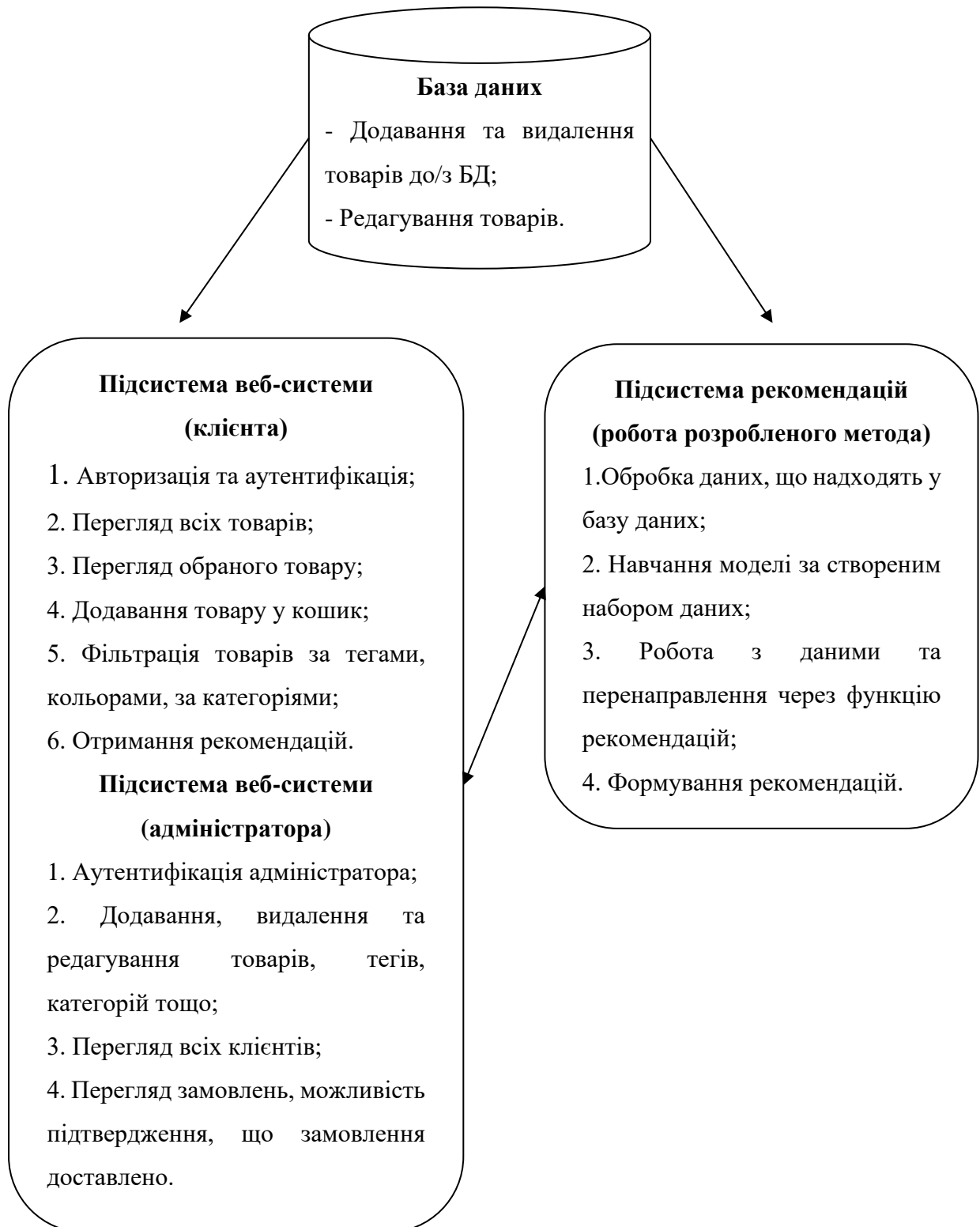


Рисунок 2.4 – Схема структура вебсистеми ігрових консолей та аксесуарів

База даних є однією з основних складових вебсистеми ігрових консолей та аксесуарів і відіграє ключову роль у забезпеченні ефективного функціонування та зберігання необхідної інформації. Завдяки базі даних, що додаватимуться з адміністративної частини будуть зберігати всю обов'язкову інформацію: інформація про користувачів, їх профілі, дані для аутентифікації, інформація про оплату, історія замовлень, товари та пов'язані з ними теги та категорії тощо. Також дані, що зберігаються у базі можуть використовуватися для аналізу даних, створення звітів та статистики щодо продажів, популярності товарів, поведінки користувачів тощо.

Невелика підсистема рекомендації, пов'язана з двома підсистемами вебсистеми ігрових консолей та аксесуарів, використовуватиме розроблений метод для візуалізації пропозицій на стороні клієнта. Відокремлення цієї підсистеми забезпечить гнучкість і модульність системи, дозволяючи легко масштабувати та змінювати рекомендаційні алгоритми без впливу на інші частини системи. Такий підхід сприяє полегшенню розвитку та підтримки системи, а також покращує її продуктивність та взаємодію з користувачами.

Дві підсистеми (для клієнта та адміністратора) використовуватимуть необхідні дані із бази даних для роботи функціонала кожної поставленої задачі у підрозділі 1.4. Робота адміністраторської частини включатиме здебільшого роботу з базою, оскільки панель повинна дозволяти додавати, видаляти та редагувати все, що пов'язано із товарами вебсистеми, точніше робота з категоріями товарів, робота з тегами товарів та робота з кольорами, які є в товарах. Також підсистема адміністратора має включати безпосередню роботу з продуктами, що мають знайтися у системі – ігрові консолі, ігри для ігрових консолей та аксесуари такі, як: чохла, зарядні станції, захисні плівки тощо. Крім того, важливою частиною адміністративної панелі має бути моніторинг, найкраще у графіках, кількості зроблених замовлень, кількості клієнтів, що є у базі даних тощо.

Клієнтський інтерфейс підсистеми вебзастосунку включає численні невеликі, проте критично важливі функції, які спрощують взаємодію користувачів, а саме: фільтрація товарів за тегами, категоріями та кольорами, можливість переглянути сторінку товару з його характеристиками тощо. Ці функції не тільки покращують загальний досвід користувача, але й забезпечують ефективну та інтуїтивно зрозумілу навігацію по сайту. Також серед ключових опцій клієнтської підсистеми виділяються реалізація кошика покупця та можливість створення замовлень, що значно спрощує процес купівлі. Кошик дозволить користувачам зберігати вибрані товари до моменту покупки, а функція створення замовлення автоматизує збір необхідної інформації для оформлення замовлення, включаючи адресу доставки, що гарантує зручність та ефективність обслуговування клієнтів. Отримана інформація – історія замовлення допоможе й надалі працювати підсистемі рекомендацій та пропонувати користувачу все більш персоналізовані пропозиції товарів. Крім того, функції аутентифікації та авторизації також повинні висвітлюватися. Окрім цього, важливо реалізувати функції аутентифікації та авторизації, які є невіддільною частиною системи безпеки вебзастосунку. Ці функції забезпечуватимуть перевірку особистості користувачів та визначатимуть їхні права доступу до різних функцій сайту, що допоможе запобігати несанкціонованому доступу та забезпечує захист конфіденційної інформації.

Розташування кожної частини, кнопки на сторінках потрібно створити для зручного користування базуючись на досвіді користувача. Вигляд сторінок системи є ключовим елементом у забезпеченні інтуїтивно зрозумілого та зручного користувацького інтерфейсу. Добре організований дизайн не тільки полегшує навігацію по системі, але й підвищує задоволення користувачів від взаємодії з вебсайтом.

Головною сторінкою клієнтської підсистеми є сторінка зі всіма товарами та з функціями фільтрації товарів (рис.2.5). На цій сторінці користувач може переглядати товари, фільтрувати за тегами, категоріями та кольорами, додавати

товар у кошик. З цієї ж сторінки можна вільно переходити на особисту сторінку товару.

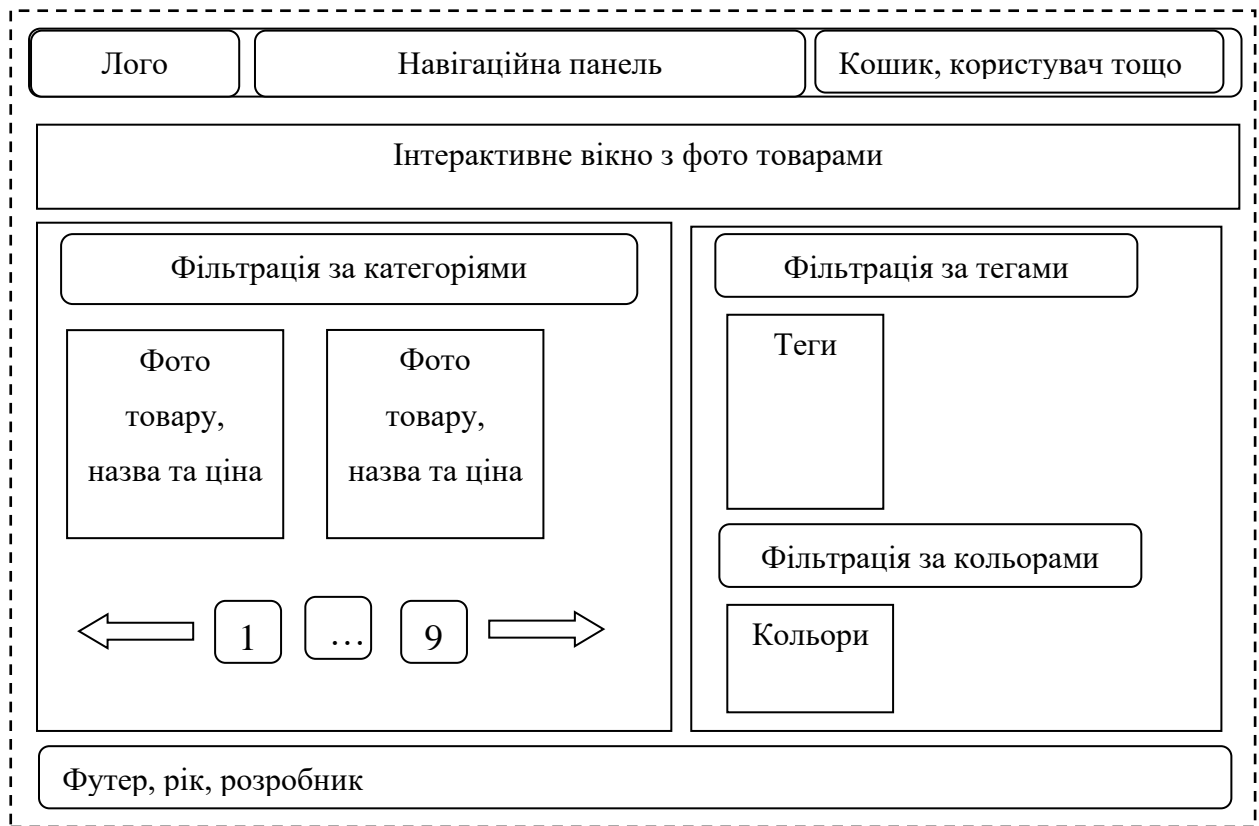


Рисунок 2.5 – Схема сторінки товарів з фільтрацією

Приклад розміщення результату роботи методу рекомендацій товарів за минулими покупками клієнта можна переглянути на головній сторінці клієнтської підсистеми (рис.2.6). Також там розміщено всі товари з бази даних, хедер сторінки з логотипом інтернет-магазину, навігаційними кнопками; футер з роками роботи магазину та інтерактивні вікна з фото товарами.

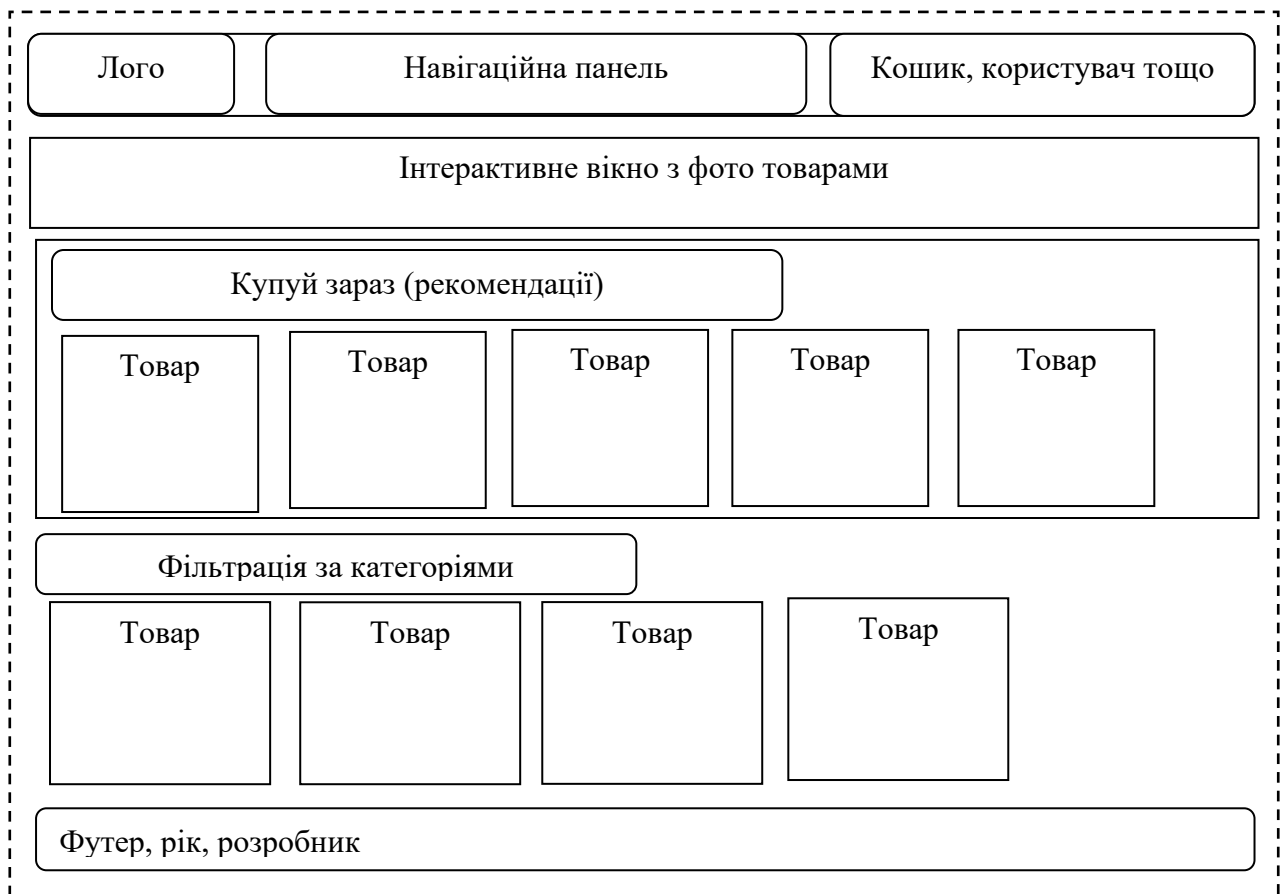


Рисунок 2.6 – Схема структури головної сторінки з рекомендаціями

Окрім клієнтської частини вебсистеми, також свій невеликий функціонал має підсистема адміністратора. Вона працює безпосередньо з даними з бази даних, адміністратор може переглядати товари, що існують, клієнтів системи та створенні замовлення від користувачів, а також додавати, редагувати та видаляти товари та все, що з ними пов'язано за необхідності. Аби краще зрозуміти послідовність взаємодії адміністратора із системою додавання нового продукту, нижче подано діаграму послідовності (рис. 2.7).

Щоб додати новий продукт до бази даних, адміністратор повинен увійти до вебсистеми у якості адміністратора. Для цього необхідно успішно автентифікуватися за допомогою відповідних облікових даних, після чого система перенаправить адміністратора до його особистої панелі. На головній сторінці розташована навігаційна панель зліва, яка надає швидкий доступ до основних розділів системи. Крім того, посередині сторінки розміщені додаткові кнопки, які

дозволяють переглядати аналітичні дані та статистику для зручного аналізу та моніторингу ефективності системи.

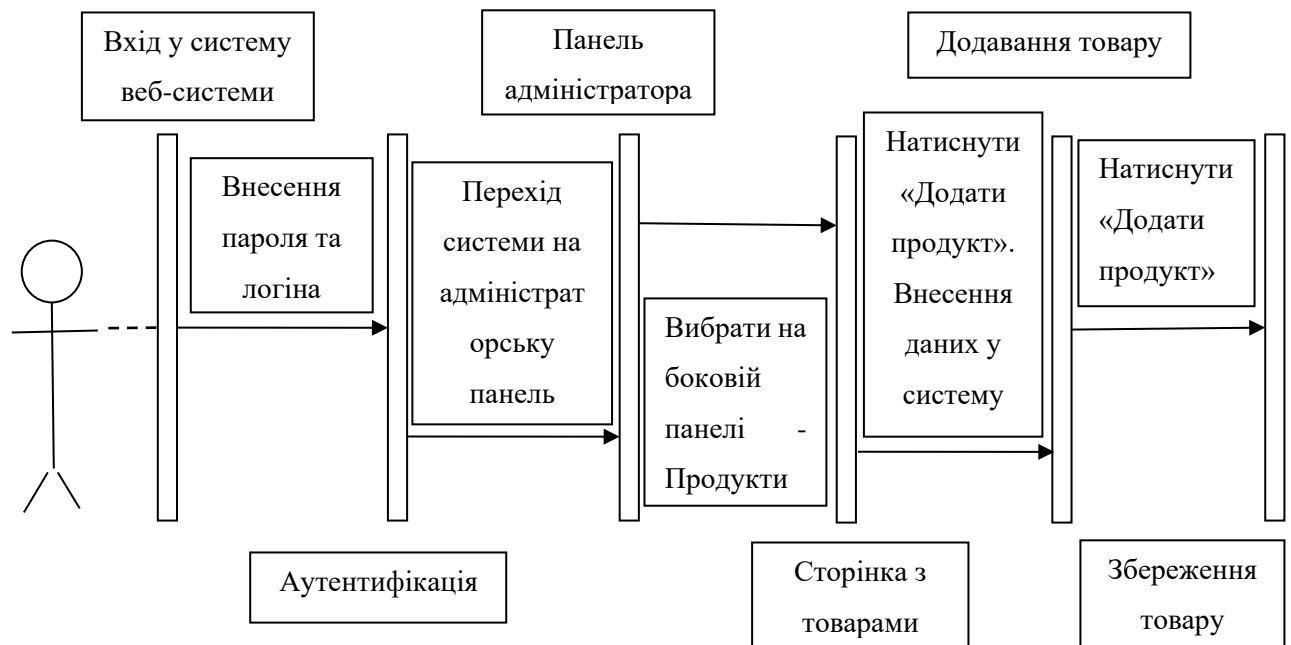


Рисунок 2.7 – Діаграма послідовності роботи адміністратора з вебсистемою

Після цього, щоб додати новий продукт, потрібно перейти до бокової панелі навігації та обрати опцію "Продукти". Одразу відкриється нова сторінка, на якій буде відображений список усіх продуктів, що наявні в системі. Тут адміністратор може знайти кнопку під назвою «Додати продукт» та клацнути на неї, щоб відкрити вікно, де можна внести інформацію про новий продукт. Процес розпочинається з введення назви продукту і може включати додавання опису, зображень, ціни та вибір категорії, тегів, які описують продукт та іншого. Після внесення усіх необхідних даних для збереження змін, натискається кнопка «Додати продукт» і новий продукт вноситься до бази даних.

Підсумовуючи вище подану інформацію, можна зробити висновок, що для ефективного подання методу рекомендацій товарів за аналізом попередньої купівельної поведінки створення правильної вебсистеми є вкрай необхідним. Ця система повинна виконувати всі функції інтернет-магазину ігрових консолей та їх аксесуарів, а саме: подання всіх товарів бази даних, фільтрація товарів за певними

ознаками, додавання товару в кошик та оформлення покупки, також рекомендовано створення адміністративної підсистеми для роботи напряму з базою даних та аналітикою ефективності системи. Метод пропозицій для найкращої наочності потрібно розмістити в клієнтській частині аби користувач отримував персоналізовані рекомендації одразу після входу в систему магазину.

2.3.2 Проектування структури бази даних

Даталогічна модель бази даних є фундаментальним етапом у процесі проектування та розробки інформаційних систем. Вона визначає структуру даних, що зберігаються в базі даних, та взаємозв'язки між ними. Важливість добре спроектованої даталогічної моделі важко переоцінити, оскільки вона забезпечує зрозумілість, ефективність та надійність управління даними.

Створена база даних буде використовуватися для тестування методу рекомендацій товарів за аналізом попередньої купівельної поведінки клієнта. Спроектвана даталогічна модель бази даних складається з дев'яти таблиць (рисунок 2.8): Products, Users, Orders, Tags, Colors, Photos, Categories, Product_Tags, Color_Products.

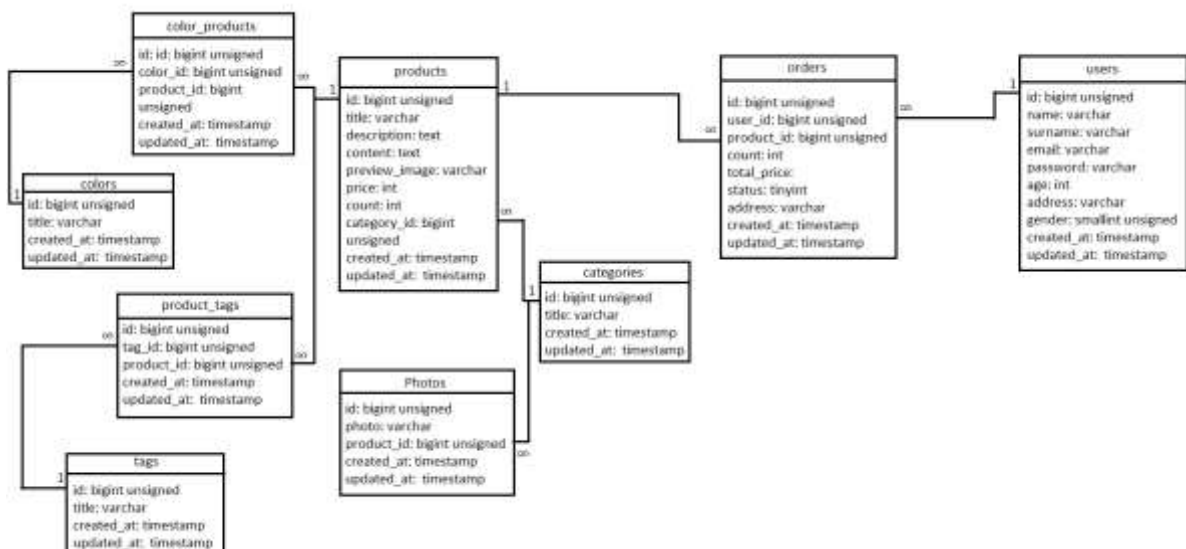


Рисунок 2.8 – Даталогічна модель бази даних

Головними у цій базі дані є три таблиці, а саме – Products, Users, Orders, кожна з цих таблиць необхідна для роботи методу рекомендацій товарів. У таблиці «Products» зберігається інформація про всі товари системи, вона має такі атрибути:

- ID – первинний ключ;
- title – назва товару;
- description – короткий опис товару;
- content – характеристика товару;
- preview_image – перше та головне фото;
- price – ціна, за одиницю товару;
- count – кількість товару, що є;
- category_id – вторинний ключ, посилається на таблицю «Categories»
- created_at – час створення товару (встановлюється автоматично);
- updated_at – час оновлення товару (встановлюється автоматично).

Інша таблиця «Users» працює з інформацією про користувачів інтернет-магазину, здебільшого вся інформація включає: ID – первинний унікальний ключ користувача, ПІБ клієнта, вік та адресу за якою проживає, також у базу даних вноситься інформація про стать клієнта, його пошту та пароль, який буде використовувати клієнт для аутентифікації в системі.

Найбільш важливою таблицею для роботи методу рекомендацій товарів за аналізом попередньої купівельної поведінки клієнта є таблиця з усіма замовленнями користувача системи – «Orders». Ця таблиця складається з таких атрибутів: ID, user_id, product_id, count, total_price, status, address та два атрибути часу – created_at та updated_at (таблиця 2.2).

Таблиця 2.2 – Атрибути таблиці «Orders»

Атрибут	Тип даних	Опис
ID	bigint unsigned	Унікальний первинний ключ
user_id	bigint unsigned	Вторинний ключ, посилається на таблицю «Users»

product_id	bigint unsigned	Вторинний ключ, посилається на таблицю «Products»
count	int	Кількість товару, що купується
total_price	int	Повна ціна замовлення
status	tinyint	Статус доставки
address	varchar (255)	Адреса доставки покупки
created_at	timestamp	Час створення замовлення
updated_at	timestamp	Час оновлення замовлення

Більшість інших таблиць у базі даних відповідають за класифікацію товарів за різними категоріями, тегами та кольорами. Оскільки один товар може бути призначений до багатьох тегів або мати різні кольори, а також один тег може відповідати кільком товарам, було розроблено додаткові проміжні таблиці, такі як "Product_Tags" та "Color_Products". Кожна з цих таблиць використовує вторинні ключі з основної таблиці товарів для встановлення зв'язку між додатковими таблицями кольорів та тегів.

Також потрібною додатковою таблицею у базі даних вебсистеми ігрових консолей та їх аксесуарів є таблиця «Categories». Таблиця категорій в базі даних магазину містить певну структуру категорій, де кожен запис має посилання на свого батька в таблиці з продуктами. Ця таблиця містить такі атрибути:

- ID – первинний ключ у таблиці з категоріями;
- title – назва категорії;
- created_at – дата та час створення запису;
- updated_at – дата та час оновлення запису, що вже існує.

Останньою таблицею є таблиця фотографій («Photos») призначена для збереження більшої кількості фотографій продуктів. Ця таблиця містить поля для збереження шляхів до файлів зображень, а також ідентифікаторів продуктів, з якими фотографії пов'язані.

Отже, у цьому підрозділі було спроектовано структуру даталогічної моделі бази даних вебсистеми ігрових консолей та їх аксесуарів для роботи системи

інтернет-магазину та методу рекомендацій товарів за аналізом попередньої купівельної поведінки клієнта. Спроектowana база даних має дев'ять основних таблиць для повного існування вебсистеми як такої.

2.4 Особливості використання спеціалізованих програмних компонентів

Для ефективної роботи методу рекомендацій товарів у вебсистемі ігрових консолей та їх аксесуарів необхідно використовувати спеціалізовані програмні компоненти. Процес роботи методу можна розбити на кілька етапів:

- розробка методу рекомендацій – етап включає реалізацію методу рекомендацій за допомогою мови програмування Python. У цьому етапі використовуватимуться бібліотеки та методи машинного навчання для аналізу та обробки даних;

- інтеграція методу до серверної-адміністративної частини – після розробки методу його необхідно інтегрувати до серверної частини вебсистеми, для цього може бути використано серверні технології, такі як Laravel у випадку бекенду. Ця інтеграція передбачає створення REST API для взаємодії з фронтендом та виклику методу рекомендацій у Python з серверної сторони;

- вивід результату клієнту на частину фронтенду – тобто необхідно передати результати до фронтенду для відображення для кінцевого користувача. Цей процес зазвичай здійснюється за допомогою фреймворків, для клієнтської частини обрано використовувати фреймворк JavaScript – Vue.js.

Для розробки методу рекомендацій обрано головною бібліотекою Scikit-learn, яка може забезпечити всі необхідні методи машинного навчання, також містить багато інструментів для роботи з даними. У методі рекомендацій використовуються алгоритми машинного навчання TfidfVectorizer для перетворення текстових даних у числовий формат та linear_kernel для обчислення косинусної схожості між описами товарів.

Окрім Scikit-learn також використовуються бібліотеки Pandas, NLTK та SQLAlchemy та інші. Кожна з них забезпечує свою роботу з даними.

Бібліотека Pandas використовується в методі для роботи з даними у форматі таблиць: завантаження, обробка та аналіз даних здійснюються з її допомогою. Спочатку, для навчання моделі використовується створюється два об'єкти, що будуть приймати наробки даних та зчитувати їх створюючи таблиці зі значеннями через бібліотеку Pandas, вона ж забезпечить об'єднання двох наборів даних у один через метод *merge*. Також ця стандартна бібліотека використовуватиметься для тестування моделі – зчитування та завантаження даних з бази даних вебсистеми через метод *read_sql*. Окрім завантаження та об'єднання даних бібліотека також застосовується для перетворення деяких даних в інші типи через метод *astype*, також за необхідності відбувається видалення дублікатів через *drop_duplicates*.

Для обробки природної мови та роботи з текстовими даними використовується бібліотека NLTK. У методі вона забезпечує обробку текстових даних перед їх використанням у моделі рекомендацій, а саме для роботи зі стоп-словами. У роботі методу ця бібліотека відкриває файл зі списком стоп-слів для української та англійської мов за допомогою *open*, а потім завантажує ці слова у список *stopwords*. Ці слова слугують для відфільтрування зайвих слів під час обробки текстових даних. Список стоп-слів використовується у створенні об'єкта *TfidfVectorizer*. При використанні цього об'єкту для створення TF-IDF векторів, слова, які знаходяться у списку стоп-слів, будуть відфільтруватися та не враховуватимуться під час обчислення TF-IDF значень.

Також важливою бібліотекою є SQLAlchemy, вона використовується для з'єднання з базою даних і виконання запитів для отримання даних. SQLAlchemy – це досить сучасна бібліотека Python, яка забезпечує SQL-взаємодію з різними типами баз даних за допомогою об'єктно-реляційного відображення та конструкцій SQL-виразів. У методі вона може використовуватися для виклику деяких функцій, наприклад *create_engine* для створення об'єкту-двигка, який встановлює з'єднання з базою даних MySQL та допомагає взаємодіяти з нею.

Також у методі можна використати об'єкт-движок – *engine* аби виконати SQL-запити до бази даних через функцію бібліотеки Pandas – *read_sql*. Ця функція використовує об'єкт-движок для виконання запиту і повертає результат у вигляді об'єкта DataFrame бібліотеки Pandas.

Функція рекомендацій надалі базується на аналізі схожості між описами товарів, використовуючи TF-IDF векторизацію. Після обчислення схожості між товарами, для кожного користувача вибираються топ-5 найбільш схожих товарів на основі його попередніх покупок.

Для того, щоб інтегрувати метод до серверної частини, а потім до клієнтської, потрібно спочатку створити REST API з використанням Laravel. Ця операція, в основному, вимагає використання структури MVC Laravel, зокрема контролерів та маршрутів. У Laravel існують спеціалізовані бібліотеки, які спрощують роботу з кодом на мові Python.

Для відображення результату методу правильно, необхідно створити контролер, який буде зв'язаний з файлом методу та буде працювати з ним. У контролері створюється функція, яка буде використовуватися у маршруті частини API. Для виконання коду мовою програмування Python використовується метод *process*. Компонент Process запускає команди в підпроцесах.

При отриманні запиту контролер викличе метод рекомендацій, отримає результати та поверне їх у відповідь на запит. Для роботи з файлом мови програмування Python, а також для зчитування та виконання методу, можна використовувати спеціалізовану бібліотеку Symfony.

Після під'єднання файлу до серверної частини, потрібно створити компоненти, які будуть взаємодіяти з цим REST API на стороні клієнта. Ці компоненти будуть відправляти запити до сервера для отримання рекомендацій та відображення їх на вебсторінці. Для цього можна використати бібліотеку Axios, яка запустить сценарій з адміністративної частини на клієнтську. Отримані дані будуть оброблятися через цикл *v-for* та виводитимуться у відповідних блоках. Axios може використовувати методи *get* або *post* для взаємодії з маршрутами Laravel.

Отже, було розглянуто особливості використання спеціалізованих програмних компонентів, а саме бібліотек та деяких методів для створення та інтеграції методу рекомендацій у вебсистему ігрових консолей для адміністративної частини на фреймворку Laravel, клієнтської частини – Vue.js та мови програмування Python.

2.5 Спосіб оцінювання ефективності вебсистеми з запропонованим методом

Протягом останніх кількох років стався значний сплеск інтересу до систем рекомендацій, особливо після того, як вони стали невіддільною частиною практично всіх онлайн-платформ для роздрібної торгівлі. Оцінка та аналіз ефективності системи рекомендацій щодо продуктів має важливе значення для роздрібних брендів, щоб оцінити її ефективність і впровадити вдосконалення.

Вклад рекомендацій в інтернет-магазини є невідворотним, лише магазин Amazon через впровадження рекомендацій отримує 35% зі всіх покупок. У 2015 році Amazon зіштовхнувся зі жахливою конкуренцією, тому вони вирішили запровадити Best Buy [28] – ефективну систему рекомендацій щодо продуктів на основі штучного інтелекту. Ця система об'єднувала дані як онлайн, так і офлайн, пропонуючи клієнтам персоналізовані рекомендації, адаптовані до конкретних магазинів через їх застосунок. Наразі цей застосунок не лише пропонує продукти, але й вказує на їх наявність у найближчих магазинах, заохочуючи клієнтів відвідувати фізичні місця. Аналіз Best Buy включає такі фактори, як: попередні покупки, поведінка вебпереглядача та використання таких функцій, як «Моя найкраща покупка», для надання персоналізованих пропозицій.

Оскільки робота програми здебільшого локальна, тоді для оцінки ефективності вебсистеми, що використовує метод рекомендацій, буде застосовуватися різні офлайн-метрики. Наприклад, аналіз середньої кількості покупок та середній чек, а також використовувати метрики точності, такі як середня абсолютна похибка або середній квадратичний відхил. Крім того, можна

враховувати нормалізований дисконтований кумулятивний приріст (NDCG), який вимірює якість порядку рекомендованих елементів, а також їх релевантність для користувача. Також для перевірки ефективності методу можна використовувати таблиці з виведеними даними на систему адміністратора. Наприклад, за тиждень можна аналізувати такі показники, як кількість зареєстрованих користувачів у системі. Оскільки створення замовлення або покупка можлива лише для авторизованих користувачів, цей показник дозволить нам зрозуміти, наскільки активно користувачі реєструються та взаємодіють з системою. Також важливо вивчати загальну кількість здійснених покупок, а також інші ключові метрики, що характеризують активність користувачів та ефективність рекомендаційної системи. Такий підхід дозволить отримати повну картину ефективності рекомендаційної системи та зрозуміти, наскільки вона задовольняє потреби користувачів і впливає на їхню покупкову поведінку.

Середня абсолютна похибка (MAE) [29] – це одна з ключових метрик, яка використовується для оцінки точності прогнозів в системах рекомендацій. Вона вимірює різницю між фактичними й прогнозованими значеннями та надає загальне уявлення про те, наскільки точно система рекомендує продукти користувачам. Ця метрика розраховується за такою формулою:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (2.2)$$

де n – це кількість спостережень (прогнозів), y_i – фактичне значення, а \hat{y}_i – прогнозоване значення.

MAE вимірюється в тих же одиницях, що і вихідні дані, що робить її легко інтерпретованою. Чим менше значення MAE, тим краще прогнози моделі. Наприклад, якщо MAE дорівнює 0.5, це означає, що прогнозовані значення в середньому відрізняються на 0.5 від фактичних значень. Чим ближче MAE до нуля, тим краще точність моделі.

Іншою корисною метрикою є середній квадратичний відхил або RMSE [30]. Метрика RMSE використовується для оцінювання точності моделей прогнозування або рекомендаційних систем. Вона вимірює різницю між фактичними й прогнозованими значеннями та виражається у тих самих одиницях,

що й дані. Хоча середній квадратичний відхил та середня абсолютна похибка є досить схожими, але різниця між ними полягає у тому, що MAE – це лінійна оцінка, яка означає, що всі індивідуальні відмінності однаково зважені в середньому, тоді як RMSE – є квадратичним правилом оцінки, яке вимірює середню величину помилки. Процес роботи метрики полягає у наступних кроках:

- для кожного прогнозованого значення обчислюється квадрат різниці між ним і фактичним значенням;
- усі ці квадрати різниць усереднюються;
- отримане значення підноситься до квадратного кореня.

Таким чином обчислення значення точності рекомендації буде виглядати таким чином:

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n}} \quad (2.3)$$

де n – це кількість спостережень, y_i – фактичне значення, а \hat{y}_i – прогнозоване значення.

Чим менше значення RMSE, тим краще модель або система рекомендацій. Високе RMSE вказує на те, що модель або система рекомендацій не так точно прогнозують або рекомендують значення, що може вказувати на їхню низьку ефективність.

Метрика NDCG може використовуватися на етапі створення рекомендації користувачу, оскільки її основна ідея полягає в тому, щоб давати вагомість більш релевантним елементам, розташованим ближче до верху списку рекомендацій. Чим вище релевантність і чим ближче елемент до початку списку, тим вище буде його значення NDCG. Процес обчислення складається з таких кроків:

- спочатку для кожного користувача отримується список рекомендацій з відповідними оцінками релевантності;
- потім рекомендації сортуються за оцінкою релевантності;
- далі обчислюється кумулятивний приріст (CG), який враховує оцінку релевантності кожного елемента у відсортованому списку;

– нарешті, обчислюється і нормалізується кумулятивний приріст (NDCG), щоб врахувати позицію кожного елементу в рекомендованому списку.

Деякі метрики можуть бути представлені у вигляді одного числового значення, тоді як інші можна краще візуалізувати за допомогою таблиць або графіків.

Таким чином, вирішено, що для перевірки ефективності вебсистеми з методом рекомендацій товарів за аналізом попередньої купівельної поведінки засобами машинного навчання будуть використовуватися такі метрики: нормалізований дисконтований кумулятивний приріст, середня абсолютна похибка та створюватимуться графіки або таблиці для виведення кількості здійснених покупок, зареєстрованих користувачів тощо.

2.6 Висновки до 2 розділу

У даному розділі було розроблено метод рекомендацій товарів, що ґрунтується на аналізі попередньої покупкової поведінки користувачів за допомогою методів машинного навчання. Цей метод був впроваджений у вебсистему, присвячену ігровим консолям та аксесуарам. Розроблений підхід передбачає кілька етапів обробки текстових даних, які проходять через різні моделі машинного навчання як: TF-IDF, метод косинусної подібності та стоп-слова. Після обробки дані передаються на вхід функції, яка генерує рекомендації товарів для користувачів.

Крім того, була спроектована вебсистема ігрових консолей та аксесуарів, щоб візуалізувати роботу розробленого методу. Спроектована вебсистема надалі має виконувати такі функції як: аутентифікація та авторизація користувача, перегляд товарів, додавання їх у кошик та створення замовлення, фільтрація товарів за тегами, кольорами та категоріями, виведення рекомендацій; також додавання, редагування та видалення товарів з бази даних адміністратором та виведення різних значень і таблиць ефективності роботи вебсистеми з методом.

Для коректної роботи з даними була спроектована даталогічна структура бази даних з головними 3 таблицями – замовлень, товарів та користувачів. Таблиця розрахована для роботи всіх функцій вебсистеми ігрових консолей та їх аксесуарів з методом рекомендацій. Для реалізації бази даних та вебсистеми обрано використовувати систему керування базами даних MySQL та Laravel 9 з Vue.js відповідно. Метод має бути реалізований мовою програмування Python.

Розділ 3 Програмна реалізація вебсистеми ігрових консолей та аксесуарів

Даний розділ посвячений програмній реалізації теми кваліфікаційної роботи – методу рекомендацій за аналізом попередньої купівельної поведінки клієнта засобами машинного навчання для вебсистеми ігрових консолей та аксесуарів. Після проєктування вебінтерфейсу програми та бази даних до неї, була розроблена структура методу та вебсистеми в цілому, проведено тестування функціональних можливостей вебресурсу та визначено ефективність вебсистеми з запропонованим методом.

Програмна реалізація вебсистеми ігрових консолей з запропонованим методом є важливим етапом для кваліфікаційної роботи. Метод, у даній роботі буде використаний для підвищення продажів, покращення користувацького досвіду, а також для забезпечення персоналізованого підходу до кожного клієнта. Інтеграція методу у вибрану вебсистему дозволить користувачам частіше здійснювати покупки, а також надалі виявити найбільш популярні продукти та відповідно коригувати асортимент. При розробці вебсистеми ігрових консолей та аксесуарів застосовувався архітектурний шаблон моделі MVC, який забезпечує чітке розмежування відповідальностей між компонентами системи. Системи рекомендацій для більшості інших магазинів допомагають підвищувати продажі приблизно на 10-30 відсотків, обраний метод може допомогти магазинів покращити статистику продажів та збільшити кількість зареєстрованих користувачів – це сприятиме збільшенню конверсії, повторних покупок і загальної задоволеності клієнтів.

3.1 Структура та особливості реалізації методу рекомендацій за аналізом попередньої купівельної поведінки клієнта

Реалізований модуль методу рекомендацій на основі аналізу попередньої купівельної поведінки клієнта включає декілька ключових компонентів. Серед

них: безпосередньо метод рекомендацій, його інтеграція з серверною частиною, що включає отримання результатів роботи методу та обробку вхідних даних, а також підключення до клієнтської частини за допомогою методу *getRecommendation()*.

Метод рекомендацій (рис.3.1) реалізовано з використанням мови програмування Python та бібліотек для машинного навчання, таких як *pandas*, *scikit-learn*, та інших. Для зручного видобування даних з бази даних використовується *SQLAlchemy*, це дозволяє ефективно взаємодіяти з базою даних *MySQL*. Початкові дані, що отримані з бази даних, це всі записи про замовлення та продукти. Для обробки та аналізу цієї інформації, вони конвертуються в *DataFrame* формат за допомогою бібліотеки *pandas*. Після цього, за допомогою бібліотеки *scikit-learn*, виконується векторизація текстових даних описів продуктів та побудова косинусної матриці схожості. Отримавши матрицю схожості, для певного користувача обчислюються схожі продукти на основі його попередньої купівельної історії. Ця інформація повертається у вигляді рекомендованих продуктів.

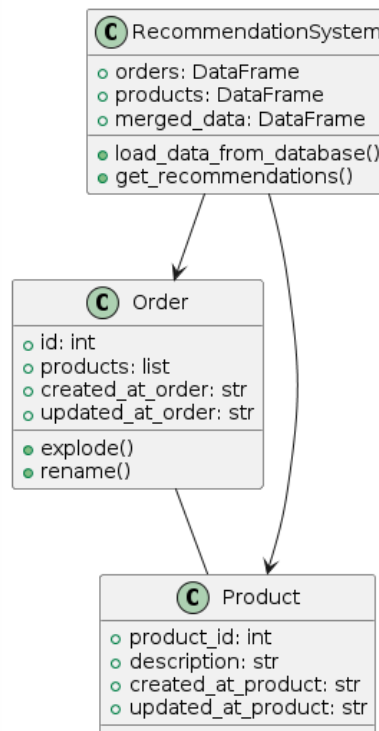


Рисунок 3.1 – Діаграма класів методу рекомендацій

Діаграма на рисунку 3.1 показує ключові класи, що використовуються в модулі рекомендаційної системи, а також їх взаємозв'язки. Вона складається з трьох головних класів: Order, Product і RecommendationSystem. Клас Order пов'язаний з класом Product через те, що одне замовлення може містити один або більше продуктів. Цей клас має такі атрибути:

- id: int – унікальний ідентифікатор замовлення;
- products: list – список продуктів у замовленні, який зберігається у вигляді JSON-стрічки;
- user_id: int – ідентифікатор користувача, що створив замовлення;
- total_price: int – загальна сума замовлення;
- status: int – статус доставки замовлення;
- address: string – адреса доставки;
- created_at_order: str – дата та час створення замовлення.
- updated_at_order: str – дата та час останнього оновлення замовлення.

Оскільки список продуктів, що є у замовленні зберігається у вигляді JSON-стрічки, необхідно було правильно розпаковувати дані та перейменувати за необхідності декотрі атрибути.

Клас Product же містить інформацію про продукти, головними атрибутами цього класу є product_id та description, хоча для аналізу використовуються й інші дані.

Клас RecommendationSystem є головним у цій діаграмі, він має залежності від класів Order і Product, оскільки використовує об'єкти цих класів для виконання своїх функцій. Він використовує такі атрибути як:

- orders: DataFrame – дані про замовлення у форматі DataFrame;
- products: DataFrame – дані про продукти у форматі DataFrame;
- merged_data: DataFrame – об'єднані дані замовлень і продуктів.

Для отримання рекомендацій на основі купівельної поведінки користувача клас RecommendationSystem використовує метод get_recommendations(). Цей метод виконує фільтрацію даних користувача, після чого отримує унікальні

ідентифікатори товарів, які користувач уже придбав. Для кожного з цих товарів метод отримує список схожих товарів, використовуючи матрицю схожостей (рис. 3.2), яка обчислює схожість між усіма товарами на основі даних товарів користувача. На цьому етапі товари також сортуються за рівнем схожості, а з отриманого списку вибираються топ-5 найкращих товарів. Такий підхід дозволяє забезпечити користувача релевантними рекомендаціями, базуючись на його попередніх покупках-вподобаннях.

```
[
  [1.          , 0.36281309, 1.          , 0.37869423, 0.20576022, 0.46010131, 0.          , 0.17391117, 0.03529503, 0.          ],
  [0.36281309, 1.          , 0.36281309, 0.42570513, 0.23130319, 0.51721805, 0.          , 0.19550042, 0.03967654, 0.          ],
  [1.          , 0.36281309, 1.          , 0.37869423, 0.20576022, 0.46010131, 0.          , 0.17391117, 0.03529503, 0.          ],
  [0.37869423, 0.42570513, 0.37869423, 1.          , 0.24142784, 0.53985784, 0.          , 0.42340285, 0.04141327, 0.          ],
  [0.20576022, 0.23130319, 0.20576022, 0.24142784, 1.          , 0.60862879, 0.          , 0.60101228, 0.30494191, 0.          ],
  [0.46010131, 0.51721805, 0.46010131, 0.53985784, 0.60862879, 1.          , 0.          , 0.2479238 , 0.05031579, 0.          ],
  [0.          , 0.          , 0.          , 0.          , 0.          , 0.          , 1.          , 0.          , 0.          , 0.05725814],
  [0.17391117, 0.19550042, 0.17391117, 0.42340285, 0.60101228, 0.2479238 , 0.          , 1.          , 0.25774081, 0.          ],
  [0.03529503, 0.03967654, 0.03529503, 0.04141327, 0.30494191, 0.05031579, 0.          , 0.25774081, 1.          , 0.          ],
  [0.          , 0.          , 0.          , 0.          , 0.          , 0.          , 0.05725814, 0.          , 0.          , 1.          ]
]
```

Рисунок 3.2 – Матриця схожості продуктів для користувача з user_id 6

Для інтеграції цього методу до серверної сторони вебсистеми був розроблений відповідний контролер у фреймворку Laravel, який має назву RecommendationController. Цей контролер відповідає за обробку запитів, пов'язаних із методом рекомендацій. Він отримує інформацію про авторизованого користувача зі сторони клієнта та передає її до Python скрипта для обробки. Після цього контролер отримує результати від Python скрипта і повертає їх на клієнтську сторону для відображення. Для виконання цих дій контролер використовує бібліотеку *Symfony* та відповідний метод *Process()*, який допомагає запускати код мови програмування Python. Це забезпечує зручний та ефективний спосіб взаємодії між серверною та клієнтською частинами вебсистеми, що дозволяє створювати перевірки на стороні сервера, чи є авторизований користувач зі сторони клієнта, та чи виконався скрипт успішно (рис.3.3). Запуск методу відбувається одразу після авторизації користувача, а також перевіряється, чи користувач раніше оформляв замовлення. Якщо обидві умови задовільні, формуються рекомендації, які відправляються на клієнтську частину вебсистеми.

Кількість рекомендацій може бути від 1 до 5, залежно від результатів аналізу попередньої купівельної поведінки користувача.

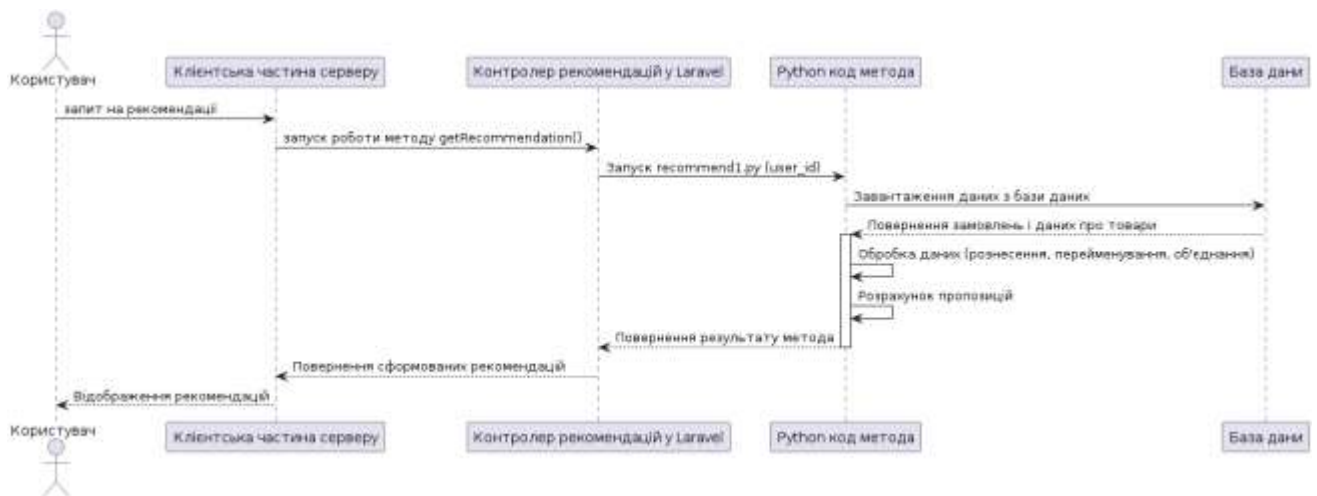


Рисунок 3.3 – Діаграма послідовностей модуля з методом

Для виведення рекомендацій на клієнтській стороні використовується Vue.js, який звертається до сервера, щоб отримати список рекомендованих продуктів для відображення користувачу. На клієнтській стороні сформовано метод `getRecommendation()`, який відповідає за отримання рекомендацій для поточного користувача через запит до серверної сторони. Метод використовує бібліотеку `axios` для відправлення GET-запиту до API за певною адресою, де передається `user_id`, який містить ідентифікатор поточного користувача. Після відбувається обробка відповіді від сервера, у разі успішного запиту відповідь сервера зберігається в змінній `recommendations`, що містить дані з рекомендаціями, у разі помилки – помилка виводиться у консоль.

Таким чином, використання бібліотек у формуванні модуля методу значно облегчило створення програмної реалізації. Зокрема, бібліотеки для роботи з даними (`pandas`), обробки тексту (`TfidfVectorizer`) та обчислення схожості, підключення до бази даних (`SQLAlchemy`) забезпечили високу ефективність та надійність процесу. Модуль сформований так, аби легко інтегруватися з серверною частиною вебсистеми та забезпечувати ефективну взаємодію з клієнтською стороною. Завдяки чітко визначеним методам та структурам даних,

модуль автоматично виконує розрахунки для визначення рекомендацій на основі купівельної поведінки користувачів одразу після авторизації користувача.

3.2 Структура та особливості реалізації вебсистеми ігрових консолей та аксесуарів

Для реалізації структури вебсистеми ігрових консолей та аксесуарів використовується фреймворк Laravel. Цей фреймворк забезпечує популярну архітектурну модель, а саме Model-View-Controller (MVC), яка підходить для розробки різноманітних застосунків, включаючи інтернет-магазини. Такий підхід дозволяє розділити додаток на три основні компоненти: Модель, Вид та Контролер, що забезпечує структурованість коду, його легку підтримку та розширення.

Моделі в архітектурі MVC відповідають за управління даними, логікою додатка та правилами бізнесу. У реалізованій системі модель включає класи для роботи з базою даних, яка зберігає інформацію про продукти, користувачів та їхні попередні покупки.

Види відповідають за відображення інформації користувачу та забезпечення взаємодії з ним. Іншими словами, це інтерфейс програми. У вебсистемі вид представляє інтерфейс, через який користувачі можуть переглядати продукти, отримувати рекомендації та здійснювати покупки. Адміністратори системи мають можливість додавати нові товари, керувати існуючими продуктами та аналізувати поведінку користувачів через цей же інтерфейс.

Контролери забезпечують інтеграцію між користувацьким інтерфейсом та логікою додатка. Вони відповідають за отримання запитів від користувачів, обробку цих запитів з використанням моделі та повернення до відповідних видів. У розробленій системі контролери обробляють запити на перегляд продуктів, генерацію рекомендацій, авторизацію та реєстрацію клієнтів. Вони також

управляють взаємодією між користувачами та системою, обробляючи запити на оформлення замовлень та інші операції.

Відповідно до розробленої структури бази даних та інформаційної системи в другому розділі було створено такі компоненти шаблону MVC:

Моделі вебсистеми:

- User – основна модель, що отримує дані про користувача, під час реєстрації в ресурсі, а також додаткових налаштуваннях профіля;

- Tag – модель, яка отримує інформацію про теги, також пов'язана з додатковою моделлю ProductTag, що з'єднує дані у базі даних продуктів та відповідних тегів до них;

- Color – модель для отримання даних про кольори товарів, пов'язана з додатковою моделлю ColorProduct;

- Category – модель з допомогою якої можна отримати дані про категорії до яких належать товари системи;

- Product – одна з основних моделей, яка забезпечує збереження та подання інформації про всі продукти системи, використовується для тестування роботи методу рекомендацій;

- Order – остання з головних моделей, що відображає всі необхідні дані про створене замовлення, також використовується для роботи методу пропозицій.

Види-інтерфейси: розроблена система має два основні типи інтерфейсів: інтерфейси адміністратора та клієнта. Для розмітки сторінок використовується мова розмітки HTML, яка поєднується з методами PHP для адміністративної системи та JavaScript для клієнтської системи.

Система адміністратора забезпечується через файли Index, Create, Edit та Show, що використовуються для виконання наступних дій з продуктами, тегами, кольорами та категоріями:

- Index – відображає список всіх елементів, наприклад, усіх продуктів, тегів, кольорів або категорій;

- Create – дозволяє адміністратору додавати нові елементи до системи;

- Edit – забезпечує можливість редагування існуючих елементів;

- Show – відображає детальну інформацію про конкретний елемент.

Підсистема роботи з замовленнями також має файли Index, Edit та Show:

- Edit – редагує статусну інформацію про замовлення.

Підсистема управління користувачами вебресурсу включає два інтерфейси:

- Перегляд всіх користувачів – відображає список всіх зареєстрованих користувачів;

- Детальний перегляд користувача – відображає детальну інформацію про конкретного користувача.

Для клієнтської системи інтерфейси забезпечують зручний і інтуїтивно зрозумілий доступ до каталогу продуктів, індивідуальних рекомендацій, кошика для покупок та процесу оформлення замовлення. Використаний фреймворк JavaScript Vue.js для реалізації інтерфейсів для клієнтів вебсистеми дозволив забезпечити високу продуктивність, динамічність і інтерактивність користувацького досвіду. Клієнтська система має такі *Views*:

- Login – інтерфейс сторінки авторизації зареєстрованого користувача у вебсистемі;

- Registration – інтерфейс сторінки з реєстрацією нових користувачів у системі;

- Index та Show – інтерфейси у папці *product*, що відповідають за відображення сторінки каталогу продуктів бази даних з блоком фільтрації товарів, а також сторінки з детальною інформацією про обраний товар;

- Index та Thank – інтерфейси у папці *cart*, які забезпечують відображення доданих у кошик товарів, можливість оформлення замовлення та сторінки вдячності за оформлене замовлення у вебсистемі;

- Index – інтерфейс у папці *main*, який відображає головну сторінку магазину з рекламою, всіма продуктами та блоком рекомендацій, який з'являється тільки після авторизації користувача у системі.

Функціонал системи забезпечується через файли з контролерами, вони працюють зі всією логікою вебсистеми. Для здійснення обробки запитів від

користувачів, взаємодію з моделями та повернення відповідних видів було створено такі два види контролерів:

1. Контролери адміністративної частини: ці контролери організовані в папки відповідно до основних функціональних модулів системи, таких як – *User*, *Category*, *Color*, *Tag*, *Product*, *Order*. Майже кожний з цих папок включає такі контролери:

- *CreateController* – обробка запитів на створення нових записів;
- *DeleteController* – відповідає за видалення записів;
- *EditController* – обробляє запити на редагування існуючих записів;
- *IndexController* – забезпечує відображення списку всіх записів;
- *ShowController* – відповідає за відображення деталей конкретного запису;
- *StoreController* та *UpdateController* – обробляє запити на збереження нових записів та оновлення існуючих відповідно.

2. Контролери клієнтської частини організовані для забезпечення функціональності, необхідної для взаємодії клієнтів з вебсистемою та включають:

- *AuthController* – контролер, що допомагає при авторизації, ідентифікації користувача, виходу з акаунту тощо;
- *StoreController* – контролер, що відповідає за збереження (реєстрацію) нових користувачів у базі даних;
- *FilterListController* – контролер, який забезпечує процес фільтрування товарів за тегами, кольорами, ціною та категоріями;
- *IndexController* – контролер, який обробляє запит на відображення каталогу товарів;
- *ShowController* – контролер, який обробляє запит на відображення даних конкретного товару;
- *StoreController* – контролер, який допомагає збереженню оформлених замовлень;

– RecommendationController – контролер для генерації та відображення персоналізованих рекомендацій.

Таким чином, було детально оглянуто реалізовану структуру вебсистеми ігрових консолей та аксесуарів. Зареєстрований та авторизований користувач має більше можливостей для створення різноманітних запитів, наприклад, оформлення замовлення, отримання персональних рекомендацій, налаштування власного профілю тощо. Перегляд, фільтрація товарів та додавання товару у кошик – ці запити може виконувати звичайний клієнт системи. Після авторизації адміністратору надається повний доступ до всього функціоналу – додавання, редагування та видалення записів, моніторинг та аналіз даних. Після створення будь-якого з цих запитів контролер відправляє їх на відповідні моделі. Моделі у свою чергу взаємодіють з базою даних та передають дані назад до контролера. Результат виводиться на відповідні інтерфейси. Структура модуля методу, що використовується у вебсистемі описана в рисунку 3.4.

Розроблена структура модуля, який реалізує метод рекомендацій у вебсистемі ігрових консолей та аксесуарів, використовує шаблон MVC, що забезпечує чітке розмежування відповідальностей між компонентами системи. Це сприяє організації коду, робить його більш структурованим, підтримуваним та легким для розширення. Контролери, які допомагають реалізувати функціональність методу рекомендацій, включають RecommendationController, а також відповідні контролери для управління продуктами, користувачами та замовленнями.

Розроблена схема модуля полегшує процес відстеження та виправлення помилок, а також внесення змін у код, розділяючи логіку роботи вебсистеми з методом на частини. Це дозволяє легко додавати нові функції та масштабувати систему відповідно до потреб бізнесу.

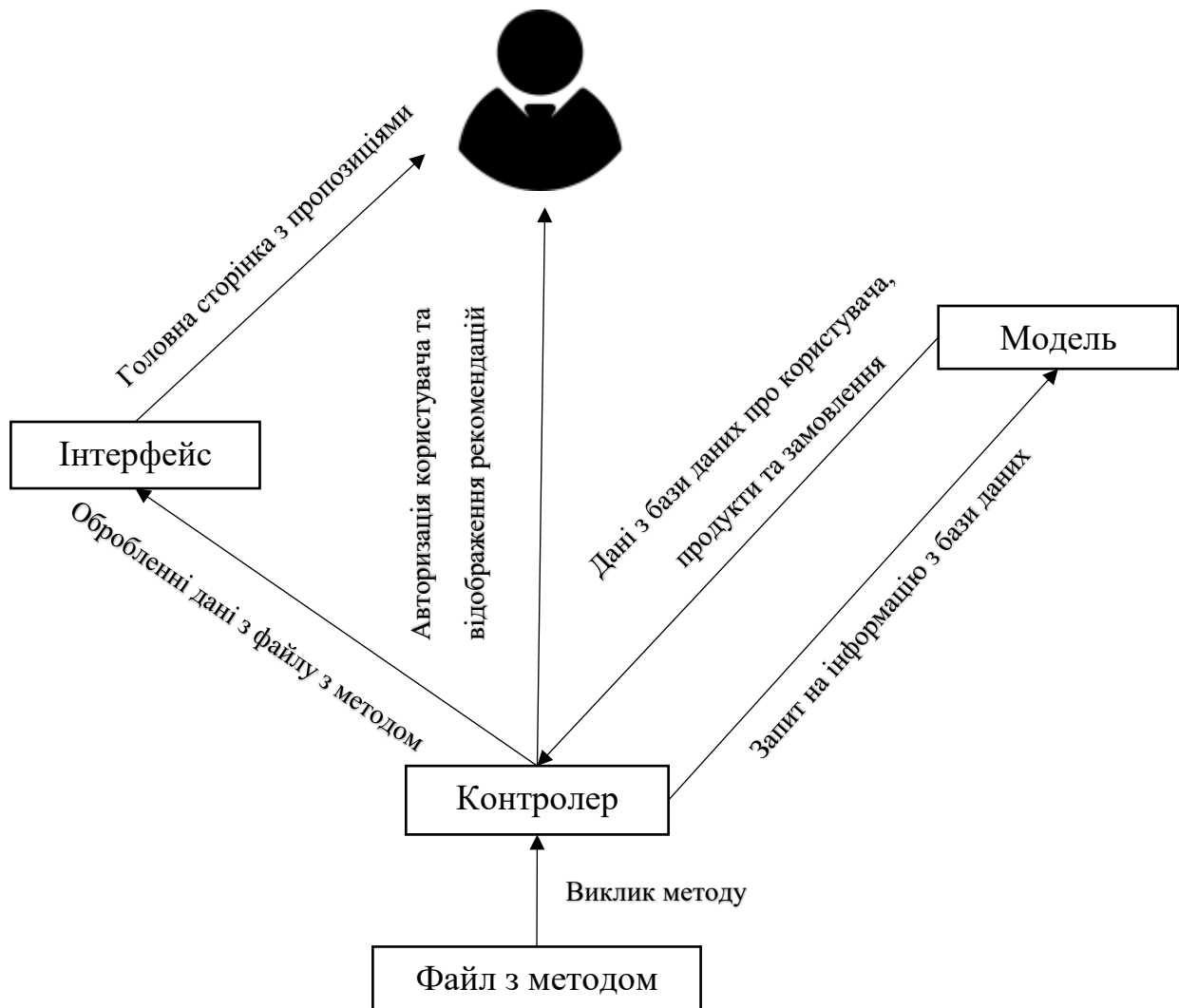


Рисунок 3.4 – Схема модуля, який реалізує метод

Таким чином, використання шаблону MVC у розробці структури вебсистеми ігрових консолей та аксесуарів, що використовує метод рекомендацій, дозволяє створити ефективну, масштабовану та підтримувану систему, яка сприяє підвищенню конкурентоспроможності та задоволення потреб споживачів.

3.3 Тестування інформаційної системи

Для перевірки коректності роботи різних функціональних можливостей вебсистеми, необхідно було провести декілька тестувань. Аби клієнт міг мати більше можливостей взаємодії з системою існують функції реєстрації та авторизації. Це необхідно аби клієнт міг отримувати надалі персоналізовані

рекомендації на основі своїх попередніх замовлень, а також для можливості оформлення покупки.

Таким чином, аби отримати доступ до цих можливостей, клієнт має зареєструватися. Для цього клієнт повинен з головної сторінки перейти на сторінку з авторизацією. Після цього необхідно обрати на формі «Створити новий акаунт GASY», через що буде переадресація на форму реєстрації. Дані клієнт повинен заповнити необхідні поля та натиснути на кнопку «Registration» (рис.3.5).

Рисунок 3.5 – Реєстрація клієнта на вебсистемі

Для перевірки роботи цієї функції (реєстрації) здійснено тест у вигляді тест-кейсу (Таблиця 3.1).

Таблиця 3.1 – Тест-кейс TC0001

Тест-кейс ID: TC0001	Пріоритет: 1	Створено: 07.05.2024, Вовк С. В.
Назва: Перевірка процесу реєстрації клієнта.		
Вхідні дані: Запущено сервер та клієнтська частина, ім'я – «Віталій», пошта – «vitaliyV@gmail.com», пароль – «19721411».		
Кроки		Очікуваний результат

<ol style="list-style-type: none"> 1. Запустити веб-браузер. 2. Перейти за посиланням «http://gasy.loc/». 3. Розгорнути головну сторінку та натиснути на кнопку зі символом людини. 4. Ввести необхідні дані у комірки. 5. Натиснути на кнопку «Registration». 	<p>У полі для гіперпосилань прописати «http://gasy.loc/». Перехід на головну сторінку. Вибрати кнопку зі символом людини. Відкривається сторінка з реєстрацією. Введені дані передаються та зберігаються у базі даних таблиці users.</p>
<p>Результат виконання тест-кейсу: пройдено успішно</p>	

При коректній роботі функції реєстрації новий користувач має додатися до бази даних (рис.3.6), а у вебсистемі відбувається переадресація на головну сторінку.

id	name	email	email_verified_at	password	remember_token	created_at	updated_at
6	Галина	test1@gmail.com	NULL	\$2y\$10\$S0AxCNn3VnxtmOxx99GxSuMd.sq5WIBbavn0i9tiAfH...	NULL	2024-05-08 12:51:19	2024-05-08 12:51:19
9	user1	user1@gmail.com	NULL	\$2y\$10\$30SleOtrW9MpCKCEr/lwYelfwHJJH5K5h52FgAuwXIY...	NULL	2024-05-09 12:54:45	2024-05-09 12:54:45
10	Олена	vovkstefa@khmnu.edu.ua	NULL	\$2y\$10\$0DmAj0sqVbUcT6mLigXmUeGkY.BqhVCnwSjPTvX3Y.4...	NULL	2024-05-17 09:30:52	2024-05-17 09:30:52
11	Світослап	lolal228@gmail.com	NULL	\$2y\$10\$V73dlbX.Ira8v98SZNEyGO1gfcTWuJ6UIY22WMksNJK...	NULL	2024-05-17 09:36:25	2024-05-17 09:36:25
12	Вікторія	minvivi@gmail.com	NULL	\$2y\$10\$njmVE7keliXg7F.4TK/4uM8wemGQc/rd3qBrJse9oD...	NULL	2024-05-17 09:45:19	2024-05-17 09:45:19
13	Віталій	vitaliyV@gmail.com	NULL	\$2y\$10\$0XG5GeM5SSZn3F74oNBR2er3I4z08ES17/R7rVa81.y...	NULL	2024-05-20 10:11:23	2024-05-20 10:11:23

Рисунок 3.6 – Додавання нового користувача до бази даних

Після того як клієнт зареєструвався у магазині, йому відкриваються додаткові можливості, а саме оформлення та створення замовлення. Новий користувач має можливість обрати необхідний товар просто переглядаючи каталог товарів на сторінці *Shop* або фільтрувати дані за певними ознаками. Для того, щоб знайти певний продукт у каталозі, наприклад гру для ігрової консолі за жанром жахи, раніше створений користувач повинен перейти до каталогу, обрати категорію «Ігри для консолі» та у тегах натиснути на «Horror», після натиснути на

кнопку «Filter». Вебсистема повинна обробити запит та видати на сторони клієнта всі ігри за обраним жанром. Перевірку описаного вище було проведено за допомогою тест-кейса (Таблиця 3.2).

Таблиця 3.2 – Тест-кейс TC0002

Тест-кейс ID: TC0002	Пріоритет: 2	Створено: 07.05.2024, Вовк С. В.	
Назва: Перевірка процесу фільтрації товарів.			
Вхідні дані: Запущено сервер та клієнтська частина, користувач обрав товари за категорією «Ігри для консолі» з тегом «Horror».			
Кроки		Очікуваний результат	
1. Перейти за посиланням « http://gasy.loc/ ».	2. Розгорнути головну сторінку та натиснути на кнопку «Shop».	3. Натиснути на «Ігри для консолі» та обрати тег «Horror».	4. Натиснути на кнопку «Filter».
1. Перейти за гіперпосиланням « http://gasy.loc/ ».		Перехід на головну сторінку. Перехід на сторінку з каталогом товарів. Відкривається сторінка з товарами та фільтрацією товарів. Обранні категорія та теги. Відправляється запит до серверної частини, запит обробляється та відправляє результат. Виведені ігри за жанром жахи.	
Результат виконання тест-кейсу: пройдено успішно			

Після успішної фільтрації товарів на сторінці каталогу повинен висвітлюватися товар «Dead Space», лише він має тег жахи у категорії «Ігри для консолі» (рис.3.7). Також варто зазначити, що гра «Dead Space» у базі даних має також тег Nintendo, оскільки вона у магазині передбачена винятково для ігрових консолей Nintendo.

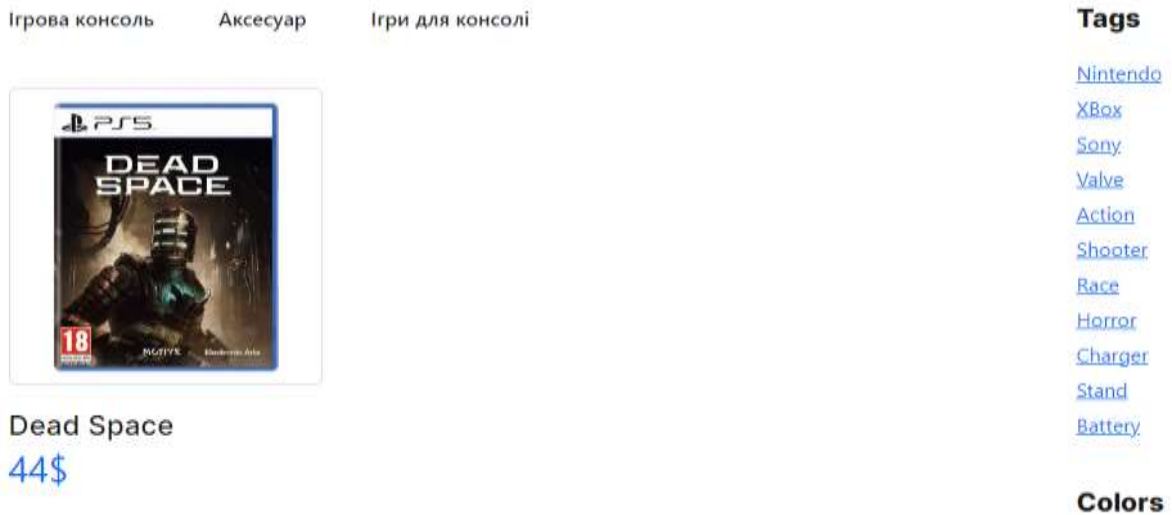


Рисунок 3.7 – Результат фільтрації товарів за жанром жахи та категорією «Ігри для консолі»

Оскільки користувач знайшов необхідний йому товар, він може додати його до кошику та оформити замовлення. Для цього необхідно навести на товар мишкою та натиснути на кнопку, що з'явиться «Add to cart», товар автоматично додається до кошика у кількості однієї одиниці. Переглянути кошик користувач може обравши символ кошика після чого відбудеться перехід на сторінку кошика. На цій сторінці користувач може одразу оформити замовлення або продовжити свої покупки. Для створення нової покупки все що необхідно заповнити користувачу адресу доставки, всі інші дані вже сформовані (рис.3.8). Перевірку коректності роботи формування та додавання замовлення до бази даних було проведено у тест-кейсі (Таблиця 3.3).

Таблиця 3.3 – Тест-кейс TC0003

Тест-кейс ID: TC0003	Пріоритет: 1	Створено: 09.05.2024, Вовк С. В.
Назва: Перевірка процесу оформлення замовлення.		
Вхідні дані: Запущено сервер та клієнтська частина, користувач обрав товар «Dead Space».		
Кроки	Очікуваний результат	

<ol style="list-style-type: none"> 1. Перейти за посиланням «http://gacy.loc/». 2. Розгорнути головну сторінку та натиснути на кнопку «Shop». 3. Обрати товар «Dead Space» та натиснути на кнопку «Add to cart». 4. Натиснути на символ корзини. 5. Перевірити дані та натиснути на «Оформити замовлення». 6. Заповнити поле з адресою доставки. 7. Натиснути на кнопку «Замовити товари». 	<p>Перехід за гіперпосиланням «http://gacy.loc/».</p> <p>Перехід на головну сторінку. Перехід на сторінку з каталогом товарів. Відкривається сторінка з товарами та фільтрацією товарів.</p> <p>Обраний товар «Dead Space». Додається товар до картки кошика. Заповнене поле адреси доставки та посилається запит через кнопку «Замовити товари». Запит обробляється та додається новий запис до бази даних.</p>
<p>Результат виконання тест-кейсу: пройдено успішно</p>	

Після успішного утворення замовлення вебсистема перенаправляє користувача на сторінку подяки за покупку на цьому вебресурсу з номером служби у разі будь-яких проблем на різних стадіях зі замовленням. Новий же запис замовлення формується у базі даних (рис.3.9). Дані, що зберігаються у базі включатимуть у собі: номер замовлення; номер користувача, що створив замовлення, JSON стрічку з всією інформацією про товар, повну суму замовлення, адресу доставки, статус доставки, а також день та час створення запису.

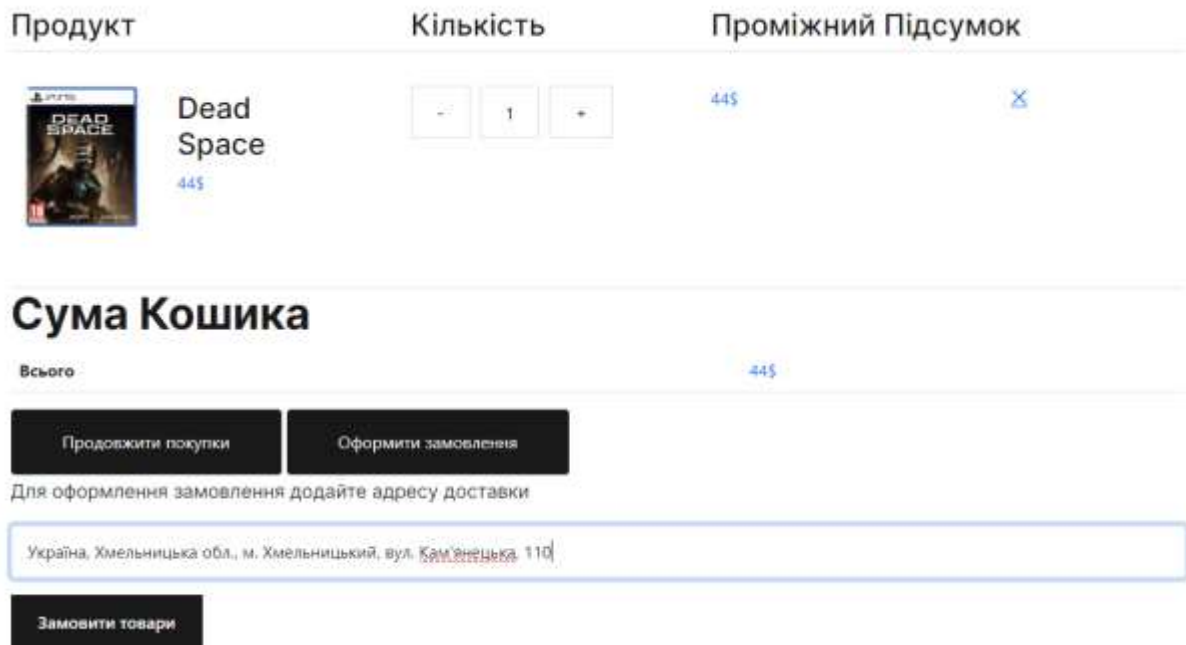


Рисунок 3.8 – Формування замовлення користувачем

id	user id	products	total price	status	address	created at	updated at
9	6	[{"id": 44, "qty": 1, "count": 50, "price": 10, "t...	46.00	0	Україна, обл. Вінницька, м. Жмеринка, вул. Соборна...	2024-05-16 20:11:55	2024-05-16 20:11:55
10	10	[{"id": 44, "qty": 2, "count": 50, "price": 10, "t...	130.00	0	Україна, Вінницька обл., м. Жмеринка, вул. Соборна...	2024-05-17 09:31:56	2024-05-17 09:31:56
11	9	[{"id": 30, "qty": 1, "count": 27, "price": 55, "t...	167.00	0	Україна, Хмельницька обл., м. Хмельницький, вул. К...	2024-05-17 09:35:08	2024-05-17 09:35:08
12	11	[{"id": 23, "qty": 1, "count": 5, "price": 70, "ti...	150.00	0	Україна, Вінницька обл., м. Жмеринка, вул. Соборна...	2024-05-17 09:37:14	2024-05-17 09:37:14
16	12	[{"id": 5, "qty": 1, "count": 4, "price": 662, "ti...	662.00	0	Україна, Хмельницька обл., м. Хмельницький, вул. К...	2024-05-17 11:07:42	2024-05-17 11:07:42
17	13	[{"id": 22, "qty": 1, "count": 3, "price": 44, "ti...	44.00	0	Україна, Хмельницька обл., м. Хмельницький, вул. К...	2024-05-20 11:05:58	2024-05-20 11:05:58

Рисунок 3.9 – Сформований новий запис у таблиці замовлень

Останнім тестом інформаційної системи є перевірка виводу рекомендацій товарів користувачу після успішного оформлення замовлення. Оскільки метод працює автоматично, то пропозиції товарів відобразяться на головній сторінці одразу як тільки користувач авторизується та створить будь-яку покупку. Для перевірки роботи методу, який інтегровано до вебсистеми створено тест-кейс (Таблиця 3.4). Метод отримує за бази даних всі необхідні дані для роботи та по завершенню створює JSON стрічку з рекомендаціями, клієнт може бачити результат роботи методу (рис.3.10).

Таблиця 3.4 – Тест-кейс TC0004

Тест-кейс ID: TC0004	Пріоритет: 1	Створено: 13.05.2024, Вовк С. В.
----------------------	--------------	----------------------------------

Назва: Перевірка процесу формування рекомендацій.	
Вхідні дані: Запущено сервер та клієнтська частина.	
Кроки	Очікуваний результат
<p><i>Передумова:</i> користувач авторизувався та має хоча б одне створене замовлення.</p> <p>Перейти за посиланням «http://gasy.loc/».</p> <p>Розгорнути головну сторінку.</p> <p>Прогорнути нижче реклами, перегляд рекомендацій.</p>	<p>Перехід за гіперпосиланням «http://gasy.loc/».</p> <p>Перехід на головну сторінку. Нижче розташовані рекомендації користувача «Віталій» після оформленого замовлення.</p> <p>Виведено від 1 до 5 рекомендованих товарів.</p>
Результат виконання тест-кейсу: пройдено успішно	

Вам Може Сподобатися



Рисунок 3.10 – Результат роботи методу персональних рекомендацій

Таким чином, було проведене тестування деяких функцій клієнтської частини вебсистеми, де було перевірено на коректність роботу реєстрації та автоматичної авторизації нового клієнта; фільтрування товарів за тегами, категоріями, кольорами та ціною; формування нового замовлення до бази даних та виведення рекомендацій користувачу. При проведенні всіх тестів відхилень не

виявлено. Клієнт може зареєструватися без виникнення помилок, інші функції працюють вірно.

3.4 Вимоги до публікації вебсистеми

Для успішної публікації вебсистеми ігрових консолей та аксесуарів, розробленої на фреймворку Laravel 9 з використанням Vue.js для клієнтської частини, необхідно забезпечити певні вимоги до сервера та клієнта:

- процесор: сучасний багатоядерний процесор (мінімум 2 ядра, рекомендовано 4 і більше);
- оперативна пам'ять: мінімум 4 ГБ оперативної пам'яті (рекомендовано 8 ГБ і більше для кращої продуктивності);
- дисковий простір: мінімум 20 ГБ вільного місця для зберігання коду, бази даних та логів (рекомендовано SSD);
- операційна система: Linux (Ubuntu 20.04 LTS або пізніша версія) або Windows щонайменше 7 версія;
- веб-сервер: Apache (версія 2.4 або пізніша) або Nginx (версія 1.18.0 або пізніша);
- PHP: версія 8.0 або пізніша;
- composer: остання версія для управління залежностями PHP;
- база даних: MySQL – версія 8;
- Node.js: версія 14.x або пізніша;
- npm (Node Package Manager): остання версія;
- браузері: підтримка сучасних браузерів таких як Google Chrome, Microsoft Edge тощо.
- надійне інтернет-з'єднання для доступу до вебсистеми.

Дотримання вищезазначених вимог забезпечить стабільну роботу вебсистеми ігрових консолей та аксесуарів, її швидку реакцію на запити користувачів та високу безпеку даних.

3.5 Визначення ефективності вебсистеми з запропонованим методом

Для визначення, ефективності запропонованого методу у сполученні з вебсистемою проведено кілька тестів, а також виведено декілька графіків з видимими змінами, після впровадження блоку рекомендацій з 13.05 (рис.3.11). Серед проведених тестів було використано : А/В-тест (опитування), обчислено за метрикою NDCG та середньою абсолютною похибкою. При опитуванні використано експертну оцінку чат-бота ChatGPT, а також декілька людей, що зареєструвалися у вебсистемі. Для отримання оцінки чат-бота було написано запит аби серед усіх товарів було рекомендовано 5 найкращих, враховуючи товари, які придбав користувач «Віталій».

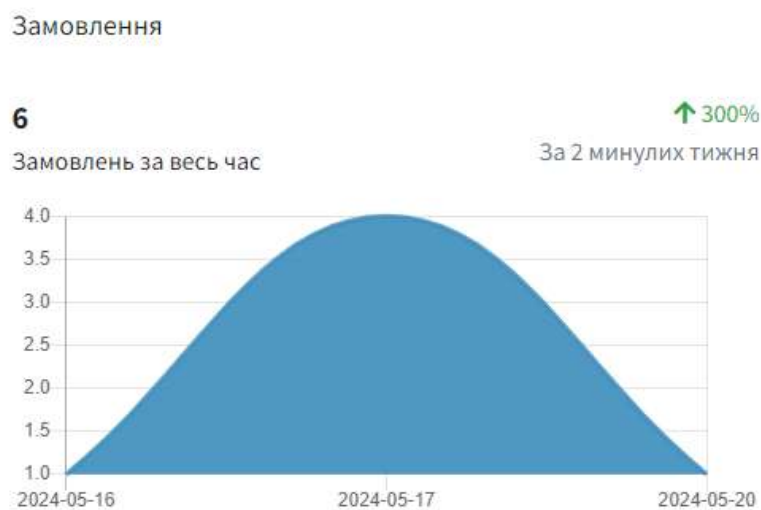


Рисунок 3.11 – Кількість здійснених покупок після додавання блоку рекомендацій (з 13.05)

Товари, які були рекомендовані користувачеві «Віталій» методом, це гра God Of War Ragnarok та три товари компанії Nintendo: контролер, зарядний пристрій та захисне скло. При аналізі наявних товарів, чат обрав дві гри: God of War Ragnarok та The Legend of Zelda: Breath of the Wild, а також зарядну станцію Nintendo, Nintendo Switch Pro контролер та Nintendo Switch OLED Blue Neon. З

проведеного дослідження з чатом метод визначив досить непогано релевантні рекомендації (рис.3.12).

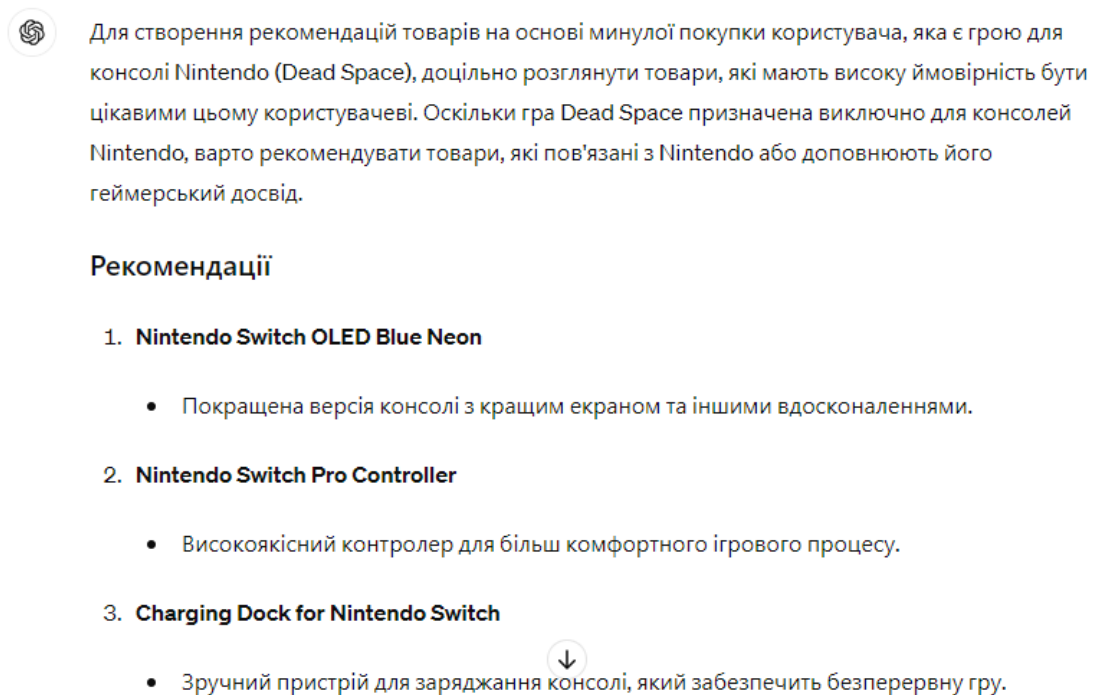


Рисунок 3.12 – Сформовані рекомендації до товару «Dead Space» чат-ботом ChatGPT

Окрім тестування роботи метода через чат-бот, також було проведено А/В-опитування декількох людей, щодо релевантності сформованих результатів методом. Після проведення опитування було отримано наступні результати:

- загальний рівень задоволеності – 67%: опитані зазначили, що рекомендовані товари були релевантними їхнім інтересам та потребам;
- при таких рекомендаціях принаймні 54% респондентів придбали б принаймні один товар із запропонованих рекомендацій;
- час на ухвалення рішення значно зменшився, але разом з цим з'явилися нові запити (40%) – створити блок рекомендацій, який би не використовував аналіз попередніх покупок, а щось інше.

За обчисленими двома метриками метод показав ефективність на 69%. Результати NDCG показали рівень релевантності вище середнього запропонованих рекомендацій, з середнім значенням $NDCG = 0.65$. Це свідчить

про те, що рекомендації були більш менш точними, але при ранжуванні інколи деякі пропозиції ставали вище, ніж ті, що були більш релевантними. Тоді як середня абсолютна похибка для методу склала 0.23, що вказує на невеликі відхилення між передбаченими та фактичними значеннями. Це продемонструвало непогану точність рекомендаційного методу.

Надалі доцільно покращувати метод для досягнення ще більш точних результатів. Це можна завдяки використанню складніших моделей машинного навчання або глибокого навчання, які можуть враховувати більше факторів, що впливають на рекомендації, також непогано розширити дані для навчання та тестування модуля рекомендацій.

Таким чином, проведення дослідження визначення ефективності вебсистеми з запропонованим методом рекомендацій дозволило отримати корисний аналіз щодо впливу рекомендацій на користувачів та загальну продуктивність. За результатами A/B-опитування та обчисленими метриками (NDCG та середньою абсолютною похибкою), метод показав непогану релевантність і точність. Після впровадження методу за графіком можна спостерігати приріст у формуванні замовлень на 300%.

3.6 Висновки до розділу 3

Для розробки методу рекомендацій за аналізом попередньої купівельної поведінки клієнта засобами машинного навчання для вебсистеми ігрових консолей та аксесуарів було обрано використовувати архітектурний шаблон MVC. Цей підхід дозволив розділити логіку роботи вебресурсу на три основні взаємопов'язані компоненти: моделі, представлення (інтерфейси) та контролери. Вибір саме цього шаблону обумовлений його здатністю легко масштабуватися відповідно до потреб, а також зручністю в підтримці та розширенні системи.

При розробці системи було використано такі фреймворки та бібліотеки:

- Laravel 9: основний PHP-фреймворк для розробки серверної частини. Використання Laravel забезпечило ефективну інтеграцію різних компонентів системи, зокрема завдяки бібліотеці *Symfony*, що інтегрувала код файлу з методом;
- Vue.js: використовувався для розробки клієнтської частини вебсистеми. Vue.js забезпечив динамічний і реактивний інтерфейс користувача, легко поєднався з серверною частиною системи, використовує бібліотеку *Axios*;
- Python – мова програмування, яка використовувалася для реалізації алгоритмів машинного навчання. Використовує такі бібліотеки як *Pandas*, *JSON*, *NumPy*, *Scikit-learn*.

Реалізована вебсистема включає такі функціональні можливості: реєстрація та авторизація користувача, оформлення замовлення, вивід каталогу та фільтрація товарів за категоріями, тегами, кольорами та ціною, додавання, редагування та видалення адміністратором записів з бази даних, формування персональних рекомендацій для користувача. Ці процеси перевірено при тестуванні та відхилень у роботі не виявлено. Також було проведено дослідження ефективності вебсистеми з запропонованим методом. Після різних тестувань було визначено, що рекомендації у вебсистемі продемонстрували непогану ефективність у наданні релевантних рекомендацій для користувачів. При тестуванні, сформовані рекомендації методом і чатом, як експертом, збігалися на 60%.

Отже, результатом розробки програмної реалізації став метод рекомендацій за аналізом попередньої купівельної поведінки клієнта засобами машинного навчання, що використовує вебсистему ігрових консолей та аксесуарів для отримання персоналізованих рекомендацій користувачем, а також допомагає користувачам знаходити продукти, що відповідають їхнім уподобанням. Розроблена система відхилень немає, у разі виникнення помилки відбувається валідація даних та запитів та оповіщення помилки.

Загальні висновки

Кваліфікаційна робота бакалавра присвячена розробці методу рекомендацій за аналізом попередньої купівельної поведінки клієнта засобами машинного навчання для вебсистеми ігрових консолей та аксесуарів, для збільшення продажів вебсистеми, покращення користувацького досвіду та допомоги користувачеві при пошуку товару за вподобанням.

Під час виконання кваліфікаційної роботи бакалавра було проведено огляд способів формування персональних рекомендацій, а також наявних засобів реалізації рекомендацій наступних покупок, поставлена мета та завдання КРБ. Для створення рекомендацій за аналізом попередньої купівельної поведінки клієнта було обрано використовувати фільтрацію на основі вмісту, що працює безпосередньо з даними тільки клієнта, без використання та порівняння даних інших користувачів системи. Для розробки методу рекомендацій було використано мову програмування Python з бібліотекою *Scikit-learn*, яка забезпечувала роботу методів обробки тексту та машинного навчання TF-IDF та косинуса подібності. Спроектовану структуру вебінтерфейсу інформаційної системи та бази даних реалізовано у програмну реалізацію – інтернет-магазин ігрових консолей та аксесуарів з блоком рекомендацій для певної категорії користувачів. Вебресурс реалізований на фреймворках Laravel (для серверної частини програми) та Vue.js (для клієнтської частини), з використанням бібліотеки *Symfony*, що інтегрує метод до решти системи.

У результаті розроблена програмна реалізація кваліфікаційної роботи забезпечує виконання таких функцій:

- реєстрація та авторизація;
- додавання, редагування та видалення адміністратором записів у базі даних;
- додавання товарів у кошик та формування нового замовлення;
- робота користувача з персональними даними;
- перегляд каталогу з товарами системи;

- фільтрація товарів за тегами, категоріями, кольорами та ціною;
- обробка даних та формування рекомендацій користувачу;
- візуалізація персональних рекомендацій.

Створена інформаційна система має потенціал для подальшого розширення функціональних можливостей. Для покращення її роботи можна покращити інтерфейс користувача, додати інші блоки рекомендацій на основі порівняння вподобань користувача з іншими клієнтами системи, впровадити нові можливості для маркетингових кампаній, такі як автоматизовані email-розсилки, впровадити розширені засоби аналітики для відстеження поведінки користувачів, що допоможе в оптимізації асортименту та маркетингових стратегій, а також додавати додаткові функції, такі як підтримка різних платіжних систем, системи управління запасами тощо.

Проаналізувавши створену програмну реалізацію можна зробити висновок, що поставленого завдання кваліфікаційна робота бакалавра досягнуто. Вимоги для розробки вебсистеми виконано та тестування інформаційної системи показало, що показує, що метод із вебсистемою працює справно та виконує поставлену задачу.

Перелік посилань

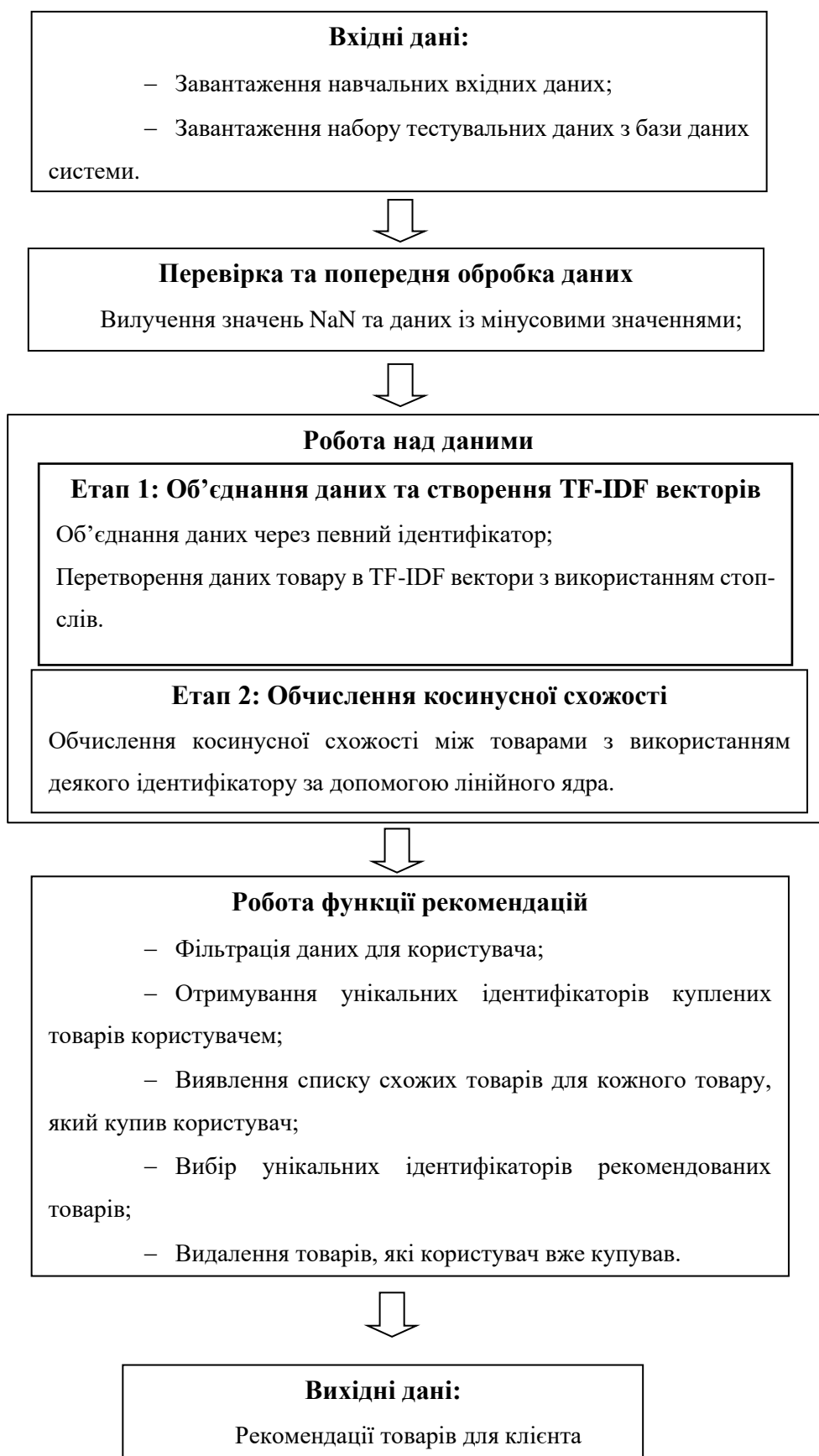
1. Sneha Khatwani, M.B. Chandak. Building Personalized and Non Personalized recommendation systems. 2016 URL: <https://ieeexplore.ieee.org/document/7877661>
2. Nikolett Lorincz. 18 Product Recommendation Examples to Increase Sales in 2024. 2024. URL: <https://www.optimonk.com/product-recommendation/>
3. Using Ecommerce Recommendation Engines to Keep Your Customers Coming Back. bigcommerce.com. URL: <https://www.bigcommerce.com/articles/ecommerce/recommendation-engine/>
4. Maruti Techlabs. What are Product Recommendation Engines? And the various versions of them? 2017. URL: <https://towardsdatascience.com/what-are-product-recommendation-engines-and-the-various-versions-of-them>
5. Cory Maklin. Memory Based Collaborative Filtering — User Based. 2022. URL: <https://medium.com/@corymaklin/memory-based-collaborative-filtering-user-based>
6. Emre Çetindemir. Personalised Product Recommendations 101: Must-Have for eCommerce Stores. 2021. URL: <https://segmentify.com/blog/personalised-product-recommendations-101-must-have-for-ecommerce-stores/>
7. Anirudha Simha. Understanding TF-IDF for Machine Learning. 2021. URL: <https://www.capitalone.com/tech/machine-learning/understanding-tf-idf/>
8. Dhilip Subramanian. Content-Based Recommendation System using Word Embeddings. 2024. URL: <https://www.kdnuggets.com/2020/08/content-based-recommendation-system-word-embeddings.html>
9. Naomy Duarte Gomes. The cosine similarity and its use in recommendation systems. 2023. URL: <https://naomy-gomes.medium.com/the-cosine-similarity-and-its-use-in-recommendation-systems>
10. K means Clustering – Introduction. www.geeksforgeeks.org. 2024. URL: <https://www.geeksforgeeks.org/k-means-clustering-introduction/>
11. Risto Hinno. Simple SVD algorithms. 2021. URL: <https://towardsdatascience.com/simple-svd-algorithms>

12. Why is deep learning used in recommender systems? www.geeksforgeeks.org. 2024.
URL: <https://www.geeksforgeeks.org/why-is-deep-learning-used-in-recommender-systems/>
13. Raditya Nurfadillah. Get To Know with Surprise. 2022. URL:
<https://medium.com/tiket-com/get-to-know-with-surprise>
14. surpriselib.com. URL: <https://surpriselib.com/>
15. making.lyst.com. URL: <https://making.lyst.com/lightfm/docs/home.html>
16. pypi.org. URL: <https://pypi.org/project/implicit/>
17. www.tensorflow.org. URL: <https://www.tensorflow.org/recommenders>
18. platform.openai.com. URL: <https://platform.openai.com/docs/guides/embeddings>
19. mahout.apache.org. URL: <https://mahout.apache.org/documentation/users/>
20. scikit-learn.org. URL: <https://scikit-learn.org/stable/>
21. www.amazon.com. URL: https://www.amazon.com/ref=nav_bb_logo
22. best.aliexpress.com. URL: <https://best.aliexpress.com/>
23. rozetka.com.ua. URL: <https://rozetka.com.ua/ua/>
24. www.foxtrot.com.ua. URL: <https://www.foxtrot.com.ua/>
25. Boris Mustapic. An Introduction to TF-IDF: What It Is & How to Use It. 2024. URL:
<https://www.semrush.com/blog/tf-idf/>
26. Arjun Prakash. Understanding Cosine Similarity: A key concept in data science. 2023. URL:
<https://medium.com/@arjunprakash027/understanding-cosine-similarity-a-key-concept-in-data-science>
27. chat.openai.com. URL: <https://chat.openai.com/>
28. Lalitesh. Product Recommendation Systems for Retail. 2024. URL:
<https://www.polestarllp.com/blog/product-recommendation-systems-for-retail>
29. Wikipedia. Mean_absolute_error. URL:
https://en.wikipedia.org/wiki/Mean_absolute_error
30. resources.eumetrain.org. Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). URL:
https://resources.eumetrain.org/data/4/451/english/msg/ver_cont_var/uos3/uos3_ko1.htm

ДОДАТКИ

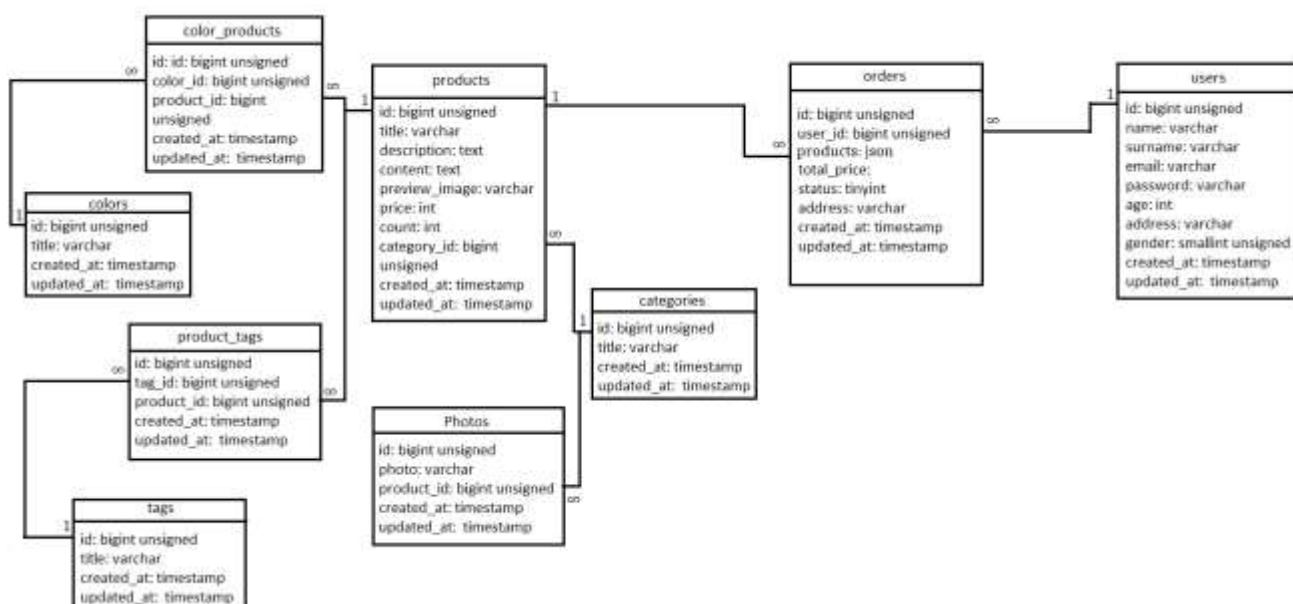
Додаток А

Схема методу рекомендацій товарів за аналізом попередньої купівельної поведінки клієнта



Додаток Б

Структура бази даних вебсистеми ігрових консолей та аксесуарів



Додаток В

Основні програмні коди

Лістинг Order/Storecontroller.php

```
<?php

namespace App\Http\Controllers\API\Order;
use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use App\Models\Order;
use App\Http\Resources\Order\OrderResource;
use App\Http\Requests\API\Order/StoreRequest;
use Illuminate\Support\Facades\Auth;
// use App\Models\User;

class StoreController extends Controller
{
    public function __invoke(StoreRequest $request){
        $data = $request->validated();

        $userId = Auth::id();
        $order = Order::create([
            'products' => json_encode($data['products']),
            'user_id' => $userId,
            'total_price' => $data['total_price'],
            'address' => $data['address'],
            'status' => 0,
        ]);

        return new OrderResource($order);
    }
}
```

Лістинг RecommendationController.php

```
<?php

namespace App\Http\Controllers\API\Recommendation;

use Symfony\Component\Process\Exception\ProcessFailedException;
use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use Symfony\Component\Process\Process;
use Illuminate\Support\Facades\Auth;

class RecommendationController extends Controller
{
    public function getRecommendations(Request $request){

        $user_id = Auth::id();

        $scriptPath = storage_path('app/python_scripts/recommend1.py');

        $process = new Process(['python', $scriptPath, $user_id]);
        $process->run();

        if (!$process->isSuccessful()) {
            throw new ProcessFailedException($process);
        }

        $output = $process->getOutput();
        $recommendations = json_decode($output, true);

        return response()->json($recommendations);
    }
}
```

```

    }
}

```

Лістинг Product.php

```
<?php
```

```

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use App\Models\Traits\Filterable;
use App\Models\ProductImage;

class Product extends Model
{
    use Filterable;

    protected $table = 'products';
    protected $guarded = false;
    protected $fillable = ['title', 'description', 'content', 'preview_image', 'price', 'count', 'category_id'];

    public function category()
    {
        return $this->belongsTo(Category::class, 'category_id', 'id');
    }

    public function tags()
    {
        return $this->belongsToMany(Tag::class, 'product_tags', 'product_id', 'tag_id');
    }

    public function colors()
    {
        return $this->belongsToMany(Color::class, 'color_products', 'product_id', 'color_id');
    }

    public function getImageUrlAttribute()
    {
        return url('storage/' . $this->preview_image);
    }

    public function productImages()
    {
        return $this->hasMany(ProductImage::class, 'product_id', 'id');
    }
}

```

Лістинг Order.php

```
<?php
```

```

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Order extends Model
{
    use HasFactory;

    protected $table = 'orders';
    protected $guarded = false;

    const STATUS_PROCESSED = 0;
    const STATUS_DELIVERED = 1;
}

```

```

static function getOrderStatusString()
{
    return [
        self::STATUS_DELIVERED => 'Доставлено',
        self::STATUS_PROCESSED => 'Оформлено',
    ];
}

protected $fillable = ['user_id', 'total_price', 'status', 'address', 'products'];

public function getOrderStatusStringAttribute()
{
    return self::getOrderStatusString()[$this->status];
}

public function user()
{
    return $this->belongsTo(User::class, 'user_id', 'id');
}
}

```

ЛІСТИНГ ProductFilter.php

```
<?php
```

```

namespace App\Http\Filters;

use App\Http\Filters\AbstractFilter;
use Illuminate\Database\Eloquent\Builder;

class ProductFilter extends AbstractFilter{

    const COLORS = 'colors';
    const TAGS = 'tags';
    const CATEGORIES = 'categories';
    const PRICES = 'prices';

    protected function getCallbacks(): array
    {
        return [
            self::COLORS => [$this, 'colors'],
            self::TAGS => [$this, 'tags'],
            self::CATEGORIES => [$this, 'categories'],
            self::PRICES => [$this, 'prices'],
        ];
    }

    protected function colors(Builder $builder, $value){
        $builder->whereHas('colors', function($build) use ($value){
            $build->whereIn('color_id', $value);
        });
    }

    protected function tags(Builder $builder, $value){
        $builder->whereHas('tags', function($build) use ($value){
            $build->whereIn('tag_id', $value);
        });
    }

    protected function categories(Builder $builder, $value){
        $builder->whereIn('category_id', $value);
    }

    protected function prices(Builder $builder, $value){
        $builder->whereBetween('price', $value);
    }
}

```

}

ЛІСТИНГ admin/category/create.blade.php

@extends('layouts.main')

@section('content')

<!-- Content Header (Page header) -->

<div class="content-header">

<div class="container-fluid">

<div class="row mb-2">

<div class="col-sm-6">

<h1 class="m-0">Додати категорію</h1>

</div><!-- /.col -->

<div class="col-sm-6">

<ol class="breadcrumb float-sm-right">

<li class="breadcrumb-item active">Головна

</div><!-- /.col -->

</div><!-- /.row -->

</div><!-- /.container-fluid -->

</div>

<!-- /.content-header -->

<!-- Main content -->

<section class="content">

<div class="container-fluid">

<!-- Small boxes (Stat box) -->

<div class="row">

<form action="{ { route('admin.category.store') } }" method="post">

@csrf

<div class="form-group">

<input type="text" name="title" class="form-control" placeholder="Введіть назву категорії">

</div>

<div class="form-group">

<input type="submit" class="btn btn-primary" value="Додати категорію">

</div>

</form>

</div>

<!-- /.row -->

</div><!-- /.container-fluid -->

</section>

<!-- /.content -->

@endsection

ЛІСТИНГ Index.vue

<template>

<div>

<section class="site-banner jarallax min-height300 padding-large"

style="background: url(src/assets/images/hero-image.jpg) no-repeat; background-position: top;">

<div class="container">

<div class="row">

<div class="col-md-12">

<h1 class="page-title">Shop page</h1>

<div class="breadcrumbs">

<router-link to="/">Home </router-link>

Shop

</div>

</div>

</div>

</div>

</section>

```

<div class="shopify-grid padding-large">
  <div class="container">
    <div class="row">
      <section id="selling-products" class="col-md-9 product-store">
        <div class="container">
          <ul class="tabs list-unstyled">
            <li v-for="category in filterList.categories" :id="category.id" :for="category.id"
              class="tab" @click.prevent="addCategory(category.id)">{{ category.title }}
            </li>
          </ul>
          <div class="tab-content">
            <div id="all" data-tab-content class="active">
              <div class="row d-flex flex-wrap">
                <div v-for="product in products"
                  class="product-item col-lg-4 col-md-6 col-sm-6">
                  <div class="image-holder">
                    
                  </div>
                  <div class="cart-concern">
                    <div
                      class="cart-button d-flex justify-content-between align-items-
center">
                      <button type="button"
                        class="btn-wrap cart-link d-flex align-items-center"
@click.prevent="addToCart(product)">add to cart
                        <i class="icon icon-arrow-io"></i>
                      </button>
                      <button type="button" class="wishlist-btn">
                        <i class="icon icon-heart"></i>
                      </button>
                    </div>
                  </div>
                  <div class="product-detail">
                    <h3 class="product-title">
                      <router-link :to="{name: 'products.show', params: {id:
product.id }}">{{ product.title }}</router-link>
                    </h3>
                    <div class="item-price text-primary">{{ product.price }}$</div>
                  </div>
                </div>
              </div>
            </div>
          </div>
          <nav class="navigation paging-navigation text-center padding-medium"
            role="navigation">
            <div class="pagination loop-pagination d-flex justify-content-center" v-
              if="pagination.last_page > 1">
              <a href="#" class="pagination-arrow d-flex align-items-center" v-
                if="pagination.current_page !== 1"
                @click.prevent="getProducts(pagination.current_page - 1)">
                <i class="icon icon-arrow-left"></i>
              </a>
              <a aria-current="page" class="page-numbers" v-for="link in pagination.links"
                @click.prevent="getProducts(link.label)" :class="link.active ? 'current' : "">
                <template v-if="Number(link.label) &&
                (pagination.current_page - link.label < 2
                && pagination.current_page - link.label > -2) ||
                Number(link.label) === 1 || Number(link.label) === pagination.last_page">
                {{ link.label }}
              </template>
              <template v-if="Number(link.label) &&
                pagination.current_page !== 3 &&
                pagination.current_page - link.label === 2 ||
                Number(link.label) &&
                pagination.current_page !== pagination.last_page - 2 &&
                pagination.current_page - link.label === -2">

```

```

        <a>...</a>
      </template>
    </a>
    <a href="#" class="pagination-arrow d-flex align-items-center" v-
if="pagination.current_page !== pagination.last_page"
    @click.prevent="getProducts(pagination.current_page + 1)">
      <i class="icon icon-arrow-right"></i>
    </a>
  </div>
</nav>
</div>
</section>
<aside class="col-md-3">
  <div class="sidebar">
    <div class="widgets widget-product-tags">
      <h5 class="widget-title">Tags</h5>
      <ul class="product-tags sidebar-list list-unstyled">
        <li class="tags-item" v-for="tag in filterList.tags">
          <a href="#" @click.prevent="addTag(tag.id)">{{ tag.title }}</a>
        </li>
      </ul>
    </div>
    <div class="widgets widget-product-brands">
      <h5 class="widget-title">Colors</h5>
      <ul class="product-tags sidebar-list list-unstyled">
        <li class="tags-item" v-for="color in filterList.colors">
          <div href="#"
            :style="`background: #${ color.title }; border: 1.5px solid black;`"
            class="color-square" :title="color.title"
            @click.prevent="addColor(color.id)"></div>
        </li>
      </ul>
    </div>
    <div class="widgets widget-price-filter">
      <h5 class="widget-title">Filter By Price</h5>
      <div class="slider-box">
        <div id="price-range" class="slider"></div>
        <div class="output-price">
          <label for="priceRange">Price: </label>
          <input type="text" id="priceRange" readonly>
        </div>
      </div>
      <button @click.prevent="filterProducts" class="btn btn-outline-
dark">Filter</button>
    </div>
  </div>
</aside>
</div>
</div>
</div>
</div>
</template>
<script>
export default {
  name: "Index",
  mounted() {
    $(document).trigger('changed')
    this.getProducts()
    this.getFilterList()
  },
  data() {
    return {
      products: [],
      filterList: [],

```

```

    categories: [],
    colors: [],
    tags: [],
    prices: [],
    pagination: [],
  }
},
methods: {
  addToCart(product){

    let cart = localStorage.getItem('cart')
    let newProduct = [
      {
        'id': product.id,
        'image_url': product.image_url,
        'title': product.title,
        'price': product.price,
        'count': product.count,
        'qty': 1
      }
    ]
    if (!cart) {
      localStorage.setItem('cart', JSON.stringify(newProduct));
    } else {
      cart = JSON.parse(cart)
      cart.forEach(productInCart => {
        if(productInCart.id === product.id){
          productInCart.qty = Number(productInCart.qty) + 1
          newProduct = null
        }
      })
      Array.prototype.push.apply(cart, newProduct)
      localStorage.setItem('cart', JSON.stringify(cart))
      console.log(cart);
    }
  },
  filterProducts(){
    let prices = $('#priceRange').val()
    this.prices = prices.replace(/[s+][$/g, ").split('-')
    this.getProducts()
  },
  addCategory(id){
    if(!this.categories.includes(id)){
      this.categories.push(id)
    } else{
      this.categories = this.categories.filter( elem =>{
        return elem !== id
      })
    }
  },
  addTag(id){
    if(!this.tags.includes(id)){
      this.tags.push(id)
    } else{
      this.tags = this.tags.filter( elem =>{
        return elem !== id
      })
    }
  },
  addColor(id){
    if(!this.colors.includes(id)){
      this.colors.push(id)
    } else{

```

```

        this.colors = this.colors.filter( elem =>{
            return elem !== id
        })
    }
},
getProducts(page = 1) {
    this.axios.post('api/products', {
        'categories': this.categories,
        'colors': this.colors,
        'tags': this.tags,
        'prices': this.prices,
        'page': page
    })
    .then(res => {
        this.products = res.data.data
        this.pagination = res.data.meta
    })
},
getFilterList() {
    this.axios.get('api/products/filters')
    .then(res => {
        this.filterList = res.data
        if ($("#price-range").length) {
            $("#price-range").slider({
                range: true,
                min: this.filterList.price.min,
                max: this.filterList.price.max,
                values: [this.filterList.price.min, this.filterList.price.max],
                slide: function(event, ui) {
                    $("#priceRange").val("$" + ui.values[0] + " - $" + ui.values[1]);
                }
            });
            $("#priceRange").val("$" + $("#price-range").slider("values", 0) + " - $" + $("#price-
range").slider("values", 1));
        }
    })
}
}
}
</script>

<style scoped>
.color-square {
    width: 30px;
    height: 30px;
    display: inline-block;
    margin-right: 5px;
}

/* Стили для блоку, що відображає вибрані значення ціни */
.output-price {
    margin-top: 20px;
    font-size: 14px;
    color: #333;
}

/* Стили для самого слайдера */
.slider {
    width: 100%; /* Ширина слайдера */
    height: 20px; /* Висота слайдера */
    -webkit-appearance: none; /* Видалення стандартного вигляду */
    appearance: none;
    outline: none; /* Видалення контуру */
    background: #d3d3d3;
    opacity: 0.7;
    -webkit-transition: .2s;

```

```

    transition: opacity .2s;
  }

.slider:hover {
  opacity: 1;
}

.slider::-ms-thumb {
  width: 20px;
  height: 20px;
  background: #4caf50;
  border: none;
  border-radius: 50%;
  cursor: pointer;
}

.output-price {
  display: flex;
  align-items: center;
  margin-top: 10px;
}

.output-price input[type="text"] {
  border: none;
  outline: none;
  font-size: inherit;
  background: #ffffff;
  margin-bottom: 2px;
}

.output-price label {
  font-size: 16px;
  margin-right: 10px;
}
</style>

```

Лістинг Login.vue

```

<template>
  <div>
    <section class="site-banner padding-small">
      <div class="container">
        <div class="row">
          <div class="col-md-12">
            <div class="breadcrumbs">
              <span class="item">
                <router-link to="/">Home </router-link>
              </span>
              <span class="item">Login</span>
            </div>
          </div>
        </div>
      </div>
    </section>

    <form class="d-flex justify-content-center">
      <div class="form-group w-25 form-container">
        <input v-model="email" type="email" class="form-control mt-3 mb-3"
placeholder="Email" required>
        <input v-model="password" type="password" class="form-control mb-3"
placeholder="Password" required>
        <div v-if="error" class="text-danger">{{ this.error }}</div>
        <button type="submit" class="btn btn-outline-dark"
@click.prevent="login">Login</button>

```

```

    <p class="mt-3 mb-3">First time here? <router-link :to="{ name: 'user.registration' }">Create
an GACY account</router-link></p>
  </div>
</form>
</div>
<hr>
</template>

<script>
export default {
  name: "Login",
  mounted() {
    $(document).trigger('changed')
  },
  data() {
    return {
      email: null,
      password: null,
      error: null,
    }
  },
  methods: {
    login() {
      this.axios.post('api/auth/login', {
        email: this.email,
        password: this.password
      })
      .then(res => {
        localStorage.setItem('access_token', res.data.access_token)
        this.$router.push({ name: 'main' })
      }).catch (error => {
        this.error = error.response.data.error
      })
    },
  }
}
</script>

<style scoped>
.form-group {
  display: flex;
  flex-direction: column;
}

input[type="email"],
input[type="password"],
button {
  margin: 0.5rem 0;
}

.form-container {
  border: 2px solid #c4c4c4;
  border-radius: 10px;
  padding: 20px;
}
</style>

```

Додаток Г
Презентаційний матеріал

**КВАЛІФІКАЦІЙНА
РОБОТА БАКАЛАВРА**

**МЕТОД РЕКОМЕНДАЦІЙ ЗА АНАЛІЗОМ
ПОПЕРЕДНЬОЇ КУПІВЕЛЬНОЇ
ПОВЕДІНКИ КЛІЄНТА ЗАСОБАМИ
МАШИННОГО НАВЧАННЯ ДЛЯ
ВЕБСИСТЕМИ ІГРОВИХ КОНСОЛЕЙ ТА
АКСЕСУАРІВ**

**ВИКОНАЛА: СТУДЕНКА 4 КУРСУ, ГРУПИ КН-20-2 ВОВК СТЕФАНІЯ
КЕРІВНИК: ЗАВ. КАФЕДРИ КН, Д.Т.Н., ПРОФЕСОР ОЛЕКСАНДР БАРМАК**

АКТУАЛЬНІСТЬ

У СЬОГОДЕННІ КОРИСТУВАЧІ ВИМАГАЮТЬ ПЕРСОНАЛІЗОВАНОГО ПІДХОДУ ДО БУДЬ-ЯКИХ ДІЙ, ПОЧИНАЮЧИ З ОТРИМАННЯ ЯКІСНОГО КОНТЕНТУ У ВІДЕОІГРАХ: ВІД СКЛАДНОСТІ ТА ГЕЙМПЛЕЮ ДО ВІДЧУТТЯ ІМПАКТУ Й ЕМОЦІЙ, ЗАКІНЧУЮЧИ ВИБОРОМ ТОВАРІВ ТА ПОСЛУГ ПІД ЧАС ОНЛАЙН-ПОКУПОК. ОСКІЛЬКИ БІЛЬШІСТЬ ЛЮДЕЙ СЬОГОДНІ ЗДІЙСНЮЮТЬ ПОКУПКИ В ІНТЕРНЕТІ, ЩОБ ВИДІЛИТИСЯ СЕРЕД КОНКУРЕНТІВ ТА ЗАБЕЗПЕЧИТИ СОБІ НАЙКРАЩІ УМОВИ, НЕОБХІДНО НАДАВАТИ ПЕРСОНАЛІЗОВАНІ РЕКОМЕНДАЦІЇ, ЩО ВІДПОВІДАЮТЬ ПОТРЕБАМ ТА ІНТЕРЕСАМ КЛІЄНТІВ. ТОМУ, СТВОРЕННЯ ПЕРСОНАЛІЗОВАНИХ РЕКОМЕНДАЦІЙ ДЛЯ КОРИСТУВАЧІВ Є НЕВІД'ЄМНОЮ ЧАСТИНОЮ КОЖНОГО БІЗНЕСУ, ЯКА СПРІЯЄ ПІДВИЩЕННЮ КОНКУРЕНТОСПРОМОЖНОСТІ ТА ЗАДОВОЛЕННЯ ПОТРЕБ СПОЖИВАЧІВ. ДОСЛІДЖЕННЯ ЦЬОЇ ТЕМИ СТАЄ АКТУАЛЬНОЮ ЧАСТИНОЮ ВЕБСИСТЕМ, ЩО СПРЯМОВАНІ НА РОЗВИТОК ТА УСПІШНУ ДІЯЛЬНІСТЬ В ОНЛАЙН-СЕРЕДОВИЩІ ПОКУПОК.



МЕТА ТА ЗАДАЧІ РОБОТИ

Мета

МЕТОЮ КВАЛІФІКАЦІЙНОЇ РОБОТИ БАКАЛАВРА Є ПОКРАЩЕННЯ ЕФЕКТИВНОСТІ ОТРИМАННЯ ПЕРСОНАЛІЗОВАНИХ РЕКОМЕНДАЦІЙ ЗА АНАЛІЗОМ ПОПЕРЕДНЬОЇ КУПІВЕЛЬНОЇ ПОВЕДІНКИ КЛІЄНТА ЗАСОБАМИ МАШИННОГО НАВЧАННЯ ДЛЯ ВЕБСИСТЕМИ ІГРОВИХ КОНСОЛЕЙ ТА АКЕСУАРІВ



Задачі

- ПРОВЕСТИ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ДЛЯ ЗАДАЧІ ОТРИМАННЯ ПЕРСОНАЛІЗОВАНИХ РЕКОМЕНДАЦІЙ ЗА АНАЛІЗОМ ПОПЕРЕДНЬОЇ КУПІВЕЛЬНОЇ ПОВЕДІНКИ КЛІЄНТА ЗАСОБАМИ МАШИННОГО НАВЧАННЯ ДЛЯ ВЕБСИСТЕМИ ІГРОВИХ КОНСОЛЕЙ ТА АКЕСУАРІВ;
- ВИКОНАТИ АНАЛІЗ СПОСОБІВ ФОРМУВАННЯ РЕКОМЕНДАЦІЙ НАСТУПНИХ ПОКУПОК ДЛЯ КОРИСТУВАЧА;
- ВИКОНАТИ ОГЛЯД ТА АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ЩОДО РЕАЛІЗАЦІЇ ПОДІБНИХ ЗАДАЧ; РОЗРОБИТИ МЕТОД РЕКОМЕНДАЦІЙ ЗА АНАЛІЗОМ ПОПЕРЕДНЬОЇ КУПІВЕЛЬНОЇ ПОВЕДІНКИ КЛІЄНТА ЗАСОБАМИ МАШИННОГО НАВЧАННЯ;
- СПРОЄКТУВАТИ СТРУКТУРУ ВЕБСИСТЕМИ ІГРОВИХ КОНСОЛЕЙ ТА АКЕСУАРІВ, А ТАКОЖ БАЗУ ДАНИХ ДЛЯ ДАНОЇ СИСТЕМИ ТА МЕТОДУ; ОБРАТИ СУЧАСНІ ЗАСОБИ РЕАЛІЗАЦІЇ МЕТОДУ РЕКОМЕНДАЦІЙ ЗА АНАЛІЗОМ ПОПЕРЕДНЬОЇ КУПІВЕЛЬНОЇ ПОВЕДІНКИ КЛІЄНТА ЗАСОБАМИ МАШИННОГО НАВЧАННЯ ДЛЯ ВЕБСИСТЕМИ ІГРОВИХ КОНСОЛЕЙ ТА АКЕСУАРІВ;
- РЕАЛІЗУВАТИ ВЕБСИСТЕМУ ІГРОВИХ КОНСОЛЕЙ ТА АКЕСУАРІВ ТА ЗАПРОПОНОВАНИЙ МЕТОД;
- ПРОТЕСТУВАТИ ПРОГРАМНУ РЕАЛІЗАЦІЮ НА КОРЕКТНІСТЬ РОБОТИ ВСІХ ФУНКЦІЙ;
- ПОКАЗАТИ ЕФЕКТИВНІСТЬ СИСТЕМИ З ЗАПРОПОНОВАНИМ МЕТОДОМ.

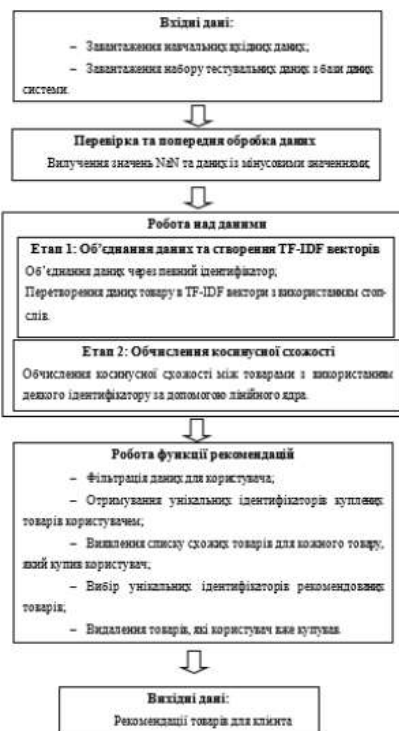


СХЕМА МЕТОДУ РЕКОМЕНДАЦІЙ ТОВАРІВ ЗА АНАЛІЗОМ ПОПЕРЕДНЬОЇ КУПІВЕЛЬНОЇ ПОВЕДІНКИ КЛІЄНТА

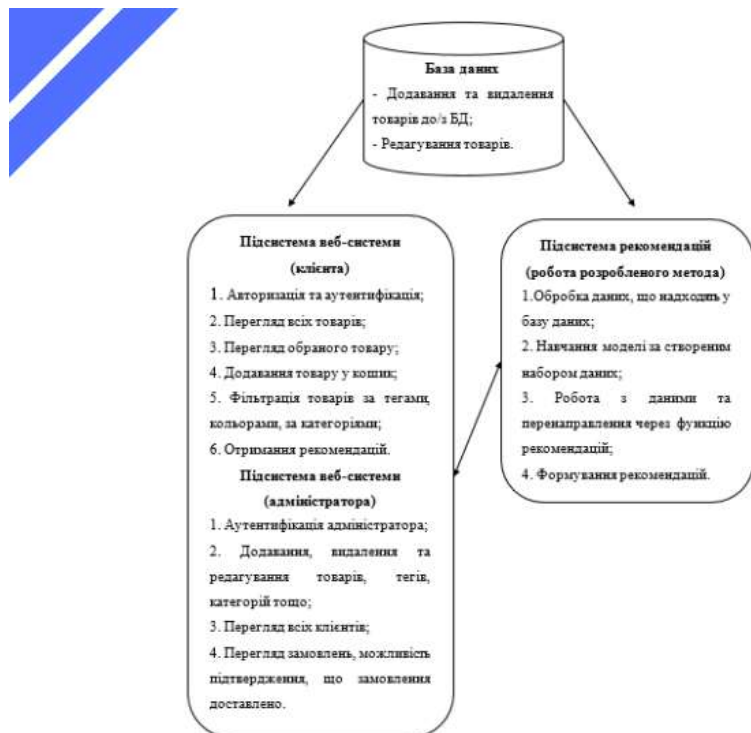
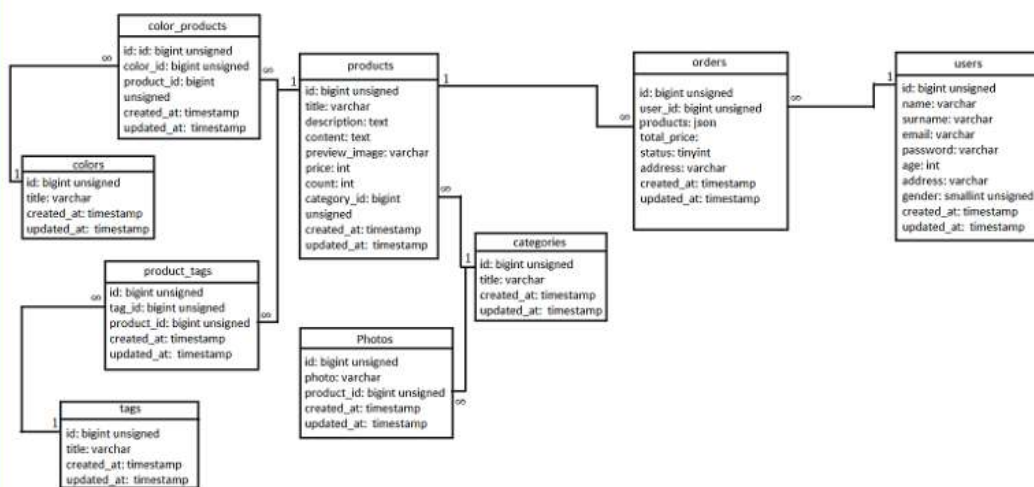


СХЕМА СТРУКТУРИ ВЕБСИСТЕМИ, ЯКА ВИКОРИСТОВУЄТЬСЯ ДЛЯ ДЕМОНСТРАЦІЇ МЕТОДУ

БАЗА ДАНИХ



ПРОГРАМНА РЕАЛІЗАЦІЯ

Продукт	Кількість	Проміжний Підсумок
 Dead Space 44\$	<input type="button" value="-"/> <input type="text" value="1"/> <input type="button" value="+"/>	44\$ <input type="button" value="X"/>

Сума Кошика

Всього 44\$

Продовжити покупки

Оформити замовлення

Для оформлення замовлення додайте адресу доставки

Україна, Хмельницька обл., м. Хмельницький, вул. Кам'янецька 110

Замовити товари

ФОРМУВАННЯ
ЗАМОВЛЕННЯ ДЛЯ
РОБОТИ МЕТОДУ
РЕКОМЕНДАЦІЙ

ПРОГРАМНА РЕАЛІЗАЦІЯ

ВИВЕДЕННІ РЕКОМЕНДАЦІЇ
ДЛЯ КОРИСТУВАЧА

Вам Може Сподобатися



God Of War Ragnarok
70\$



Nintendo Switch Charger
36\$



Nintendo Switch Pro
Controller
120\$



Nintendo Switch
Screen Protector
10\$

Дослідження ефективності

1

ЕКСПЕРТНА ОЦІНКА ШТУЧНОГО ІНТЕЛЕКТУ ЧАТ-БОТУ

При тестуванні, сформовані рекомендації методом і чатом, як експертом, збігалися в середньому на 60%.

2

ПЕРЕВІРКА ЗА ДОПОМОГОЮ МЕТРИК

Результати NDCG показали рівень релевантності вище середнього запропонованих рекомендацій. Це свідчить про те, що рекомендації були більш менш точними, але при ранжуванні інколи деякі пропозиції ставали вище, ніж ті, що були більш релевантними.

3

A/B-ТЕСТ

При таких рекомендаціях принаймні 54% респондентів придбали б один товар із запропонованих рекомендацій.



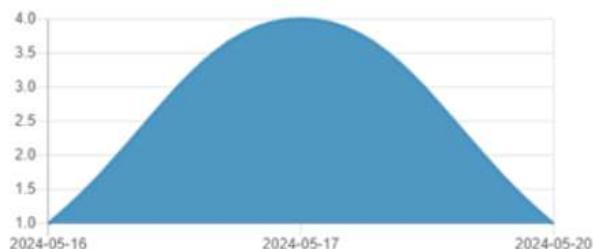
ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ МЕТОДУ

Замовлення

6

Замовлень за весь час

↑ 300%
За 2 минулих тижня



ВИСНОВКИ

ПРОВЕДЕНО

огляд способів формування персональних рекомендацій, а також наявних засобів реалізації рекомендацій наступних покупок, поставлена мета та завдання роботи

СПРОЄКТОВАНО

схему роботи методу рекомендацій, структуру вебінтерфейсу інформаційної системи та бази даних

РЕАЛІЗОВАНО

метод рекомендацій за аналізом попередньої купівельної поведінки клієнта засобами машинного навчання для вебсистеми ігрових консолей та аксесуарів, також проведено тестування та дослідження ефективності методу пропозицій

Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 2.0%

Словники перевірки: en_US, ru_RU, ua_UA. Помилки в документах: 10%

ID: 129527 Назва: КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА на тему Метод рекомендацій за аналізом попередньої купівельної поведінки клієнта засобами машинного навчання для вебсистеми ігрових консолей та аксесуарів Додано в БД: 2024-06-10 Автора: Стефанія ВОВК Керівники: Олександр БАРМАК Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	92134	1383	2483 (3%)	42 (3%)

Джерело плагиату

ID	Опис	Наявність плагиату в документі	
		Символи	Лексеми



Ім'я користувача:
Кафедра КН

Дата перевірки:
10.06.2024 19:53:40 EEST

Дата звіту:
10.06.2024 19:59:06 EEST

ID перевірки:
1016344045

Тип перевірки:
Doc vs Internet + Library

ID користувача:
100005671

Назва документа: КН-20-2 Воєн ЗАПИСКА

Кількість сторінок: 75 Кількість слів: 14479 Кількість символів: 115529 Розмір файлу: 2.03 MB ID файлу: 1016145563

3.44% Схожість

Найбільша схожість: 1.6% з джерелом з Бібліотеки (ID файлу: 1016139735)

2.57% Джерела з Інтернету 318

Сторінка 77

2.00% Джерела з Бібліотеки 143

Сторінка 79

0% Цитат

Вилучення цитат вимкнено

Вилучення списку бібліографічних посилань вимкнено

0% Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи 1

**РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ КАФЕДРИ КОМП'ЮТЕРНИХ НАУК
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ**

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Метод рекомендацій за аналізом попередньої купівельної поведінки клієнта засобами машинного навчання для вебсистеми ігрових консолей та аксесуарів

Автор: студентка групи КН-20-2 Стефанія ВОВК

Спеціальність: 122 – Комп'ютерні науки

Освітня програма: освітньо-професійна

Науковий керівник: д.т.н., проф. Олександр БАРМАК

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	<i>відповідає</i>
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданій поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданій поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

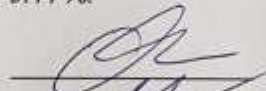
Запозичення, виявлені в роботі Стефанії ВОВК, не є плагіатом, оскільки: запозичення розміщені в розділі огляду існуючих підходів, не описують безпосередньо авторську роботу і не стосуються її результатів; усі запозичення фрагментарні; до запозичень входять фрагменти що не мають авторства і містять поширені конструкції; серед запозичень знаходяться загальновідомі терміни та скорочення.

Обсяг запозичень, визначений системами виявлення збігів/ідентичності/схожості, складає:

- за системою Anti-Plagiarism – 2%;

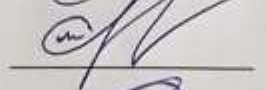
- за системою Unicheck – 3.44 %.

Керівник роботи



Олександр БАРМАК

Гарант ОП



Олександр МАЗУРЕЦЬ

Завідувач кафедри КН



Олександр БАРМАК



ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
МОН УКРАЇНИ

Кафедра комп'ютерних наук



РЕЦЕНЗІЯ

на кваліфікаційну роботу бакалавра

студентка *гр. КН-20-2 Вовк Стефанії Віталіївни*

за темою: Метод рекомендацій за аналізом попередньої купівельної поведінки клієнта засобами машинного навчання для вебсистеми ігрових консолей та аксесуарів

1. Актуальність обраної теми

З розвитком електронної комерції та зростанням кількості онлайн-магазинів важливо забезпечити індивідуальний підхід до кожного клієнта. Використання методів машинного навчання для аналізу попередньої купівельної поведінки дозволяє створювати персоналізовані рекомендації, що підвищують задоволення клієнтів та збільшують продажі. Тому актуальність такої теми роботи є безсумнівною.

2. Повнота розкриття мети та завдань роботи

Мета роботи була повністю розкрита, а завдання виконані на високому рівні, зокрема збір та аналіз даних предметної області, а також реалізація програмної системи з впровадженням рекомендаційної системи.

3. Зміст кожного розділу роботи

Записка кваліфікаційної роботи бакалавра містить три розділи. У першому розділі подано теоретичні основи рекомендаційних систем та методів машинного навчання, що використовуються для створення рекомендацій на основі аналізу попередньої купівельної поведінки клієнта. У другому розділі достатньо розписано метод рекомендацій за аналізом попередньої купівельної поведінки клієнта засобами машинного навчання, а також спроектовано всі необхідні модулі вебсистеми ігрових консолей та аксесуарів. Третій розділ описує процес розробки методу та інформаційної системи, включаючи її архітектуру, реалізацію, результати тестування та дослідження ефективності роботи методу у вебсистемі.

4. Оцінка розробленої інформаційної системи, її практична цінність

Розроблена вебсистема дозволяє автоматизувати процес створення персоналізованих рекомендацій для клієнтів, що може суттєво підвищити ефективність маркетингових стратегій та рівень задоволеності клієнтів. Створена вебсистема має вигляд інтернет-магазину, де клієнти можуть переглядати та оформляти замовлення таких товарів як: ігрові консолі та аксесуари. Система інтегрована з базою даних, що зберігає інформацію про попередні покупки клієнтів, та алгоритмами машинного навчання, які аналізують ці дані для створення релевантних рекомендацій.

5. Якість оформлення кваліфікаційної роботи бакалавра

Якість оформлення роботи відповідає усім вимогам. Текст чітко структурований. Графіки, таблиці та інші візуальні матеріали чітко ілюструють ключові моменти роботи та добре вписуються у загальний контекст.

6. Недоліки кваліфікаційної роботи бакалавра

Доцільно було б розширити навчальну та тестову множину знань, а також розглянути та реалізувати альтернативні методи машинного навчання та їх порівняння за ефективністю для конкретного завдання.

7. Загальний висновок (допускається чи не допускається до захисту), та оцінка на яку заслуговує кваліфікаційна робота.

Враховуючи рівень виконання та забезпечення усіх необхідних вимог, робота може бути допущена до захисту. Рекомендована оцінка «відмінно».

Рецензент

доцент кафедри ІТБ Лисиня О.М.



ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
МОН УКРАЇНИ



Кафедра комп'ютерних наук

ВІДГУК НАУКОВОГО КЕРІВНИКА на кваліфікаційну роботу бакалавра

студентки *гр. КН-20-2 Вовк Стефанії Віталіївни*

за темою Метод рекомендацій за аналізом попередньої купівельної поведінки клієнта засобами машинного навчання для вебсистеми ігрових консолей та аксесуарів

1. Актуальність теми

Оскільки більшість людей сьогодні здійснюють покупки в інтернеті, щоб виділитися серед конкурентів та забезпечити собі найкращі умови, необхідно надавати персоналізовані рекомендації, що відповідають потребам та інтересам клієнтів. Тому, створення персоналізованих рекомендацій для користувачів є невід'ємною частиною кожного бізнесу, яка сприяє підвищенню конкурентоспроможності та задоволення потреб споживачів. Для ефективного використання рекомендацій у вебсистемах необхідно спроектувати та розробити метод, який би дозволив сформувати персональні рекомендації за попередньою купівельною поведінкою клієнта. Застосування такого методу стає актуальним для вебсистем, та спрямоване на розвиток та успішну діяльність в онлайн-середовищі покупок.

2. Відповідність роботи предметній області Стандарту спеціальності 122 Комп'ютерні науки

За стандартом, а саме описом предметної області, об'єктами вивчення та діяльності є математичні, інформаційні, імітаційні моделі реальних явищ, об'єктів, систем і процесів та методи і технології отримання, зберігання, обробки, передачі та використання інформації. Метою роботи саме є покращення ефективності отримання персоналізованих рекомендацій за аналізом попередньої купівельної поведінки клієнта засобами машинного навчання для вебсистеми ігрових консолей та аксесуарів. При вирішенні поставленої задачі використано математичні моделі, методи та алгоритми розв'язання теоретичних і прикладних задач, що виникають при розробці вказаного методу. Тому результати виконання кваліфікаційної роботи бакалавра відповідають стандарту бакалавра спеціальності 122 – Комп'ютерні науки.

3. Професійні та особистісні якості бакалавра

При роботі над кваліфікаційною роботою бакалавра Вовк Стефанія Віталіївна проявила себе кваліфікованим фахівцем та дисциплінованою студенткою, вчасно

виконуючи поставлені задачі дослідження. Як в процесі написання пояснювальної записки, так і при розробці прикладного програмного забезпечення проявила достатні для одержання успішного результату компетентності та результати навчання. Опанувала професійні скіли за напрямком «Комп'ютерні науки» та достатньо значний софт скіл.

4. Ступінь самостійності під час виконання кваліфікаційної роботи

Одержані в роботі результати є наслідком особистої діяльності студентки, яка самостійно виконувала всі поставлені задачі.

5. Ступінь оволодіння методами дослідження

При реалізації кваліфікаційної роботи показала достатній рівень компетентностей та володіння необхідними інструментами та обладнанням, методами, методиками та технологіями предметної області комп'ютерних наук.

6. Повнота та якість розкриття теми роботи

Тема роботи в повній мірі обґрунтована й розкрита, проведено аналіз актуальності та відомих досліджень в межах обраної теми, поставлені завдання, які у роботі виконані, та розроблено програмне забезпечення для валідації та верифікації запропонованого метода.

7. Логічність, послідовність, аргументованість, літературна грамотність викладення матеріалу

Структура роботи та послідовність викладення логічні та відповідають поставленій меті. Викладення матеріалу послідовне, аргументоване, літературно грамотне.

8. Можливість практичного застосування кваліфікаційної роботи бакалавра, окремих її частин

Розроблений у роботі метод та його програмна реалізація може бути використана для покращення доходу власника вебсистеми та пропонує користувачам системи персоналізовані рекомендації, які враховують попередні замовлення.

9. Висновок про можливість допуску кваліфікаційної роботи бакалавра до захисту, на яку оцінку заслуговує робота

Враховуючи високий рівень виконання та забезпечення усіх необхідних вимог, робота може бути допущена до захисту. Рекомендована оцінка «відмінно».

Керівник



д.т.н., проф. зав. каф. КН Олександр БАРМАК