

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЗАПОРІЗЬКА ПОЛІТЕХНІКА»

Факультет комп'ютерних наук та технологій
Кафедра комп'ютерних систем та мереж

Пояснювальна записка

до дипломного проекту (роботи)

магістра

(ступінь вищої освіти)

на тему ВЕБПЛАТФОРМА ПІДТРИМКИ ВИБОРУ ТОВАРІВ З
ЕЛЕМЕНТАМИ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ

Виконав: студент 2 курсу, групи КНТ-514м
Спеціальності: 123 Комп'ютерна інженерія
(код і найменування спеціальності)

Освітня програма (спеціалізація):

Комп'ютерні системи та мережі

ПАЛЬЧИК Т.В.

(ПРИЗВИЩЕ та ініціали)

Керівник

ГОЛУБ Т.В.

(ПРИЗВИЩЕ та ініціали)

Рецензент

ХВОРОСТЯН В.І.

(ПРИЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет Комп'ютерних наук і технологій
 Кафедра «Комп'ютерні системи та мережі»
 Ступінь вищої освіти магістерський
 Спеціальність 123 Комп'ютерна інженерія
 (код і найменування)
 Освітня програма (спеціалізація) Комп'ютерні системи та мережі
 (назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ
 Зав. кафедри Кудерметов Р.К.

“15” жовтня 2025 року

З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА

ПАЛЬЧИКА Тараса Валерійовича

(ПРІЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Вебплатформа підтримки вибору товарів з елементами інтелектуального аналізу

керівник проєкту (роботи) к.н.т, доцент, Голуб Тетяна Василівна
 (науковий ступінь, вчене звання, ПРІЗВИЩЕ, ім'я, по батькові)

затверджені наказом вищого навчального закладу від 30» жовтня 2025 року № 491

2. Строк подання студентом проєкту (роботи) 26 грудня 2025 року

3. Вихідні дані до проєкту (роботи) наявні методи та алгоритми функціонування рекомендаційних систем вибору товарів, підтримка контентного аналізу

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

Основи рекомендаційних систем. Проєктування програмної частини системи. Реалізація загальної структури програми. Експериментальна перевірка програмного забезпечення.

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

6. Консультанти розділів проєкту (роботи)

Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4	ГОЛУБ Т. В., доцент		
нормоконтроль	ЩЕРБАК Н.В., ст. викл.		

7. Дата видачі завдання 27.10.2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Аналіз наявних методів та алгоритмів підбору товарів, перегляд отриманих результатів, виділення недоліків та переваг	03.10.2025 р.	
2	Розробка вебплатформи інтелектуального підбору товарів у сфері електронної комерції.	10.11.2025 р.	
3	Розробка алгоритму роботи системи для підбору	15.11.2025 р.	
4	Реалізація системи підбору товарів	25.11.2025 р.	
5	Дослідження системи підбору товарів	30.11.2025 р.	
6	Оформлення отриманих результатів у ПЗ	05.12.2025 р.	
7	Оформлення графічного матеріалу	10.12.2025 р.	
8	Оформлення допоміжного матеріалу	15.12.2025 р.	

Студент _____ **Тарас ПАЛЬЧИК**
(підпис) (ім'я, ПРИЗВИЩЕ)Керівник проєкту (роботи) _____ **Тетяна ГОЛУБ**
(підпис) (ім'я, ПРИЗВИЩЕ)

РЕФЕРАТ

ПЗ: 81 с., 14 рис., 7 табл., 7 формул, 21 джерел.

API, ML-ПІДХОДИ, UI, ГІБРИДНІ МОДЕЛІ, ІНТЕЛЕКТУАЛЬНИЙ ПІДБІР ТОВАРІВ, РЕКОМЕНДАЦІЙНІ СИСТЕМИ, МОДУЛІ ПРОГРАМНОЇ ЧАСТИНИ.

Об'єкт дослідження - процес функціонування цифрових платформ підтримки вибору товарів у вебсередовищі.

Предмет дослідження - методи та моделі інтелектуального аналізу даних, критерії інтелектуальності систем підбору товарів, що використовуються для підвищення ефективності роботи цифрових вебплатформ.

Метою роботи є розробка інформаційної системи з інтегрованою цифровою платформою, яка забезпечує вдосконалену пошукову функцію на основі гібридної моделі.

Практична цінність роботи полягає у створенні готового рішення для надійного підбору товару з гарантією їх цілісності та можливістю верифікації. Розроблена система може бути застосована в різних сферах, де критично важлива швидкість підбору - від індивідуальних потреб до промислових масштабів.

Наукова новизна роботи полягає у поєднанні інтелектуальних методів аналізу даних з архітектурними рішеннями вебплатформи, орієнтованої на підвищення якості вибору товарів користувачем. Запропонована система не лише дозволяє автоматизувати процес рекомендацій, а й забезпечує адаптивність до змін у поведінці користувачів і контенті бази даних.

Практичне значення роботи полягає у створенні функціонального прототипу вебплатформи, який може бути використаний як навчальний, демонстраційний або комерційний інструмент. Галузь використання - вдосконалення механізмів рекомендацій у наявних онлайн-платформах, як навчальний приклад побудови гібридних інтелектуальних моделей.

ABSTRACT

Explanatory note to the master's work: 81 p., 14 figures, 7 tables, 7 formulas, 21 sources.

API, ML APPROACHES, UI, HYBRID MODELS, INTELLIGENT PRODUCT SELECTION, RECOMMENDATION SYSTEMS, SOFTWARE MODULES.

The object of the study is the process of functioning of digital platforms supporting the selection of goods in the web environment.

The subject of the study is methods and models of intelligent data analysis, criteria for the intelligence of product selection systems used to improve the efficiency of digital web platforms.

The purpose of the work is to develop an information system with an integrated digital platform that provides an improved search function based on a hybrid model.

The practical value of the work lies in creating a ready-made solution for reliable product selection with a guarantee of their integrity and the possibility of verification. The developed system can be applied in various areas where the speed of selection is critically important - from individual needs to industrial scales.

The scientific novelty of the work lies in the combination of intelligent data analysis methods with architectural solutions of a web platform focused on improving the quality of product selection by the user. The proposed system not only allows you to automate the recommendation process, but also provides adaptability to changes in user behavior and database content.

The practical significance of the work lies in the creation of a functional prototype of a web platform that can be used as an educational, demonstration or commercial tool. The field of use is the improvement of recommendation mechanisms in existing online platforms, as an educational example of building hybrid intelligent models.

ЗМІСТ

Скорочення та умовні позначки	7
Вступ.....	8
1 Основи рекомендаційних систем	10
1.1 Аналіз рекомендаційних систем при виборі товару.....	10
1.2 Аналіз технологій та моделей інтелектуального підбору товарів	13
1.3 Огляд програмних засобів розробки з інтелектуальними аналізом	17
1.4 Огляд апаратної частини при розробці рекомендаційного підбору	19
1.5 Особливості розробки платформи та дослідження	19
1.6 Висновок до розділу.....	21
2 Проєктування програмної частини системи	22
2.1 Модуль контентного аналізу та інтелектуального підбору	23
2.2 Модуль кластеризації та персоналізації користувача	25
2.3 Модуль персоналізації та інтелектуального формування і його взаємодія	26
2.4 Функціонування та взаємодія модулів, їх реалізація	27
2.5 Математична структура гібридної моделі інтелектуального підбору товарів..	30
2.6 Висновки до розділу.....	34
3. Реалізація загальної структури програми	35
3.1 Розробка практичної структурної схеми платформи	40
3.2 Розробка функціонального серверного API-вузла	42
3.3 Висновки до розділу.....	46
4 Експеримент та аналіз результатів веб-платформи.....	47
4.1 Моделювання платформи	47
4.2 Тестування платформи та оцінка працездатності шляхом порівняння	49
4.3 Висновки щодо експериментального дослідження	53
Висновки.....	55
Перелік джерел посилання	57
Додаток А Лістинг програми.....	59
Додаток Б Слайди презентації.....	74

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧКИ

- ПЗ – Пояснювальна записка;
- БД – База даних;
- ШІ – Штучний інтелект;
- РС – Персональний комп'ютер;
- TF-IDF – Term Frequency-Inverse Document Frequency - метод векторизації текстових даних;
- AWS – Amazon Web Services - хмарна платформа компанії Amazon;
- ORM – Object-Relational Mapping - об'єктно-реляційне відображення;
- UI – User Interface - Інтерфейс користувача;
- ML – Machine Learning - машинне навчання;
- GPU – Graphics Processing Unit - графічний процесор.
- IDE – Integrated Development Environment - інтегроване середовище розробки;
- GUI – Graphical User Interface - графічний інтерфейс користувача;
- API – Application Programming Interface - інтерфейс прикладного програмування.

ВСТУП

Сучасний розвиток інформаційних технологій, стрімке зростання кількості електронних сервісів та популяризація онлайн-торгівлі створили нові умови для взаємодії споживача і бізнесу. Сьогодні покупці очікують не лише швидкого доступу до товарів, а й персоналізованого вибору - такого, який враховує їхні інтереси, історію переглядів, фінансові можливості та навіть контекст попередніх дій. Тому наявність систем підтримки вибору товарів, інтегрованих у вебплатформу, є невід'ємною частиною сучасних інтернет-магазинів. Традиційні каталоги інтернет-магазинів, навіть з великою кількістю фільтрів і сортувань, не здатні повноцінно забезпечити індивідуальний підхід до кожного користувача. Людина витрачає значну кількість часу на пошук потрібного товару серед тисяч позицій, у той час як інтелектуальні алгоритми здатні зробити цей процес швидшим і точнішим. Тому впровадження елементів інтелектуального аналізу у вебплатформи є актуальним напрямом розвитку сучасних цифрових систем.

Особливе місце у цьому процесі займають рекомендаційні системи - інструменти, які використовують методи машинного навчання, статистики, кластеризації та гібридних алгоритмів для передбачення уподобань користувача. Серед найвідоміших прикладів - Amazon, Rozetka, OLX, які впроваджують власні механізми персоналізації: від простих фільтрів до складних нейронних мереж, що аналізують поведінку користувача. Проте більшість таких рішень є закритими або дорогими у впровадженні, що обмежує можливості малого й середнього бізнесу, а також навчальні чи дослідницькі ініціативи.

Мета дипломної роботи полягає у розробці комбінованої моделі інтелектуального підбору товарів, на основі порівнянь різних методів, яка поєднує контентно-орієнтований аналіз характеристик, кластеризацію параметрів та вагове ранжування для покращення якості рекомендацій і зручності користувача.

Для досягнення поставленої мети передбачено виконання таких завдань:

- провести аналітичний огляд сучасних вебплатформ і систем підтримки вибору товарів;
- визначити основні критерії інтелектуальності та підходи до персоналізації контенту;
- розробити методологію створення вебплатформи з елементами інтелектуального аналізу;
- створити архітектуру програмного рішення та реалізувати його прототип;
- провести експериментальну оцінку точності та ефективності інтелектуального підбору товарів.

Методи дослідження базуються на комплексному підході, який поєднує аналіз сучасних інформаційних технологій, методів машинного навчання, структурного та поведінкового моделювання користувача, а також принципів побудови архітектури вебзастосунків. У роботі використано елементи методів класифікації, колаборативної фільтрації, контентного аналізу та гібридних моделей рекомендаційних систем.

1 ОСНОВИ РЕКОМЕНДАЦІЙНИХ СИСТЕМИ

1.1 Аналіз рекомендаційних систем при виборі товару

Рекомендаційна (інтелектуальна) система - це комплекс апаратно-програмних засобів, який на основі накопичених знань, досвіду та алгоритмів машинного навчання здатний самостійно або з мінімальним втручанням людини вирішувати завдання, що раніше вимагали участі експерта. Основною метою таких систем є імітація процесів людського мислення - аналізу, узагальнення, прогнозування та прийняття рішень.

Основні характеристики (рис.1.1) системи відрізняються від звичайних програмних продуктів низкою ознак:

- автономність (персоналізація) - мінімальна залежність від людини під час прийняття рішень, що дозволяє системі функціонувати без постійного ручного керування;

- прогнозування - здатність застосовувати набуті знання у нових, раніше невідомих ситуаціях;

- пояснюваність (explainability) - можливість надати користувачу логічне обґрунтування прийнятого рішення. Це особливо важливо для систем, які мають приймати відповідальні рішення для (медицина, фінанси, безпека). Застосовуються для розширеної масштабованості системи;

- адаптивність - здатність змінювати свою поведінку під впливом нових даних або зовнішніх факторів. Наприклад, система рекомендацій товарів, отримуючи інформацію про зміну попиту, оновлює модель прогнозу;

- інтерактивність (learning capability) - здатність покращувати результати роботи на основі минулого досвіду. У цьому проявляється ключова відмінність від традиційних алгоритмічних програм.

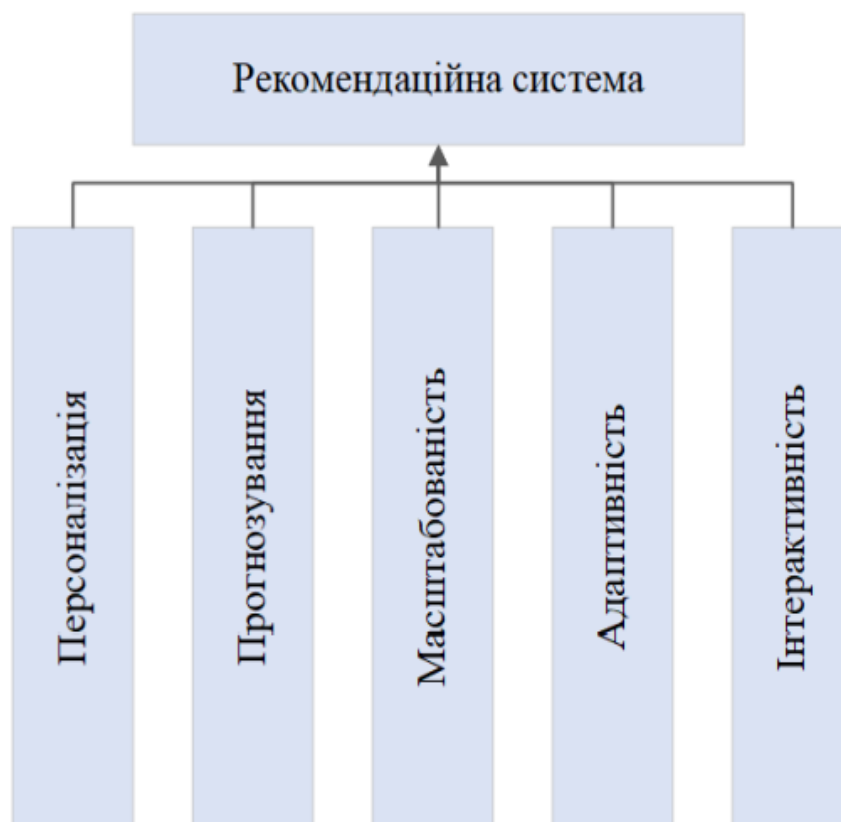


Рисунок 1.1 - Основні характеристики рекомендаційних систем

Область застосування рекомендаційних систем охоплюють методи та моделі, спрямовані на автоматизований відбір товарів відповідно до певних критеріїв таких як:

- характеристик продукції;
- поведінкових даних користувача;
- контексту взаємодії або історії взаємодії з платформою.

Вони покликані зменшити інформаційне навантаження, звузити простір пошуку та надати користувачу обґрунтовані варіанти вибору. У межах предметної області виділяють кілька основних класів рекомендаційних систем, кожен з яких має свої переваги, обмеження та сфери застосування.

До класичних підходів належать контентно-орієнтовані системи, які формують рекомендації на основі характеристик товарів і схожості атрибутів продуктів. Такий підхід особливо ефективний у сферах, де ключову роль відіграють технічні параметри, такі як електроніка, комп'ютерне обладнання чи

IoT-пристрої. Колаборативні методи використовують статистику взаємодії великої кількості користувачів і дозволяють виявити приховані закономірності у виборі товарів, проте залежать від масштабів платформи та наявності користувацьких даних. Поряд із цим виокремлюють гібридні системи, які поєднують властивості кількох методів для підвищення точності рекомендацій та усунення недоліків окремих підходів.

Однією з областей застосування таких систем у сучасних цифрових платформах є процес підбору товарів, який поступово перетворився на складне аналітичне завдання, що потребує обробки великої кількості параметрів, зіставлення альтернатив та врахування індивідуальних потреб користувача. Це обумовлено як швидким розширенням асортименту онлайн-магазинів, так і зростанням вимог до точності підбору, оскільки користувач очікує отримати релевантну інформацію без значних витрат часу. Таким чином, рекомендаційні системи стали одним із ключових інструментів, що визначають ефективність взаємодії користувача з вебплатформою та впливають на загальний рівень задоволеності сервісом.

У межах завдання створення вебплатформи для підбору технічних товарів важливо враховувати специфіку предметної області. Велика кількість параметрів, значна варіативність моделей, необхідність перевірки сумісності та потреба в точному формуванні вимог. Тому для таких систем найбільш релевантними є підходи, що дозволяють здійснювати інтелектуальну обробку структурованих даних, аналізувати характеристики товарів та відсіювати нерелевантні позиції ще на ранньому етапі формування результатів.

Саме на цьому етапі рекомендаційні системи переходять від загальних алгоритмічних моделей до інтелектуальних методів, що включають контентно-орієнтовану фільтрацію, кластеризацію характеристик, формування профілю користувача та застосування вагових коефіцієнтів для багатокритеріального порівняння товарів [1; 11].

1.2 Аналіз технологій та моделей інтелектуального підбору товарів

В умовах зростаючої складності товарних каталогів та значного обсягу структурованих даних особливу увагу необхідно приділяти методам, що забезпечують не лише фільтрацію за параметрами, а й інтелектуальне ранжування результатів відповідно до потреб користувача. Далі наведено основні підходи, які мають практичну цінність для реалізації вебплатформи вибору товарів.

Контентно-орієнтована модель (Content-Based Filtering) базується на аналізі характеристик товарів. На відміну від колаборативних методів (рис.1.2), цей підхід не потребує великої кількості користувацьких даних, що робить його актуальним для нових платформ або вузьких товарних ніш.

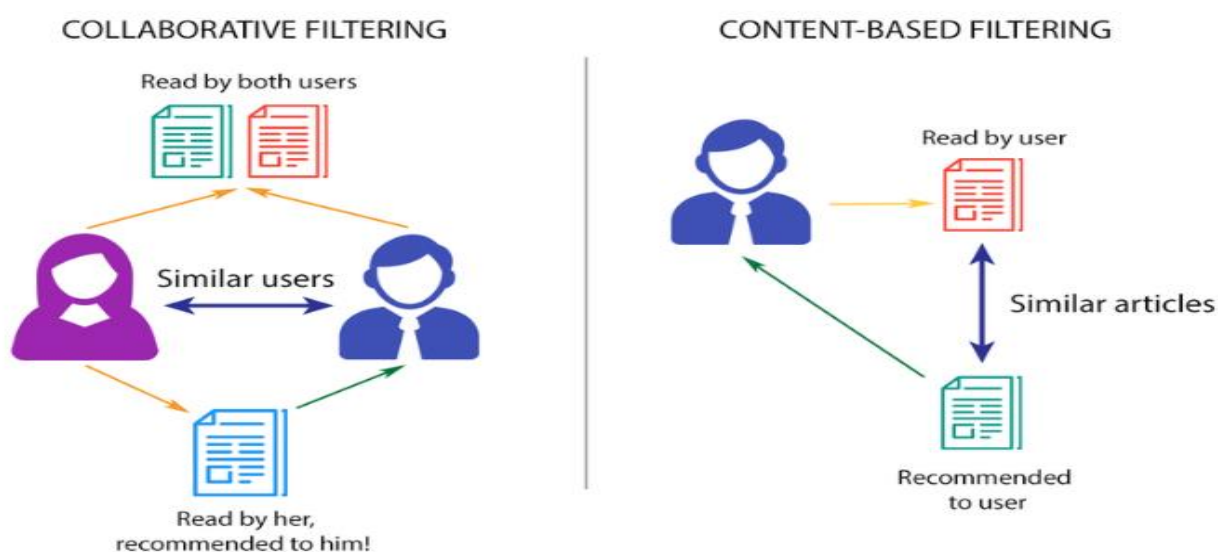


Рисунок 1.2 - Різниця між колаборативною та контентно-орієнтованою моделями

Перевагою підходу є висока точність у сферах, де технічні характеристики визначають якість вибору (ноутбуки, комплектуючі ПК, електроніка). Недоліком виступає складність автоматизованого виділення характеристик, якщо дані недостатньо структуровані.

Кластеризація характеристик (Unsupervised Learning) дозволяє автоматично групувати товари з подібними параметрами, усуваючи потребу ручного створення категорій або фільтрів (рис. 1.3). Кластеризація дає змогу виділяти технічно подібні групи товарів при цьому зменшувати розмір пошукового простору, підвищувати швидкість рекомендацій та уникати порівняння товарів, що належать до різних сегментів.

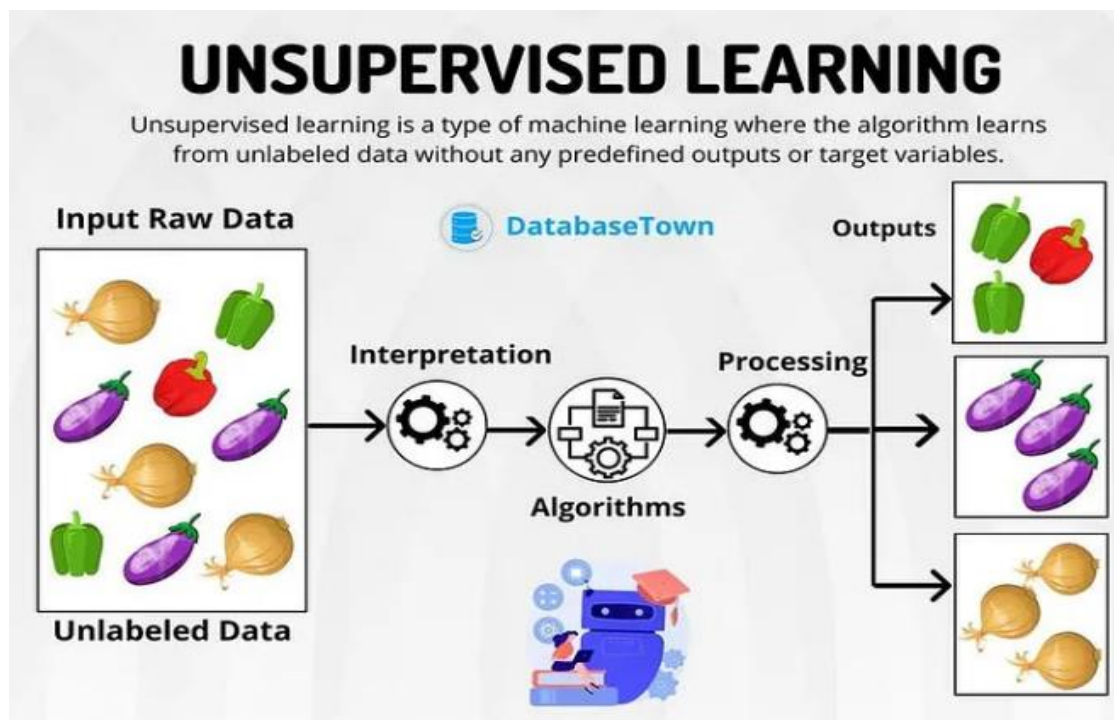


Рисунок 1.3 - Кластеризація характеристик

Такий підхід особливо ефективний при великій кількості моделей з мінімальними варіаціями (смартфони, відеокарти, роутери).

Модель вагових коефіцієнтів (Weighted Scoring Model) використовуються для багатокритеріального аналізу. Кожному параметру товару надається певна вага залежно від його важливості для користувача або системи.

Підсумковий рейтинг обчислюється як сума добутків параметра на його вагу. На початку реєструються опції (інновації, новизна) та вибираються критерії (вартість, якість) для яких призначається вага і оцінюємо в зважений бал для кожного варіанта оцінювання (рис.1.4). Цей підхід забезпечує прозорість

формування рекомендацій, можливість адаптації під різні сценарії підбору та гнучкість у зміні пріоритетів користувача. Вагова модель є критично важливою, оскільки дозволяє узагальнити різноманітні характеристики та формалізувати процес прийняття рішень.

Steps to Create Weighted Scoring Model for Product Development

The following slide highlights the steps to draft weighted scoring model for product development illustrating enlist options, select criteria, assign weights and weighted scores



Рисунок 1.4 - Кроки для створення зваженої моделі для розробки продукту

Для наглядної картини проведено порівняльну характеристику моделей (табл.1.1).

Таблиця 1.1 - Порівняльна характеристика моделей

Критерій порівняння	Контентно-орієнтована модель (Content-Based Filtering)	Кластеризація характеристик (Unsupervised Learning)	Модель вагових коефіцієнтів (Weighted Scoring Model)
Тип навчання	Навчання з ознаками користувача (без явного навчання моделі)	Навчання без учителя	Аналітична модель без навчання
Вхідні дані	Характеристики товарів та профіль користувача	Характеристики товарів	Характеристики товарів та задані ваги
Рівень персоналізації	Високий	Низький (опосередкований)	Середній-високий (через налаштування ваг)
Інтерпретованість результатів	Середня	Низька	Висока
Обчислювальна складність	Середня	Середня-висока	Низька
Масштабованість	Обмежена при великій кількості ознак	Висока	Висока
Адаптація до користувача	Через історію взаємодії	Відсутня без додаткових моделей	Через зміну вагових коефіцієнтів
Основне призначення	Персоналізований підбір схожих товарів	Структурування та групування даних	Фінальне ранжування та вибір
Недоліки	«Ефект обмеженості» рекомендацій	Складність інтерпретації	Залежність від коректності ваг

З метою підвищення точності, адаптивності та інтерпретованості рекомендацій запропоновано гібридну модель, що поєднує контентно-орієнтований підхід, методи кластеризації та модель вагових коефіцієнтів. Кластеризація структурує простір даних, контентно-орієнтована модель виконує персоналізацію вибору, модель вагових коефіцієнтів забезпечує ранжування результатів та формує пояснення [1; 2; 12].

Кожен із зазначених методів виконує окрему функціональну роль у загальній структурі системи та впливає на формування фінального результату.

1.3 Огляд програмних засобів розробки з інтелектуальними аналізом

Розробка вебплатформи підтримки вибору товарів вимагає застосування сукупності програмних інструментів, що забезпечують створення надійної, масштабованої та інтелектуальної системи. До програмних засобів належать технології для бекенда, фронтенда, баз даних, а також інструменти машинного навчання, що формують основу інтелектуального аналізу.

Серед серверних технологій найбільш поширеними є Node.js, Python (Django/Flask/FastAPI) та PHP (Laravel). Для систем, де потрібно обробляти значні масиви даних або реалізовувати модулі інтелектуального аналізу, доцільним є застосування Python, оскільки він має великий спектр бібліотек для машинного навчання: scikit-learn, TensorFlow, PyTorch, Pandas, NumPy. Саме завдяки ним можлива реалізація контент-орієнтованих та гібридних рекомендаційних систем, що були розглянуті в дослідженні.

Для клієнтської частини використовуються HTML5, CSS3, JavaScript, а також сучасні фреймворки - React, Vue.js або Angular. Завдяки компонентній структурі React можна побудувати інтерактивний інтерфейс, у якому відображатимуться рекомендації, товарні картки, порівняння характеристик та персоналізовані підказки.

Для збереження даних про товари, відгуки, рейтинги, а також профілі користувачів доцільно використовувати як реляційні (MySQL, PostgreSQL), так і документно-орієнтовані (MongoDB, Firestore). Вибір залежить від характеру даних. Для структурованих характеристик товарів підійдуть реляційні бази, у той час як для зберігання логів взаємодії користувача, історії кліків та персоналізованих моделей зручнішими є документно-орієнтовані рішення, які й були застосовані в роботі.

Важливу роль відіграють хмарні сервіси, зокрема AWS, Google Cloud або Azure. Вони надають можливість запуску модулів машинного навчання,

масштабування бази даних і використання розподіленого зберігання для обробки великих масивів інформації [3; 16; 17].

Таблиця 1.2 - Загальний огляд використання інструментів

Мета використання	Програмні інструменти / технології
Реалізація серверної логіки (бекенд)	Python (Django, Flask, FastAPI), Node.js, PHP (Laravel)
Обробка запитів та реалізація API	REST API, GraphQL, FastAPI
Реалізація алгоритмів машинного навчання	scikit-learn, TensorFlow, PyTorch
Аналіз та обробка даних	Pandas, NumPy
Реалізація рекомендаційних алгоритмів	scikit-learn, TensorFlow, PyTorch
Контентний аналіз та робота з текстовими описами	TF-IDF (scikit-learn), NLP-бібліотеки
Формування клієнтського інтерфейсу	HTML5, CSS3, JavaScript
Побудова інтерактивного UI	React, Vue.js, Angular
Зберігання структурованих даних про товари	MySQL, PostgreSQL
Зберігання напівструктурованих та поведінкових даних	MongoDB, Firestore
Зберігання логів і історії взаємодії користувачів	MongoDB, Firestore
Хмарне розгортання та масштабування	AWS, Google Cloud, Microsoft Azure
Масштабування та відмовостійкість	Хмарні сервіси (AWS, GCP), контейнеризація
Підтримка модулів машинного навчання в хмарі	AWS ML Services, Google AI Platform

Отже, програмні засоби розробки включають комплекс технологій для створення вебінтерфейсу, серверної логіки, системи аналітики та інтелектуальної обробки даних, що забезпечує адаптивність і персоналізацію вебплатформи.

1.4 Огляд апаратної частини при розробці рекомендаційного підбору

Апаратна частина розроблюваної вебплатформи визначаються необхідністю забезпечення стабільної роботи серверної інфраструктури, зберігання даних та обробки запитів користувачів у реальному часі.

Базовим елементом апаратної частини є сервер, на якому розміщується бекенд вебплатформи та модулі інтелектуального аналізу. Потужність серверного обладнання залежить від кількості користувачів, складності моделей машинного навчання та обсягів даних. У більшості випадків застосовуються сервери з багатоядерними процесорами (Intel Xeon, AMD EPYC), оперативною пам'яттю від 16 до 64 ГБ і SSD-накопичувачами для швидкого доступу до баз даних.

При використанні хмарних технологій апаратна складова делегується провайдерам, що дозволяє автоматично масштабувати ресурси. Це особливо важливо для систем рекомендацій, оскільки обробка подібності товарів, формування профілю користувача або кластеризація потребують значних обчислювальних потужностей.

Для високопродуктивних модулів, які містять нейронні мережі або використовують глибоке навчання, застосовуються GPU-сервери. Вони забезпечують прискорення операцій з великими матрицями та підвищують продуктивність при навчанні моделей [4; 9].

1.5 Особливості розробки платформи та дослідження

Метою даної роботи є розробка вебплатформи з інтелектуальною системою підтримки вибору товарів, яка забезпечує користувачеві доступ до персоналізованих рекомендацій, порівняння характеристик та визначення найбільш оптимальних варіантів відповідно до індивідуальних потреб.

Основне завдання полягає в тому, щоб об'єднати традиційні механізми пошуку товарів з методами інтелектуального аналізу даних. На основі дослідження аналогічних платформ та існуючих моделей рекомендацій було сформульовано необхідність застосування гібридних підходів. Система має забезпечувати адаптацію до користувача, аналізуючи його вподобання, поведінкові показники та контекст взаємодії.

До основних задач розробки веб-платформи належить формування цілісної архітектури системи, яка визначає взаємодію між її програмними компонентами та забезпечує масштабованість і надійність функціонування. У межах цього етапу здійснюється обґрунтований вибір програмних і апаратних засобів з урахуванням вимог до продуктивності, безпеки та можливості подальшого розширення. Важливою складовою розробки є створення бази даних товарів і їх характеристик, яка виступає основним джерелом інформації для аналітичних та інтелектуальних модулів платформи. Паралельно реалізуються механізми збору та збереження даних про взаємодію користувачів із системою, що дозволяє фіксувати поведінкові патерни та використовувати їх для персоналізації рекомендацій.

Наступним етапом є впровадження алгоритмів рекомендаційного характеру з використанням методів машинного навчання та елементів штучного інтелекту, які забезпечують автоматизований аналіз даних і формування релевантних пропозицій відповідно до індивідуальних вимог користувача. Особлива увага приділяється побудові користувацького інтерфейсу, орієнтованого на простоту, зрозумілість та інформативність, що сприяє зменшенню когнітивного навантаження та скороченню часу пошуку необхідних товарів.

Завершальним етапом розробки є тестування системи та оцінка ефективності реалізованих рекомендаційних моделей, що включає перевірку коректності роботи функціональних компонентів, аналіз точності та релевантності рекомендацій, а також загальну оцінку зручності використання вебплатформи. Отримані результати тестування використовуються для подальшої оптимізації алгоритмів і вдосконалення архітектури системи [4; 5].

У рамках дослідження виконується аналіз існуючих систем, релевантність результатів та зручність навігації, визначення їхніх обмежень і формування концепції веб-платформи, що забезпечує підвищення точності вибору товарів завдяки застосуванню методів штучного інтелекту й аналізу даних.

1.6 Висновок до розділу

У першому розділі було розглянуто основні аспекти, пов'язані з рекомендаційними системами, їх актуальністю, структурою, розвитком та впливом на користувачів. Рекомендаційні системи відіграють ключову роль у пошуку й відкритті релевантної інформації, оптимізації процесів вибору та підвищенні ефективності взаємодії з цифровими платформами. Актуальність таких систем особливо проявляється в умовах постійного зростання обсягів інформації та складності її обробки. Визначено, що такі системи функціонують на основі індивідуалізованого підходу, аналізуючи поведінку користувачів і пропонуючи рекомендації на основі різних методів. Основні етапи роботи рекомендаційних систем включають збір даних, аналіз та генерацію рекомендацій, що забезпечує їхню точність і релевантність. Встановлено, що рекомендаційні системи пройшли шлях від базових алгоритмів, заснованих на правилах, до сучасних технологій, що використовують потужність штучного інтелекту й машинного навчання. Цей прогрес став можливим завдяки розвитку обчислювальних потужностей і технологій обробки великих обсягів даних. Аналіз впливу рекомендаційних систем на користувачів засвідчив їх важливість у формуванні персоналізованого досвіду, підвищенні зручності використання цифрових платформ і формуванні довіри до брендів. Водночас рекомендаційні системи відкривають нові можливості для бізнесу, сприяючи утриманню клієнтів, підвищенню їхньої лояльності та зростанню продажів. Системи є потужним інструментом персоналізації, який дозволяє не лише полегшити процес вибору для користувачів, але й значно

покращити ключові показники ефективності бізнесу. Рекомендаційні системи мають величезний потенціал. Вони не тільки допомагають користувачам ефективніше знаходити потрібну інформацію, але й сприяють розвитку бізнесу, наукових досліджень і технологій. Завдяки їм ми можемо покращити якість життя в цифровому світі, зробивши його більш зручним, ефективним та індивідуалізованим. Перший розділ закладає основу для подальшого дослідження рекомендаційних систем, демонструючи їхню багатогранність, значення та потенціал для подальшого розвитку в умовах динамічного цифрового середовища.

2 ПРОЄКТУВАННЯ ПРОГРАМНОЇ ЧАСТИНИ СИСТЕМИ

Проєктування програмної частини вебплатформи підтримки вибору товарів з елементами інтелектуального аналізу спрямована на побудову комплексної технологічної системи, здатної обробляти великі масиви інформації та надавати користувачеві обґрунтовані рекомендації.

Програмна частина платформи включає в себе незалежність модулів, високу масштабованість та можливість розширення функціоналу. Модулі контентного аналізу, що забезпечують автоматизоване виділення структурованих характеристик товарів, модулі кластеризації, які автоматично групують товари за прихованими ознаками, а також системи персоналізації, що формують індивідуальні рекомендації, використовуючи поведінкові профілі користувачів. Відповідність програмних модулів вебплатформи та їх алгоритмів представлені у (табл. 2.1).

Таблиця 2.1 - Відповідність модулів до алгоритмів

Модуль системи	Основні алгоритми та методи
Модуль контентного аналізу	TF-IDF, cosine similarity, one-hot encoding, embeddings
Модуль кластеризації	K-Means, Hierarchical Clustering, DBSCAN
Модуль персоналізації	Content-Based Filtering, Hybrid Recommender, Neural CF
Модуль поведінкових профілів	User profiling, implicit feedback
Серверний API	REST, JSON, асинхронна обробка
Data Layer	SQL / NoSQL, кешування
UI Layer	Фільтрація, ранжування, відображення рекомендацій

Усі ці компоненти інтегруються через серверне середовище та API, що забезпечує узгоджену роботу всієї платформи [1; 5; 12].

2.1 Модуль контентного аналізу та інтелектуального підбору

Даний модуль є ядром інтелектуальної частини вебплатформи та виконує ключову роль у процесі автоматизованого підбору товарів відповідно до вимог користувача. Основним завданням модуля є аналіз характеристик товарів, виділення найбільш значущих параметрів та формування рейтингу релевантності, що відображає ступінь відповідності кожної позиції заданим критеріям. Функціонування модуля розпочинається з автоматичного розбору технічних параметрів, під час якого здійснюється витягнення характеристик із структурованих і напівструктурованих джерел, зокрема JSON- та HTML-описів. Отримані дані проходять етап нормалізації, що забезпечує приведення числових параметрів, таких як потужність, об'єм або діапазони значень, до єдиного

масштабу, а текстові атрибути перетворюються у структурований формат, придатний для подальшого аналізу.

Основою методу є обчислення схожості між товарами за їх атрибутами (технічними параметрами, типом, функціональністю, форм-факторами). Для цього використовують алгоритми:

- TF-IDF для текстових описів;
- cosine Similarity для обчислення подібності векторів характеристик;
- one-hot. Embedding моделі для категоріальних параметрів.

На основі підготовлених даних реалізується контентно-орієнтована фільтрація, яка передбачає визначення схожості між товарами шляхом векторизації їхніх характеристик та розрахунку відповідних параметрів. Такий підхід дозволяє враховувати не лише наявність окремих ознак, але й їх відносну важливість у контексті запиту користувача, що забезпечує більш точне формування переліку товарів, найбільш наближених до заданих вимог. Паралельно з цим модуль виконує попереднє виключення нерелевантних позицій, здійснюючи фільтрацію товарів, які не відповідають обмеженням за бюджетом, сумісністю або функціональним призначенням. Завдяки такому відсіканню «зайвих» варіантів істотно скорочується пошуковий простір і зменшується час обробки запиту.

У результаті застосування описаного підходу модуль забезпечує формування результатів пошуку з підвищеною точністю та релевантністю порівняно з традиційними механізмами фільтрації, що базуються лише на жорстких умовах або статичних параметрах. Запропоноване рішення дозволяє поєднати автоматизований аналіз даних, персоналізований підхід до користувача та ефективну оптимізацію процесу пошуку в межах єдиної інтелектуальної компоненти вебплатформи [6; 8].

2.2 Модуль кластеризації та персоналізації користувача

Даний модуль відповідає за адаптацію вебплатформи до індивідуальних потреб користувача та забезпечує персоналізацію процесу підбору товарів. У результаті автоматичного групування товари об'єднуються за призначенням, ключовими характеристиками та рівнем продуктивності, що дає змогу формувати логічні сегменти, орієнтовані на різні категорії користувачів, зокрема рішення базового, оптимального та професійного рівнів. Така структура спрощує навігацію в межах каталогу та створює основу для більш точного подальшого аналізу.

Паралельно з цим модуль здійснює формування динамічного профілю користувача на основі аналізу поведінкових даних, до яких належать історія пошукових запитів, кліків, а також час перебування на сторінках товарів. Отримана інформація використовується для визначення параметрів, що мають найбільшу значущість для конкретного користувача, та дозволяє автоматично коригувати вагові коефіцієнти, які застосовуються в механізмах оцінювання та ранжування. Такий підхід забезпечує поступову адаптацію системи до змін у вподобаннях користувача без необхідності його прямого втручання.

На основі сформованого профілю та результатів кластеризації реалізується механізм персоналізованих рекомендацій, що враховує поведінкові патерни та індивідуальні уподобання. Модель підбору товарів оновлюється в процесі накопичення нових даних, що дозволяє виконувати повторне навчання та підвищувати точність рекомендацій з часом. У результаті застосування даного модуля рекомендації стають більш релевантними, а пошуковий шлях користувача скорочується, що позитивно впливає на зручність взаємодії з вебплатформою та загальну ефективність її використання [1; 10; 13].

2.3 Модуль персоналізації та інтелектуального формування і його взаємодія

Одним із ключових функціональних компонентів розроблюваної вебплатформи є модуль персоналізації, призначений для формування індивідуальних рекомендацій товарів з урахуванням поведінкових характеристик користувачів. Основною метою цього модуля є підвищення релевантності результатів підбору товарів шляхом адаптації рекомендацій до індивідуальних уподобань, інтересів та історії взаємодії користувача з платформою.

Основою роботи модуля персоналізації є побудова поведінкових профілів користувачів, які формуються на основі аналізу таких даних, як історія переглядів товарів, пошукові запити, кліки, фільтрація за характеристиками, тривалість сесій, а також результати попередніх рекомендацій. Отримані дані структуруються у вигляді векторів ознак, що дозволяє використовувати методи інтелектуального аналізу для подальшої обробки.

Для виявлення прихованих закономірностей у поведінці користувачів та сегментації аудиторії в системі застосовується модуль кластеризації. Він дозволяє групувати користувачів за схожими поведінковими ознаками, формуючи кластери з подібними інтересами та моделями взаємодії з платформою. Для цього можуть використовуватися алгоритми k-means, DBSCAN або ієрархічної кластеризації залежно від характеру даних та вимог до масштабованості. Результати кластеризації використовуються для уточнення рекомендацій шляхом врахування переваг не лише окремого користувача, але й його кластера.

Паралельно з цим у системі реалізовано модуль контентного аналізу, який відповідає за аналіз характеристик самих товарів. У межах цього модуля обробляються структуровані та напівструктуровані дані, зокрема технічні параметри, категорії, описи, теги та рейтинги товарів. Контентний аналіз дозволяє визначати ступінь подібності між товарами та формувати рекомендації на основі їх властивостей незалежно від активності інших користувачів.

Інтеграція модулів персоналізації, кластеризації та контентного аналізу забезпечує гібридний підхід до формування рекомендацій. Остаточний список рекомендованих товарів формується шляхом комбінування результатів індивідуального поведінкового аналізу, групових уподобань кластера та контентної подібності товарів. Такий підхід дозволяє зменшити проблему «холодного старту», підвищити точність рекомендацій та забезпечити адаптивність системи до змін у поведінці користувачів [3; 4; 13; 18].

Таким чином, модуль персоналізації виступає центральним елементом інтелектуальної складової вебплатформи та забезпечує ефективну підтримку вибору товарів шляхом комплексного аналізу користувацьких і контентних даних.

2.4 Функціонування та взаємодія модулів, їх реалізація

Функціонування зазначених модулів має тісно інтегрований характер і ґрунтується на послідовному обміні результатами аналізу. Взаємодія між цими модулями організована таким чином, що кожен з них виконує окрему аналітичну функцію, водночас доповнюючи та уточнюючи результати іншого, що забезпечує комплексний підхід до формування рекомендацій.

На початковому етапі аналізу модуль контентного аналізу здійснює обробку вхідних даних про товари, перетворюючи їх із неструктурованих або напівструктурованих описів у формалізований набір характеристик. Отримані структуровані ознаки формують єдиний інформаційний простір, який використовується разом з модулем кластеризації як спільна база для подальших обчислень. Таким чином, якість та повнота контентного аналізу безпосередньо впливають на ефективність кластеризації та персоналізації, оскільки саме на основі цих даних визначається подібність товарів і релевантність рекомендацій.

На наступному етапі модуль кластеризації використовує сформований простір характеристик для групування товарів за ступенем їх схожості. Результати

кластеризації не є кінцевими рекомендаціями, а виконують роль аналітичного фільтра, який структурує дані та зменшує обсяг альтернатив, що підлягають подальшому аналізу. Завдяки цьому модуль контентного підбору отримує вже впорядкований набір товарів, у межах якого може виконувати більш точне порівняння з профілем користувача. Таким чином, кластеризація опосередковано підвищує точність контентної фільтрації, зменшуючи вплив нерелевантних варіантів.

Результати контентно-орієнтованого аналізу використовуються модулем персоналізації для уточнення профілю користувача. Інформація про те, які характеристики товарів виявилися найбільш релевантними, дозволяє коригувати вагові коефіцієнти параметрів і адаптувати модель до індивідуальних уподобань. Таким чином, відбувається зворотний зв'язок між модулями, за якого результати підбору впливають на подальшу логіку персоналізації.

Додатково важливим аспектом взаємодії модулів є централізоване керування потоками даних та узгодження результатів аналізу на рівні серверної логіки. Сервер додатка виконує роль координуючого елемента, який забезпечує послідовність виклику модулів, контроль цілісності даних та кешування проміжних результатів обчислень. Такий підхід дозволяє зменшити обчислювальне навантаження, уникнути дублювання операцій та підвищити загальну швидкодію системи, що є критично важливим для роботи платформи в режимі реального часу.

Окрему роль у забезпеченні ефективної взаємодії модулів відіграє механізм адаптації моделей на основі накопичених даних. У процесі експлуатації платформи поведінкові дані користувачів (перегляди, вибори, час взаємодії) поступово оновлюють профілі персоналізації та можуть використовуватися для періодичного перенавчання моделей контентного аналізу і кластеризації. Це дозволяє системі динамічно реагувати на зміну уподобань користувачів і структури каталогу, підтримуючи актуальність рекомендацій без необхідності ручного втручання.

Аналіз у системі відбувається ітеративно, тобто з кожною новою взаємодією користувача результати кластеризації та контентного аналізу уточнюються, а профіль користувача оновлюється. Це дозволяє поступово зміщувати акцент

рекомендацій у бік тих характеристик, які мають найбільше значення для конкретного користувача, не порушуючи при цьому загальної структури даних. У підсумку модуль контентного аналізу забезпечує точне зіставлення товарів і вимог, тоді як модуль кластеризації та персоналізації відповідає за адаптацію цього зіставлення до поведінкових особливостей користувача [1; 4].

Таким чином, взаємодія модулів реалізує єдиний аналітичний цикл, у якому структуризація даних, інтелектуальний підбір і персоналізація не існують ізольовано, а формують узгоджений механізм прийняття рішень. Такий підхід дозволяє підвищити релевантність рекомендацій, скоротити пошуковий простір і забезпечити гнучку адаптацію системи до змін у вподобаннях користувача [13].

Для візуальної структури критеріїв побудована таблиця порівнянь (табл 2.1).

Таблиця 2.1 - Порівняльна характеристика модулів інтелектуальної вебплатформи

Модуль	Основне призначення	Вхідні дані	Вихідні результати	Методи та алгоритми
Модуль кластеризації	Групування користувачів або товарів за подібністю	Поведінкові дані, характеристики товарів	Кластери користувачів / товарів	k-means, DBSCAN, ієрархічна кластеризація
Модуль контентного аналізу	Аналіз властивостей товарів	Опис товарів, атрибути, метадані	Вектор ознак товару	TF-IDF, cosine similarity, embedding
Модуль інтелектуального підбору	Формування рекомендацій	Дані кластеризації + контентні ознаки	Список рекомендованих товарів	Гібридні рекомендаційні моделі
Модуль персоналізації	Адаптація рекомендацій під користувача	Історія дій користувача	Персоналізований рейтинг товарів	Вагові коефіцієнти, профілі користувачів

2.5 Математична структура гібридної моделі інтелектуального підбору товарів

Математична структура є формалізованою основою для проведення експериментального дослідження, аналізу релевантності роботи системи для відображення взаємозв'язків між елементами параметрів. Запропонована структура має гібридний характер і поєднує контент-орієнтоване подання товарів, кластеризацію та персоналізацію на основі профілю користувача. Вона забезпечує єдиний підхід до кількісної оцінки релевантності товарів, що дає змогу коректно порівнювати ефективність різних вебплатформ пошуку та підбору товарів у межах експерименту.

Базові складові гібридної моделі та контент-орієнтоване подання товару задані як - процес формування узагальненого векторного подання товару на основі різномірних джерел інформації:

$$\vec{v}_i = [w_{text} \times TF-IDF(d_i), w_{attr} \times \vec{a}_i], \quad (2.1)$$

$$w_{text} + w_{attr} = 1$$

де $\vec{v}_i$ - контентний вектор товару;

i - товар;

d_i - текстовий опис товару;

$\vec{a}_i$ - нормалізований вектор атрибутів;

w_k - вагові коефіцієнти складових моделей.

Текстовий опис товару ($\vec{d}_i$) містить назву, опис, технічні характеристики та інші текстові атрибути. Основна функція $TF-IDF(\vec{d}_i)$ перетворює цей текст у числовий вектор, що відображає важливість термів усієї колекції товарів та дозволяє врахувати семантичну схожість між ними.

Вектор $\vec{a}_i$ відповідає за структуровані числові характеристики товару, такі як технічні параметри, категорії, форм-фактор або інші формалізовані ознаки. Перед об'єднанням ці характеристики проходять нормалізацію, що забезпечує коректне порівняння між різними товарами.

Коефіцієнти w є ваговими множниками, які визначають відносний внесок текстових та атрибутивних ознак у загальний вектор товару. Вони дозволяють адаптувати модель до специфіки предметної області, підсилюючи або зменшуючи вплив окремих типів даних.

Операція об'єднання в квадратних дужках означає конкатенацію векторів, тобто формування єдиного векторного простору ознак без їх прямого арифметичного додавання. Такий підхід дозволяє коректно поєднувати різновимірні та семантично різні характеристики, що є критично важливим для подальших етапів кластеризації, персоналізації та розрахунку схожості товарів.

На другому рівні гібридної моделі застосовується процедура кластеризації товарів, яка полягає у групуванні об'єктів на основі їх векторних представлень $\vec{v}_i$ сформованих на попередньому етапі контентного аналізу. Формально кожному товару ставиться у відповідність кластер $\vec{C}_j$, до якого він належить, відповідно до результатів роботи алгоритму кластеризації. Таким чином, товари з подібними характеристиками автоматично об'єднуються в однорідні групи без попереднього задання класів:

$$C_j = \text{Cluster}(\vec{v}_i), i \in C_j, \quad (2.2)$$

де C_j - кластер, до якого належить товар i ;

$\vec{v}_i$ - контентний вектор товару;

i - товар.

Кластеризація виконує роль структурного фільтра, оскільки дозволяє суттєво зменшити простір пошуку релевантних товарів на наступних етапах обробки. Замість порівняння користувацького профілю з усім каталогом система

обмежується лише тими кластерами, які найбільше відповідають інтересам користувача. Це не лише підвищує обчислювальну ефективність, але й зменшує вплив нерелевантних альтернатив, що позитивно позначається на точності та стабільності рекомендацій. Крім того, результати кластеризації слугують проміжним рівнем узагальнення даних, що сприяє більш стійкій роботі моделі в умовах великого обсягу товарів та динамічних змін каталогу.

На третьому рівні гібридної моделі реалізується формування персоналізованого профілю користувача, який відображає його індивідуальні вподобання та поведінкові характеристики. Профіль користувача будується на основі історії взаємодій із товарами, зокрема переглядів, виборів, додавання до списків бажань або покупок. Кожна взаємодія враховується з відповідним ваговим коефіцієнтом, що дозволяє диференціювати вплив різних типів дій:

$$\vec{p}_u = \frac{1}{\sum_{i \in I_u} a_i} \sum_{i \in I_u} a_i \times \vec{v}_i, \quad (2.3)$$

- де $\vec{p}_u$ - персоналізований профіль користувача;

u - користувач;

a_i - множина взаємодій користувача;

I_u - вага типу взаємодії.

Отриманий профіль представляється у вигляді вектору ознак, сумісного з контентними векторами товарів, що забезпечує можливість подальшого обчислення схожості та формування релевантних рекомендацій.

Четвертий рівень гібридної моделі спрямований на кількісну оцінку релевантності кожного товару для конкретного користувача. На цьому етапі обчислюються часткові показники відповідності, зокрема контентна схожість між профілем користувача та товаром, належність товару до релевантного кластеру, а також відповідність індивідуальним параметрам та обмеженням користувача.

Контента схожість:

$$S_{const}(u, i) = \cos(\vec{p}_i, \vec{v}_i), \quad (2.4)$$

Кластерна релевантність:

$$S_{clust}(u, i) = \begin{cases} 1, & i = C_u \\ 0, & i \neq C_u \end{cases}, \quad (2.5)$$

Персоналізована відповідність:

$$S_{const}(u, i) = f(persfs_u, \vec{a}_i), \quad (2.6)$$

Кожен з цих компонентів відображає окремий аспект взаємодії користувача з товарним простором і використовується для формування узагальненої оцінки. Такий багаторівневий підхід дозволяє уникнути переваги одного джерела інформації та підвищити стійкість рекомендаційної системи.

Узагальнена формула гібридної моделі об'єднує результати контентного аналізу, кластеризації та персоналізації в єдиний числовий показник релевантності $R(u, i)$, який визначає доцільність рекомендації товару i користувачу u . Формула базується на методі зваженої агрегації, де кожна складова моделі має власний коефіцієнт впливу:

$$R(u, i) = w_1 \times \cos(\vec{p}_u \vec{v}_i) + w_2 \times S_{clust}(u, i) + w_3 \times S_{pers}(u, i), \quad (2.7)$$

де w_k - вагові коефіцієнти складових моделей.

Це дозволяє гнучко налаштовувати систему відповідно до особливостей предметної області, типу товарів або поведінки користувачів, змінюючи відносну важливість окремих компонентів.

Використання вагових коефіцієнтів у гібридній формулі забезпечує адаптивність системи та можливість її подальшого навчання на основі зворотного

зв'язку. Значення ваг можуть коригуватися автоматично в процесі експлуатації платформи або задаватися експертно на етапі налаштування. Завдяки цьому модель здатна реагувати на зміну вподобань користувачів, сезонні коливання попиту або розширення каталогу товарів. Узагальнена формула слугує центральним елементом рекомендаційного механізму, забезпечуючи баланс між точністю, обчислювальною ефективністю та інтерпретованістю результатів, що є критично важливим для практичного використання інтелектуальної вебплатформи.

2.6 Висновки до розділу

У даному розділі було розглянуто проектування програмної частини вебплатформи підтримки вибору товарів з елементами інтелектуального аналізу. Основну увагу приділено побудові модульної архітектури системи, яка забезпечує гнучкість, масштабованість та можливість подальшого розширення функціоналу без суттєвих змін у загальній структурі платформи.

У процесі роботи було проаналізовано та спроектовано модуль контентного аналізу та інтелектуального підбору, який відповідає за автоматизоване опрацювання описів товарів, виділення ключових характеристик і формування структурованих ознак, необхідних для подальшого аналізу. Даний модуль створює основу для коректної роботи алгоритмів машинного навчання та рекомендаційних механізмів.

Також було розглянуто модуль кластеризації та персоналізації користувача, що дозволяє групувати товари та користувачів за прихованими закономірностями й поведінковими патернами. Застосування методів кластеризації дає змогу зменшити розмірність даних, підвищити ефективність обробки інформації та покращити якість персоналізованих рекомендацій.

Окрему увагу приділено модулю персоналізації та інтелектуального формування рекомендацій, який інтегрує результати контентного аналізу та кластеризації з поведінковими профілями користувачів. Взаємодія цього модуля з іншими компонентами системи забезпечує адаптивність рекомендацій, їх актуальність та здатність враховувати зміни у вподобаннях користувачів і характеристиках товарів.

Детально наведено математичну структуру узагальненої гібридної моделі, яка формалізує процес інтелектуального підбору товарів шляхом поєднання контентно-орієнтованого аналізу, кластеризації та персоналізації користувача в єдину систему обчислення релевантності. Запропонована формалізація дозволяє наочно представити взаємодію окремих модулів платформи, обґрунтувати використання вагових коефіцієнтів для балансування впливу різних факторів та забезпечити адаптивність рекомендаційної системи до змін поведінки користувачів і структури товарного каталогу. Наявність узагальненої математичної моделі підвищує прозорість алгоритмів, спрощує їх подальшу модифікацію та створює основу для експериментальної перевірки ефективності запропонованих рішень.

У підсумку було описано функціонування та взаємодію модулів, а також принципи їх реалізації через серверне середовище та API. Запропонована архітектура забезпечує узгоджену роботу всіх компонентів системи, сприяє підвищенню продуктивності платформи та створює надійне підґрунтя для впровадження інтелектуальних методів аналізу даних у процесі вибору товарів.

3 РЕАЛІЗАЦІЯ ЗАГАЛЬНОЇ СТРУКТУРИ ПРОГРАМИ

Загальна структура розробленої вебплатформи побудована за трирівневим принципом - Data Layer, Logic Layer та UI Layer (рис. 3.1), що забезпечує модульність, гнучкість, масштабованість і можливість незалежного розвитку кожного компонента. Такий підхід дозволяє системі залишатися стабільною навіть

за умови розширення каталогів, зміни моделей машинного навчання або збільшення кількості користувачів.

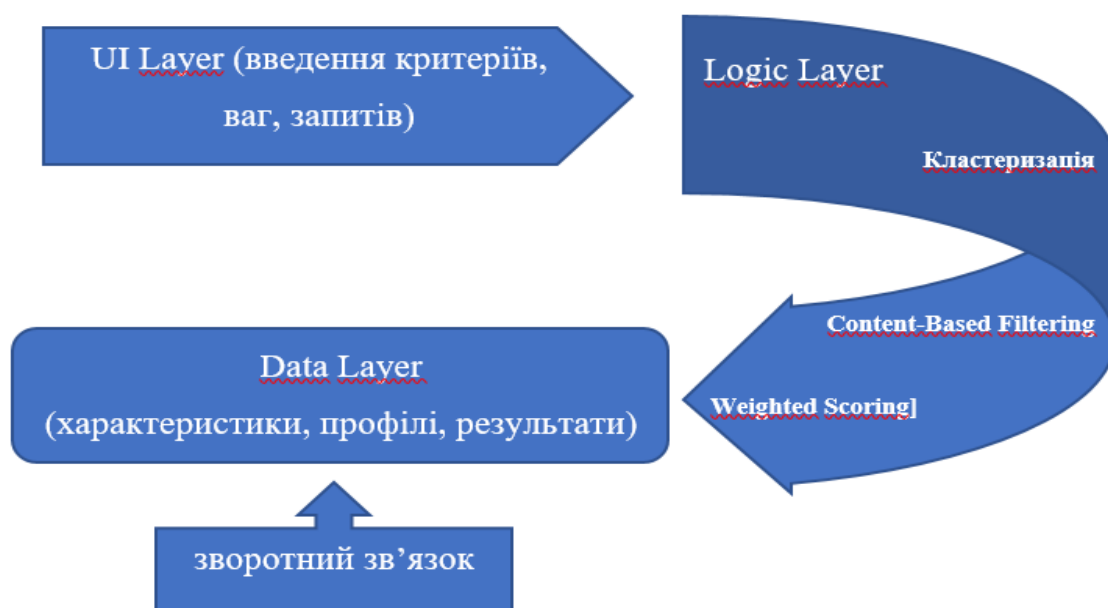


Рисунок 3.1 - Загальна структура платформи

На рівні даних (Data Layer) використано документно-орієнтоване сховище MongoDB, у якому зберігаються структуровані набори інформації: товарні дані, їхні параметри, метрики, вагові коефіцієнти, а також історії взаємодії користувачів. Структура колекцій організована за типами товарів та їх характеристиками, що дозволяє виконувати гнучкі запити без складних схем нормалізації. Для прискорення пошуку створено індекси за ключовими параметрами, що позитивно впливає на швидкодію платформи та підвищує ефективність обробки запитів.

Логічний рівень (Logic Layer) реалізує бізнес-логіку системи та виконує основну обчислювальну роботу. До нього віднесено модуль аналізу та обробки товарного контенту, який відповідає за вилучення характеристик та формування структурованих описів; модуль кластеризації, що реалізує групування товарів на основі їх подібності та визначення латентних зв'язків; модуль персоналізації, який відстежує поведінкові патерни, аналізує історію кліків та формує динамічний профіль користувача. Окрім цього, логічний рівень включає REST API -

універсальний інтерфейс взаємодії між сервером та клієнтом, через який UI отримує результати пошуку, рекомендації та порівняльні дані.

Інтерфейсний рівень (UI Layer) забезпечує взаємодію користувача з платформою через вебклієнт, побудований на HTML/CSS/JS (ДОДАТОК А). Інтерфейс орієнтований на зручність взаємодії: він включає розширену систему фільтрації, модуль порівняння товарів, інтерактивні товарні картки з рекомендаційними вставками та адаптивний дизайн, що дозволяє використовувати платформу на мобільних пристроях без втрати функціональності чи читабельності.

Загальна архітектура розробленої вебплатформи побудована за трирівневим принципом - Data Layer, Logic Layer та UI Layer, що забезпечує модульність, гнучкість, масштабованість і можливість незалежного розвитку кожного компонента. Такий підхід дозволяє системі залишатися стабільною навіть за умови розширення каталогів, модифікації моделей машинного навчання або збільшення кількості користувачів.

У межах цієї архітектури реалізовано гібридний механізм інтелектуального підбору, що поєднує контентно-орієнтовану модель, методи кластеризації та модель вагових коефіцієнтів. Кожен із підходів інтегрований у відповідний рівень системи та виконує чітко визначену функцію.

Для зручності узагальнення, структура платформи представлена в (табл. 3.1).

Таблиця 3.1 - Архітектура вебплатформи

Архітектурний рівень	Основні функції	Ключові характеристики
Data Layer	Збереження товарів, параметрів, користувацьких даних; ведення історії взаємодії	MongoDB; документно-орієнтована модель; індексація для прискорення пошуку; колекції, організовані за типами товарів
Logic Layer	Контентний аналіз, кластеризація, побудова рекомендацій, персоналізація; серверне REST API	Модулі ML-обробки; формування профілів користувачів; обчислення вагових коефіцієнтів та груп; забезпечення логіки пошуку
UI Layer	Візуалізація товарів, взаємодія з фільтрами, відображення рекомендацій, порівняння товарів	Вебінтерфейс на HTML/CSS/JS або React; адаптивний дизайн; інтерактивні картки товарів; мінімалістичний UI

Така інтеграція дозволяє поєднати переваги кожного підходу та мінімізувати їхні окремі недоліки, забезпечуючи водночас точність рекомендацій, масштабованість системи та інтерпретованість результатів для кінцевого користувача. На початковому етапі функціонування системи здійснюється збір та попередня обробка даних про товари, що включає нормалізацію числових параметрів, уніфікацію категоріальних ознак та формування векторного подання характеристик. Саме на цьому етапі закладається основа для подальшої коректної роботи всіх компонентів моделі, оскільки якість вхідних даних безпосередньо впливає на результати аналізу та рекомендацій.

Після формування єдиного простору ознак виконується кластеризація товарів із використанням методів навчання без учителя, що дозволяє автоматично згрупувати об'єкти за ступенем подібності їх характеристик. Отримані кластери відображають приховану структуру даних та виконують роль семантичних груп, у межах яких товари мають спільні властивості. Це рішення дає змогу суттєво зменшити розмір пошукового простору, підвищити продуктивність системи та

створити основу для більш точного подальшого аналізу. Водночас результати кластеризації не є кінцевими рекомендаціями, а використовуються як проміжний аналітичний шар, що передає інформацію до наступних компонентів системи.

На основі визначених кластерів реалізується контентно-орієнтований аналіз, спрямований на врахування індивідуальних вимог та уподобань користувача. У межах релевантного кластера здійснюється порівняння профілю користувача з характеристиками товарів, що дозволяє визначити ступінь відповідності кожного об'єкта заданим критеріям. Профіль користувача формується динамічно та може враховувати як явно задані параметри, так і результати попередніх взаємодій із системою. Такий підхід забезпечує персоналізацію рекомендацій і дозволяє адаптувати їх до конкретних потреб, водночас уникаючи проблеми надмірної універсальності, характерної для простих алгоритмів підбору.

Фінальний етап обробки даних реалізується за допомогою моделі вагових коефіцієнтів, яка виконує функцію інтеграції результатів попередніх етапів та формує остаточний рейтинг товарів. Кожен об'єкт оцінюється на основі зваженої суми його параметрів, де ваги відображають відносну важливість окремих критеріїв для користувача. Застосування даного підходу забезпечує прозорість логіки прийняття рішень, оскільки кінцевий результат може бути пояснений через вплив конкретних характеристик та їхніх вагових коефіцієнтів. Крім того, модель дозволяє гнучко змінювати пріоритети без необхідності повного перенавчання системи, що є важливою перевагою з точки зору практичної експлуатації [5; 7; 15].

В результаті, взаємодія між компонентами гібридної моделі має послідовний і взаємодоповнюючий характер: кластеризація формує структуроване середовище для аналізу, контентно-орієнтований підхід забезпечує персоналізацію в межах цього середовища, а модель вагових коефіцієнтів агрегує отримані результати та перетворює їх у зрозумілий і керований механізм прийняття рішень.

3.1 Розробка практичної структурної схеми платформи

Структурна схема веб-платформи є фундаментальною моделлю, що визначає логічну організацію її програмних та апаратних компонентів, а також взаємодію між ними. Вона включає кілька функціональних блоків, кожен із яких виконує окрему групу завдань, але при цьому тісно пов'язаний з іншими елементами архітектури. Така модульність дозволяє забезпечити масштабованість, надійність та стійкість системи до зростання навантаження, а також її адаптивність до майбутніх оновлень чи розширення функціональності.

Першим ключовим елементом є серверна інфраструктура, яка становить основу платформи та відповідає за виконання бізнес-логіки, обробку вхідних запитів, керування даними та проведення аналітичних операцій. У складі цієї інфраструктури центральним компонентом виступає сервер додатка. Він реалізує бекенд-функціональність, включаючи маршрутизацію запитів, валідацію параметрів, виконання алгоритмів порівняння, формування рекомендацій та інтеграцію зі сторонніми сервісами. Сервер додатка працює у середовищі Linux або у хмарних контейнерах, що дозволяє забезпечувати високу стабільність, автоматичне оновлення, гнучкий розподіл ресурсів та горизонтальне масштабування у разі підвищеного навантаження.

Другим критичним компонентом цієї інфраструктури є база даних, яка забезпечує централізоване зберігання структурованої інформації про товари, їх параметри, категорії, рейтинги та метадані. Була використана документно-орієнтована система (наприклад, MongoDB) для забезпечення гнучкості, високої швидкості пошуку та легкого масштабування. База даних інтегрована із сервером додатка через стандартизовані драйвери та ORM-рівень, що сприяє оптимізованому виконанню запитів і зручному керуванню сутностями.

Третім компонентом серверної інфраструктури виступає сховище медіафайлів, у якому зберігаються зображення товарів, додаткові матеріали, проміжні результати обчислень або інші великі файли, що не можуть ефективно

зберігатися у структурованому вигляді бази даних. Таке сховище може бути реалізоване у вигляді локального файлового сервера або ж за допомогою хмарних рішень, таких як Amazon S3 чи Google Cloud Storage, що дозволяє забезпечити високу доступність, географічне дублювання та швидку доставку даних.

За потреби система може включати окремий GPU-сервер, який використовується для навчання моделей машинного навчання або виконання ресурсоємних операцій. Наявність графічного прискорювача дає змогу значно підвищити швидкість опрацювання великих масивів характеристик товарів, покращити точність алгоритмів ранжування та скоротити час, потрібний для оновлення моделей рекомендацій. GPU-вузол є опціональним, але його інтеграція дозволяє платформі підтримувати сучасні ML-підходи та забезпечувати кращу масштабованість інтелектуальних функцій у перспективі.

Другим великим блоком структурної схеми є комунікаційний модуль, який визначає методи та протоколи взаємодії між елементами системи. Основу його роботи становлять стандартизовані HTTP(S)-запити, які забезпечують безпечну та захищену передачу даних на всіх етапах. Взаємодія між клієнтською частиною та сервером реалізується через REST-інтерфейси, де кожен виклик відповідає окремому ресурсу або функції. Передача структурованої інформації здійснюється у форматі JSON, що робить дані легкими для обробки, сумісними з більшістю мов програмування та придатними для швидкого серіалізованого обміну. Комунікаційний блок також включає API для роботи з товарами, характеристиками та рекомендаційними результатами, що забезпечує гнучкість інтеграції з зовнішніми сервісами або мобільними застосунками.

Наступним компонентом структури є користувацький доступ, який забезпечує взаємодію кінцевих користувачів із платформою. Він включає вебклієнт - основний інтерфейс, доступний через браузер, що дозволяє здійснювати пошук товарів, їх порівняння, фільтрацію за характеристиками та перегляд рекомендацій. Мобільна версія платформи адаптована для смартфонів і планшетів, забезпечуючи зручність використання у будь-яких умовах та підтримуючи адаптивний дизайн.

Таким чином, побудована схема (рис.3.2) забезпечує повну логічну та апаратну цілісність веб-платформи.

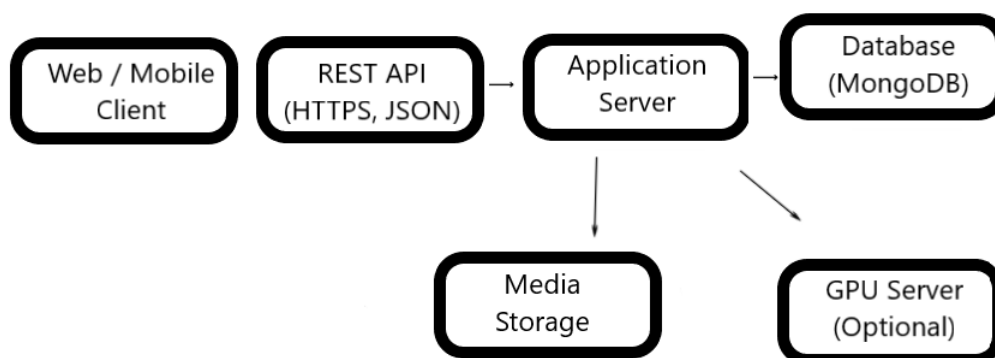


Рисунок 3.2 - Структурний вид платформи

Це дозволяє чітко відокремити вузли, оптимізувати їх взаємодію, підвищити продуктивність і створити масштабовану систему, придатну до подальшого розвитку. Така архітектура не лише сприяє стабільній роботі в умовах зростаючого навантаження, але й забезпечує можливість гнучкої модернізації відповідно до майбутніх потреб і технологічних змін [3; 9; 12].

3.2 Розробка функціонального серверного API-вузла

Окремим ключовим компонентом архітектури розробленої веб-платформи є функціональний вузол, оскільки саме він забезпечує оброблення бізнес-логіки, виконання обчислювальних операцій, підтримку інтеграцій та централізований доступ до даних. Його основна роль полягає у формуванні стабільного та масштабованого середовища, яке здатне обробляти значний потік запитів від користувацького інтерфейсу, виконувати алгоритмічні операції машинного навчання і гарантувати високу надійність під час взаємодії з базою даних і зовнішніми сервісами.

Основна конфігурація передбачає використання Linux-орієнтованих дистрибутивів, переважно Ubuntu Server, що є стабільним, безпечним, масштабованим та широко підтримуваним середовищем для розгортання серверних застосунків. У якості альтернативи передбачена можливість розгортання у хмарній інфраструктурі AWS або Google Cloud, що дозволяє гнучко масштабувати ресурси, автоматизувати оновлення, застосовувати балансування навантажень та отримувати доступ до керованих сервісів зберігання даних, телеметрії та логування. Використання хмарних інстансів також забезпечує вищу відмовостійкість системи, оскільки унеможливорює зупинку сервісу через апаратний збій.

Функціонально серверний вузол виконує кілька ключових завдань. Одне з центральних - операційна підтримка бекенд-програми, що реалізує API платформи, модулі керування сесіями, механізми авторизації та аутентифікації, а також інтерфейси для роботи із збереженими даними. Взаємодія з клієнтською частиною відбувається через REST- або GraphQL-інтерфейси, які забезпечують стандартизований обмін даними та чітке розмежування функцій між компонентами системи. Сервер приймає, обробляє та маршрутизує запити, проводячи валідацію введених параметрів і оптимізуючи виконання складних операцій.

Окремий блок роботи серверного вузла стосується виконання вбудованих алгоритмів машинного навчання, які забезпечують функціонал рекомендаційної системи та інтелектуального порівняння товарів. Серед таких алгоритмів можуть бути моделі класифікації, ранжування, пошуку найближчих сусідів або контент-орієнтовані методи зіставлення характеристик. Сервер відповідає за підготовку даних для моделей, запуск обчислювальних процесів та повернення користувачу структурованих результатів аналізу. У разі використання великих моделей або потреби в інтенсивних матричних обчисленнях передбачена можливість підключення графічного процесора (GPU), що забезпечує значне прискорення обчислень.

Ще одна важлива функція серверного вузла - взаємодія з базою даних. Саме через сервер здійснюється централізований доступ до товарів, їхніх характеристик,

метаданих, результатів порівняння та історії взаємодій користувачів із системою. Тут реалізовано модулі для ефективної роботи з даними: кешування часто використовуваних запитів, оптимізація індексів, механізми транзакційного контролю, а також захист від некоректних або дубльованих операцій. База даних може бути як реляційною, так і документно-орієнтованою - залежно від специфіки збережених структур, обсягів даних та вибраної архітектури системи. В обох випадках серверний застосунок виконує роль посередника, що забезпечує коректну обробку запитів і стабільний доступ до даних.

Не менш важливим компонентом є система логування та телеметрії. Сервер відповідає за фіксацію системних подій, помилок, а також статистики використання, необхідної для аналізу роботи платформи та подальшого її вдосконалення. Зокрема, здійснюється моніторинг часу відповіді, навантаження на процесор, частоти звернень до окремих функціональних модулів, а також фіксація виняткових ситуацій. Накопичені логи використовуються для аналізу поведінки системи, пошуку потенційних вузьких місць та виявлення нерівномірності розподілу навантаження. Це також дозволяє забезпечити оперативне реагування на проблему у разі виникнення відмови, що є критично важливим для платформ такого типу.

Для забезпечення надійної роботи та можливості горизонтального чи вертикального масштабування серверний вузол має відповідати певним вимогам щодо апаратної конфігурації. Рекомендованою конфігурацією є 4-8 віртуальних процесорних ядер (vCPU), що гарантують достатній рівень паралельності обробки запитів. Обсяг оперативної пам'яті на рівні 16-32 GB дозволяє ефективно виконувати алгоритми машинного навчання, працювати з кешованими даними та забезпечувати стабільну роботу при великій кількості одночасних підключень. Використання NVMe SSD забезпечує високу швидкість роботи з файлами та базою даних, що є критично важливим у сценаріях великої кількості операцій читання та запису. Додаткова можливість підключення GPU є опціональною та використовується у випадках необхідності прискорення ML-обчислень. Для наглядності функціонального вузла надано архітектуру у вигляді (рис. 3.3).



Рисунок 3.3 - Загальна архітектура системи і роль серверного вузла в ній

У підсумку серверний вузол виступає центральним елементом архітектури веб-платформи, який забезпечує повноцінну підтримку бекенд-логіки, оброблення запитів, роботу з даними, виконання алгоритмів аналізу та контроль телеметрії. Ретельно спроектована структура вузла дозволяє масштабувати систему відповідно до зростання навантаження, забезпечувати високу продуктивність та гарантувати надійність роботи в реальних умовах експлуатації. Його ефективність безпосередньо впливає на загальну якість роботи платформи, швидкість реагування на запити користувачів та точність аналітичних результатів, що робить цей компонент критично важливим у контексті функціонування всієї системи [5; 9; 12].

3.3 Висновки до розділу

У даному розділі було розглянуто реалізацію загальної структури програмної частини вебплатформи підтримки вибору товарів. Архітектура системи побудована за трирівневим принципом Data Layer - Logic Layer - UI Layer, що забезпечує чіткий розподіл відповідальностей між компонентами, підвищує надійність програмного забезпечення та спрощує його супровід і подальший розвиток.

У ході роботи було розроблено та обґрунтовано структурну схему вебплатформи, яка відображає логічну організацію програмних компонентів та механізми їх взаємодії. Запропонований підхід дозволяє досягти високої модульності системи, забезпечити її стійкість до зростання навантаження, а також адаптивність до змін у вимогах та розширення функціональності.

Особливу увагу приділено серверній інфраструктурі як ключовому елементу платформи. Сервер додатка реалізує основну бізнес-логіку, відповідає за обробку запитів, виконання аналітичних операцій, формування рекомендацій та інтеграцію зі сторонніми сервісами. Використання Linux-середовища та контейнеризації забезпечує стабільність роботи системи, автоматичне масштабування та ефективне керування обчислювальними ресурсами.

Обґрунтовано вибір документно-орієнтованої бази даних, що дозволяє ефективно зберігати та обробляти інформацію про товари, їх характеристики та метадані, забезпечуючи гнучкість структури даних і високу продуктивність. Додатково було виділено сховище медіафайлів як окремий компонент, що оптимізує роботу з великими обсягами неструктурованих даних.

Окремо розглянуто функціональний вузол платформи, який забезпечує централізовану обробку бізнес-логіки, виконання алгоритмів машинного навчання та взаємодію з іншими компонентами системи. Його впровадження створює надійну основу для реалізації інтелектуальних сервісів і гарантує стабільну роботу платформи при високій кількості користувачів.

4 ЕКСПЕРИМЕНТ ТА АНАЛІЗ РЕЗУЛЬТАТІВ ВЕБ-ПЛАТФОРМИ

4.1 Моделювання платформи

Етап моделювання розробленої веб-платформи був спрямований на комплексну перевірку її працездатності, продуктивності та здатності стабільно функціонувати за умов різних рівнів навантаження. Метою цього етапу стало підтвердження коректності архітектурних рішень, відповідності вибраних технологій функціональним вимогам та забезпечення надійності системи під час реальної експлуатації.

Крім розробки архітектури, у роботі важливе місце займає тестування програмних компонентів, яке підтверджує коректність їх роботи у реальних сценаріях використання. Функціональні перевірки, навантажувальні випробування та експериментальне оцінювання продуктивності дозволили визначити стабільність алгоритмів, швидкість відгуку API та точність рекомендацій.

Першою фазою стало моделювання навантаження, під час якого проводилися інтенсивні перевірки роботи серверної частини та бази даних у режимах зростаючої кількості одночасних запитів. Для цього використовувалися інструменти, що генерували запити до API платформи з різною частотою та у різних комбінаціях. Аналізувалися такі показники, як час відповіді сервера, стабільність підтримання підключення, швидкість виконання операцій вибірки та фільтрації даних, а також вплив обсягу бази на загальну продуктивність. Тестування показало, що система здатна стабільно опрацьовувати понад 10000 запитів на хвилину без критичного збільшення часу відповіді. Навіть за умов пікової активності середнє значення затримки залишалося в межах норми, що свідчить про достатню оптимізацію серверної логіки та ефективну систему кешування.

Окремим напрямом моделювання стало тестування відмовостійкості, під час якого відтворювалися ситуації можливих збоїв, характерних для реальної роботи веб-сервісів. Імітувалися часткові та повні зупинки окремих модулів, зниження пропускної здатності мережевого каналу, тимчасова недоступність бази даних або

перевантаження окремих сервісів. Метою було визначити, як платформа реагує на подібні ситуації, чи забезпечується коректне оброблення помилок і чи зберігається цілісність даних. Результати свідчать, що система адекватно реагує на відмови: під час симульованих збоїв сервер відновлював роботу без критичних втрат продуктивності, а механізми обробки виняткових ситуацій забезпечували стабільність роботи користувацького інтерфейсу. Це демонструє достатній рівень опрацьованості архітектури, а також правильний вибір технологічного стека.

Особливу увагу було приділено моделюванню великої кількості запитів до бази даних. Оскільки функціонування платформи ґрунтується на швидкому порівнянні параметрів товарів, ефективність роботи бази є одним з ключових аспектів. У рамках тестування база наповнювалася значним обсягом структурованих записів, після чого проводилися масові операції пошуку, сортування, фільтрації та агрегації(рис. 4.1).

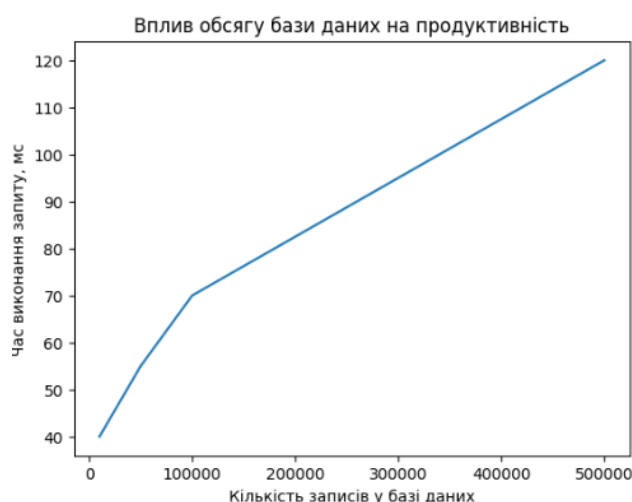


Рисунок 4.1 - Тестування бази даних на продуктивність

Для підвищення наочності отриманих результатів та кращого їх сприйняття основні показники продуктивності й стабільності системи подано у вигляді діаграми, що дозволяє наочно оцінити поведінку платформи за різних сценаріїв навантаження також побудована таблиця показників (табл 4.1):

Таблиця 4.1 - Результат моделювання вебплатформи

Показник	Низьке навантаження	Середнє навантаження	Пікове навантаження
Кількість запитів, запитів/хв	1 000	5 000	10 000+
Середній час відповіді API, мс	60	85	120
Максимальний час відповіді, мс	90	130	180
Кількість помилок, %	0	<0,1	<0,3
Завантаження CPU сервера, %	22	48	71
Час вибірки з БД, мс	35	55	78
Стабільність UI	Повна	Повна	Повна

Дослідження продемонструвало відсутність суттєвих втрат продуктивності навіть при значному збільшенні обсягу даних. Система ефективно обробляла складні запити, що підтверджує коректність побудови індексів та оптимізацію SQL- або NoSQL-структур (залежно від реалізації конкретного прототипу).

Під час узагальнення результатів моделювання встановлено, що веб-платформа демонструє високий рівень стабільності та продуктивності при різних сценаріях використання. Взаємодія між серверною частиною, базою даних та клієнтським інтерфейсом відбувається без затримок, характерних для перевантажених систем. Архітектурні рішення, покладені в основу розробки, підтвердили свою ефективність навіть у випадках симуляції пікових навантажень або часткових технічних збоїв [1; 4; 14].

4.2 Тестування платформи та оцінка працездатності шляхом порівняння

Тестування включало два ключові етапи: функціональну перевірку модулів системи та експериментальне порівняльне оцінювання, спрямоване на визначення впливу реалізованих алгоритмів на швидкість і точність пошуку товарів. Такий підхід дозволив всебічно проаналізувати якість програмної реалізації та перевірити

заявлену гіпотезу щодо підвищення зручності та ефективності взаємодії користувачів із платформою.

Функціональний етап тестування був спрямований на перевірку коректності роботи кожного компонента серверної та клієнтської частин системи. Перевірялися модулі логіки фільтрації, API-інтерфейси, механізми взаємодії користувацького інтерфейсу з сервером, а також підсистема машинного навчання, що відповідає за класифікацію та групування товарних позицій. Основна увага приділялася саме відпрацюванню алгоритмів обробки даних та відповідності фактичних результатів очікуваням.

Тестування API-взаємодії - аналізувалися стабільність відповіді REST-ендпоінтів, час обробки запитів, відповідність структури JSON-повідомлень специфікації. Середній час відповіді становив 80–120 мс, що відповідає вимогам до високопродуктивних вебплатформ та підтверджує оптимальність реалізованої логіки бекенду.

При тестуванні роботи моделі кластеризації перевірялась правильність групування товарів за подібними характеристиками. У тестових сценаріях модель стабільно формувала однорідні кластери та коректно відносила нові елементи до існуючих груп.

Перевірка відображення рекомендацій у користувацькому інтерфейсі. Оцінювалася коректність інтеграції серверних рекомендацій у UI, зокрема синхронізація між швидкими фільтрами, блоками «Схожі товари» та «Рекомендовані». Платформа забезпечила узгодженість даних у всіх тестових кейсах.

Проведене функціональне тестування засвідчило повну працездатність усіх ключових модулів. Жодних критичних помилок виявлено не було, а продуктивність API підтвердила відповідність цільовим вимогам до швидкодії.

Другим етапом стало експериментальне тестування, спрямоване на практичну оцінку того, як розроблена платформа впливає на якість та швидкість пошуку товарів у порівнянні з існуючими рішеннями. Дослідження проводилося на вибірці з 20 учасників, яким пропонувалося виконати серію сценаріїв пошуку за

однаковими товарами, використовуючи як розроблену систему, так і альтернативні вебплатформи.

Для кожного учасника фіксувалися три кількісні критерії:

- середній час пошуку одного товару. Час від початку взаємодії до знаходження коректної товарної позиції. Стимулює оцінку зручності інтерфейсу та ефективності рекомендацій;

- кількість помилкових виборів. Показує, наскільки точно користувач розпізнає та відбирає потрібну категорію або товар, що особливо важливо для великих каталогів;

- кількість кліків, необхідних для отримання цільового результату. Відображає глибину навігації та загальну оптимальність користувацького сценарію.

Результати кількісного порівняння розробленої експериментальної платформи з популярними комерційними вебсервісами та сценарієм ручного пошуку наведено в (табл. 4.1).

Таблиця 4.1 - Кількісні критерії

Платформа	Час, хв	Помилки	Кліки
Rozetka	4,8	2	13
Amazon	5,1	3	15
Hotline	4,5	1	11
Ручний каталог	5,0	4	18
Експериментальна платформа	2,95	0–1	6

Для кожної платформи оцінювалися три ключові показники: середній час пошуку товару, кількість помилкових виборів та кількість кліків, необхідних для досягнення цільового результату.

Аналіз наведених даних показує, що експериментальна платформа демонструє найкращі результати за всіма критеріями. Середній час пошуку товару

становить 2,95 хв, що майже на 35–45 % менше порівняно з існуючими вебплатформами та ручним каталогом. Це свідчить про високу ефективність реалізованих механізмів фільтрації, кластеризації та рекомендацій.

Кількість помилкових виборів на експериментальній платформі коливалася в межах від 0 до 1, тоді як для інших сервісів цей показник є суттєво вищим. Даний факт підтверджує підвищену точність інтелектуального підбору товарів і здатність системи коректно спрямовувати користувача до релевантних результатів.

Показник кількості кліків також є мінімальним для експериментальної платформи (у середньому 6 кліків), що свідчить про оптимізований користувацький сценарій та зменшену глибину навігації. На відміну від цього, альтернативні сервіси потребують значно більшої кількості взаємодій, що збільшує когнітивне навантаження на користувача та час виконання завдання.

Для наглядності навели діаграму з порівняннями вебсервісів (рис.4.1).

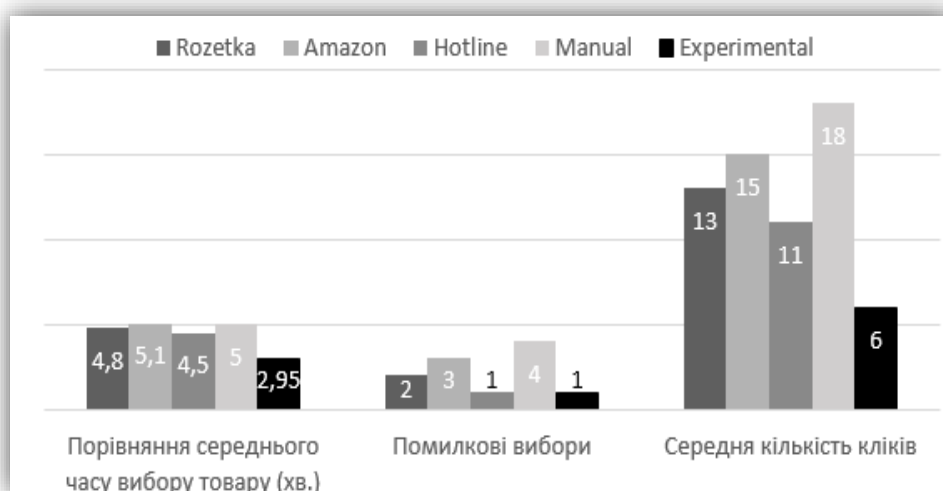


Рисунок 4.1 - Порівняльні результати вебсервісів

У процесі дослідження була проведена комплексна оцінка працездатності та ефективності створеної інтелектуальної вебплатформи, призначеної для підбору товарів з використанням методів машинного навчання та рекомендаційних моделей. Для оцінки ефективності розробленої системи було проведено

порівняльний аналіз з існуючими рішеннями. Результати засвідчили, що використання створеної інтелектуальної платформи дозволило суттєво скоротити час виконання завдання та зменшити кількість навігаційних дій. Завдяки інтеграції алгоритмів кластеризації та рекомендацій кількість помилкових виборів зменшилася майже до нуля [3; 6]. Це підтверджує ключову гіпотезу дослідження щодо того, що використання моделей машинного навчання в інтерфейсах пошуку товарів підвищує точність взаємодії та зменшує когнітивне навантаження на користувача [20].

4.3 Висновки щодо експериментального дослідження

Проведене експериментальне тестування підтвердило ефективність розробленої інтелектуальної вебплатформи підтримки вибору товарів. Отримані результати демонструють, що інтеграція методів машинного навчання, зокрема кластеризації та рекомендаційних моделей, дозволяє суттєво підвищити швидкість пошуку, зменшити кількість помилкових дій та оптимізувати користувацьку навігацію.

Порівняльний аналіз із існуючими вебсервісами показав, що експериментальна платформа забезпечує більш інтуїтивну та результативну взаємодію користувача з товарним каталогом. Зменшення кількості кліків і помилкових виборів свідчить про зниження когнітивного навантаження та підвищення загальної зручності користування системою.

Окрему увагу в межах експериментального дослідження було приділено аналізу продуктивності та відмовостійкості вебплатформи за різних рівнів навантаження. Згідно з результатами моделювання система демонструє стабільну роботу як за низького, так і за пікового навантаження. При цьому середній час відповіді серверного API відповідає вимогам до сучасних інтерактивних вебсистем.

Аналіз серверних метрик свідчить про ефективне використання обчислювальних ресурсів. Навіть за пікового навантаження рівень завантаження CPU не перевищує 71 %, а кількість помилок залишається меншою за 0,3 %, що підтверджує коректність реалізації асинхронної обробки запитів та оптимізованої взаємодії з базою даних. Час вибірки з БД зростає пропорційно навантаженню, проте не створює критичних затримок у загальному ланцюгу обробки запиту.

Крім того, результати тестування підтверджують високу стабільність клієнтського інтерфейсу платформи. За всіх сценаріїв навантаження не було зафіксовано збоїв у відображенні рекомендацій або порушень логіки навігації, що є важливим чинником для забезпечення позитивного користувацького досвіду. Отримані результати свідчать про готовність розробленої вебплатформи до експлуатації в умовах реального комерційного середовища та підтверджують доцільність обраної архітектури й технологічних рішень.

Таким чином, результати експерименту підтверджують заявлену гіпотезу дослідження про доцільність використання інтелектуальних алгоритмів у системах пошуку та підбору товарів. Розроблена платформа може розглядатися як ефективна альтернатива традиційним підходам та має потенціал для подальшого масштабування і впровадження у реальні комерційні середовища.

ВИСНОВКИ

У дипломній роботі було досліджено сучасні підходи до побудови рекомендаційних систем у контексті цифрових вебплатформ електронної комерції. Особливу увагу приділено проблемі інформаційного перевантаження користувачів та необхідності інтелектуальної підтримки процесу вибору товарів на основі аналізу даних. Аналіз наукових джерел і практичних рішень показав, що класичні методи фільтрації поступово доповнюються або замінюються гібридними моделями, які поєднують контентні, поведінкові та машинні підходи, забезпечуючи вищу релевантність і персоналізацію рекомендацій.

Важливою складовою розробленої рекомендаційної системи є її математична структура, що ґрунтується на формалізації користувацьких уподобань та характеристик товарів у вигляді векторних моделей. Застосування методів зважування ознак, метрик подібності та агрегування результатів дозволило описати процес підбору товарів у вигляді узагальненої математичної моделі, яка забезпечує адаптивність рекомендацій і враховує динамічну зміну поведінкових профілів користувачів.

У результаті проведеного дослідження було розроблено комплексну архітектуру вебплатформи підтримки вибору товарів з елементами інтелектуального аналізу, що поєднує сучасні технології обробки даних, модульність серверної логіки та адаптивні інтерфейсні рішення. Побудована тривірнева модель - Data Layer, Logic Layer та UI Layer - забезпечила чіткий розподіл функціональних компонентів, що сприяє масштабованості системи, її гнучкості та можливості подальшого вдосконалення без порушення роботи інших модулів.

Реалізовані модулі контентного аналізу, кластеризації й персоналізації створюють інтелектуальне ядро платформи. Вони забезпечують автоматичне структурування товарних характеристик, групування продуктів за ознаками подібності та формування рекомендацій з урахуванням індивідуальних

поведінкових патернів користувачів. Інтеграція цих алгоритмів із серверним оточенням та REST API дозволила сформувати цілісну систему, здатну оперативно опрацьовувати запити та забезпечувати високу точність пошуку й порівняння товарів.

Моделювання та тестування показали стабільність роботи платформи під навантаженням, відсутність значних затримок обробки запитів, а також коректність реалізації алгоритмів. Функціональні тести підтвердили правильність роботи фільтрів, API та механізмів відображення рекомендацій, тоді як експериментальні випробування за участю 20 користувачів засвідчили суттєве скорочення часу пошуку товарів і мінімізацію кількості помилкових виборів. Це доводить практичну ефективність застосованих методів інтелектуального аналізу та доцільність їх використання у порівняльних вебплатформах.

Отже, розроблена програмна частина вебплатформи підтвердила свою працездатність, ефективність і відповідність поставленим вимогам. Вона може бути використана як основа для створення повноцінного сервісу рекомендацій та порівняння товарів, а також слугувати базою для подальших досліджень, спрямованих на розширення алгоритмічного функціоналу, оптимізацію продуктивності та покращення користувацької взаємодії.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Charu C. Aggarwal. Recommender Systems: The Textbook. Springer, 2016.
2. Верес О., Гадзало О. Застосування методів рекомендацій в процесах аналізу комп'ютерних комплектуючих. Інформаційні системи та мережі. 2023. URL: <https://science.lpnu.ua/sites/default/files/journal-paper/2024/jan/33140/231474maket1-86-100.pdf>.
3. Корнієнко Д. С. «Гібридні рекомендаційні системи: методи, метрики, виклики». Полтава: Полтавський університет, 2025. URL: <https://reposit.nupp.edu.ua/bitstream/PoltNTU/18461/1/Корнієнко.pdf>.
4. Жуков О. П., Петренко В., Галушко Д. Ефективність архітектур гібридних рекомендаційних систем. UNIVERSUM, 2024. URL: <https://archive.liga.science/index.php/universum/article/view/1123>.
5. Донцова О. Є. ««Інформаційна технологія рекомендації товарів для платформи онлайн продажів». Суми: СумДУ, 2020. URL: https://essuir.sumdu.edu.ua/bitstream/123456789/78592/1/Dontsova_mag.rob.pdf.
6. Çano E., Morisio M. Hybrid Recommender Systems: A Systematic Literature Review, arXiv preprint. 2019. URL: <https://arxiv.org/abs/1901.03888>.
7. Чумак В. Огляд рекомендаційних систем у дипломному проєкті: «Рекомендаційна система онлайн бібліотеки» Київ: КІП ім. І. Сікорського, 2023. URL: https://ela.kpi.ua/bitstream/123456789/58648/1/Chumak_bakalavr.pdf.
8. Кучерук В., Глушко М. Покращення якості рекомендованих систем на основі кваліметричних методів вимірювання. Вимірювальна та обчислювальна техніка в технологічних процесах. 2022. URL: <https://vottp.khmnu.edu.ua/index.php/vottp/article/download/50/50>.
9. Portugal I., Alencar P., Cowan D. The Use of Machine Learning Algorithms in Recommender Systems: A Systematic Review, arXiv preprint. 2015. URL: <https://arxiv.org/abs/1511.05263>.

10. Mueller J. Combining Aspects of Genetic Algorithms with Weighted Recommender Hybridization, arXiv preprint. 2017 URL: <https://arxiv.org/abs/1710.10177>.
11. Russell S., Norvig P. Artificial Intelligence: A Modern Approach. Pearson, 2021.
12. Ricci F., Rokach L., Shapira B. Recommender Systems Handbook. Springer, 2022.
13. Burke R. Hybrid Recommender Systems: Survey and Experiments. *User Modeling and User-Adapted Interaction*. 2002.
14. Koren Y., Bell R., Volinsky C. Matrix Factorization Techniques for Recommender Systems. *IEEE Computer*. 2009.
15. Jannach D., Zanker M., Felfernig A. Recommender Systems: An Introduction. Cambridge University Press, 2016.
16. Goodfellow I., Bengio Y., Courville A. Deep Learning. MIT Press, 2016.
17. Zhang S., Yao L., Sun A. Deep Learning based Recommender System: A Survey. *ACM Computing Surveys*. 2019.
18. He X., Liao L., Zhang H. Neural Collaborative Filtering. *Proceedings of the WWW Conference*. 2017.
19. Shani G., Gunawardana A. Evaluating Recommendation Systems. *Recommender Systems Handbook*. Springer, 2011.
20. Amazon Science. Item-to-Item Collaborative Filtering. *IEEE Internet Computing*. 2003.
21. Пальчик Т.В., Голуб Т.В. Порівняння рекомендаційних вебсервісів підбору товарів на цифрових платформах. Інтелектуальні інформаційні системи: всеукр. наук.-практ. конф. (Миколаїв, 4-5 грудня 2025). Миколаїв, 2025. (у друці)

ДОДАТОК А

ЛІСТИНГИ ПРОГРАМИ

Лістинг А.1 - Текстовий вигляд платформи в HTML

```
<!DOCTYPE html>
  <html lang="uk">
    <head>

      <meta charset="utf-8" />

      <meta name="viewport" content="width=device-width,initial-
scale=1"/>

      <title>Інтелектуальна платформа - MVP</title>
      <link rel="stylesheet" href="2var.css" />
    </head>
    <body>
      <header class="site-header" role="banner">
        <div class="header-inner">
          <h1>Інтелектуальна Платформа Підбору Товарів
          </h1>
          <div class="header-controls">
            <label class="theme-toggle">
              <input id="themeToggle" type="checkbox" aria-
label="Увімкнути темну тему">
                <span>☾</span>
            </label>
          </div>
        </div>
        <p class="tagline">Швидкий і розумний підбір товарів:
контентний аналіз + персоналізація</p>
      </header>
      <main class="container" role="main">
        <aside class="sidebar" aria-label="Фільтри">
          <form id="filters" class="filters" autocomplete="off">
            <div class="search-block">
```

```

        <label for="searchInput" class="visually-hidden">Пошук
товарів</label>
        <input id="searchInput" type="text" placeholder="Пошук
(наприклад: ноутбук)" />
        <ul id="autocompleteList" class="autocomplete" aria-
hidden="true"></ul>
    </div>

    <fieldset class="filter-group">
        <legend>Бюджет</legend>
        <input id="budget" type="range" min="1000" max="100000"
step="500" value="25000" />
        <div class="budget-row">
            <output id="budgetValue">25 000 грн</output>
            <button type="button" id="resetBudget" class="btn-clear"
title="Скинути">✕</button>
        </div>
    </fieldset>

    <fieldset class="filter-group">
        <legend>Пріоритети</legend>
        <label><input type="checkbox" id="prefPerf"> Висока
продуктивність</label>
        <label><input type="checkbox" id="prefBattery"> Велика
автономність</label>
    </fieldset>

    <fieldset class="filter-group">
        <legend>Сумісність IoT</legend>
        <label>
            <input type="checkbox" id="prefIoT"> Підтримка Smart Home
        </label>
    </fieldset>

    <div class="filter-actions">

```

```

        <button type="button" id="applyFilters" class="btn
primary">Застосувати
    </button>
    <button type="button" id="resetFilters"
class="btn">Скинути
    </button>
</div>
</form>
</aside>

<section class="content" aria-label="Результати підбору">
    <div class="content-header">
        <h2>Результати</h2>
        <div class="results-meta">
            <p id="resultCount">Знайдено 0 товарів</p>
            <select id="sortBy" aria-label="Сортування">
                <option value="score">Релевантність</option>
                <option value="price-asc">Ціна (зростання)</option>
                <option value="price-desc">Ціна (зменшення)</option>
            </select>
        </div>
    </div>

    <section id="productGrid" class="product-grid" aria-
live="polite">
        <!-- Карточки товарів додаватимуться JS -->
    </section>
</section>
</main>

<template id="cardTpl">
    <article class="product-card">
        <img class="prod-img" alt="">
        <div class="prod-body">
            <h3 class="prod-title"></h3>
            <p class="prod-desc"></p>

```

```

    <div class="prod-meta">
      <div class="rating"></div>
      <div class="price"></div>
    </div>
    <div class="card-actions">
      <button class="btn btn-fav" title="Додати в
уподобані">♥</button>
      <button class="btn btn-details"
title="Деталі">Деталі</button>
    </div>
  </div>
</article>
</template>

<script src="2var.js" defer></script>
</body>
</html>

```

Лістинг А.2 - Текстовий вигляд платформи в CSS

```

/* Базова система */
:root{
  --bg: #e8ecf0;
  --card:#fff;
  --accent: #2c3e50;
  --muted:#6b6b6b;
  --glass: rgba(255,255,255,0.7);
  --shadow: 0 6px 18px rgba(0,0,0,0.08);
  --radius:30px;
  text-align: center;
  position: relative;
}
*{box-sizing:border-box}
html,body{height:100%}

```

```

body{
margin:0;
font-family:Inter, Segoe UI, Arial, sans-serif;
background:var(--bg);
color:#111;
-webkit-font-smoothing:antialiased;
-moz-osx-font-smoothing:grayscale;
}

/* Dark theme wrapper */
body.dark{
--bg:#0f1113;
--card:#141618;
--accent:#2c3e50;
--muted:#bdbdbd;
--glass: rgba(20,20,20,0.6);
--shadow: 0 8px 20px rgba(0,0,0,0.6);
color:#eaeaea;
}

.site-header{
padding:18px 20px 6px;
background:linear-gradient(90deg,var(--accent), #9bc769);
color:white;
box-shadow:var(--shadow);
}

.header-inner{display:flex;align-items:center;justify-
content:space-between;gap:12px;max-width:1200px;margin:0 auto;}
  .site-header h1{font-size:1.15rem;margin:0}
  .tagline{opacity:0.95;margin:8px auto 0;max-
width:1200px;padding:0 20px;color:rgba(255,255,255,0.95)}

/* Layout */
.container{display:flex;gap:20px;max-width:1200px;margin:20px
auto;padding:0 20px}

```

```

        .sidebar{flex:0 0 300px;background:var(--
card);padding:18px;border-radius:var(--radius);box-shadow:var(--
shadow)}

        .content{flex:1;background:var(--card);padding:18px;border-
radius:var(--radius);box-shadow:var(--shadow)}

        /* Accessibility */

        .visually-
hidden{position:absolute!important;height:1px;width:1px;overflow:hid
den;clip:rect(1px,1px,1px,1px);white-space:nowrap}

        /* Filters */

        .filters .search-block{position:relative;margin-bottom:12px}
        #searchInput{width:100%;padding:10px 12px;border-
radius:8px;border:1px solid #e5e7eb;font-size:14px}
        .dark #searchInput{background:#1b1b1b;border:1px solid
#2b2b2b;color:inherit}

        .autocomplete{
            list-style:none;margin:6px 0 0;padding:6px;border-
radius:8px;border:1px solid #eaeaea;background:var(--
card);position:absolute;left:0;right:0;z-index:30;display:none;max-
height:220px;overflow:auto}
        .autocomplete li{padding:8px;border-radius:6px;cursor:pointer}
        .autocomplete li:hover{background:#f1f5f9}

        /* budget */

        .filter-group{margin-bottom:14px}
        #budget{width:100%}
        .budget-row{display:flex;align-items:center;justify-
content:space-between;margin-top:8px}
        #budgetValue{font-weight:600}

        /* buttons */

        .btn{padding:8px 12px;border-radius:8px;border:1px solid
transparent;background:transparent;cursor:pointer}

```

```

    .btn.primary{background:var(--accent);color:#fff;border-
color:transparent}

    .btn-clear{background:transparent;border:none;color:var(--
muted);cursor:pointer}

/* Content header */
    .content-header{display:flex;align-items:center;justify-
content:space-between;gap:12px;margin-bottom:12px}
    .results-meta{display:flex;align-items:center;gap:12px}

/* Grid */
    .product-grid{display:grid;grid-template-columns:repeat(auto-
fit,minmax(240px,1fr));gap:16px}
    .product-card{display:flex;gap:12px;align-
items:center;padding:12px;border-radius:10px;background:var(--
card);border:1px solid rgba(0,0,0,0.04);transition:transform .18s
ease,box-shadow .18s ease}
    .product-card:hover{transform:translateY(-6px);box-shadow:0
10px 30px rgba(0,0,0,0.08)}
    .prod-img{width:84px;height:84px;border-radius:8px;object-
fit:cover;flex-shrink:0}
    .prod-title{margin:0 0 6px;font-size:1rem}
    .prod-desc{margin:0;color:var(--muted);font-size:0.92rem}
    .prod-meta{display:flex;align-items:center;justify-
content:space-between;margin-top:8px;gap:12px}
    .rating{color:gold}
    .price{font-weight:700}

/* actions */
    .card-actions{display:flex;gap:8px;margin-top:8px}
    .btn-fav{background:transparent;border:1px solid
rgba(0,0,0,0.06);padding:6px 10px;border-radius:8px}
    .btn-details{background:transparent;border:1px solid
rgba(0,0,0,0.06);padding:6px 10px;border-radius:8px}

/* responsive: mobile */

```

```

@media (max-width:820px){
  .container{flex-direction:column;padding:0 12px}
  .sidebar{position:sticky;top:12px;z-index:40;width:100%}
  .product-card{flex-direction:column;align-items:flex-start}
  .prod-img{width:100%;height:180px;border-radius:8px}
}

/* small niceties */
.btn:focus, input:focus{outline:2px solid
rgba(0,0,0,0.08);outline-offset:2px}

```

Лістинг А.2 - Текстовий вигляд платформи в JS

```

/* platform.js
MVP: lightweight "intelligent" ranking + UI interactions
*/

// ----- Demo product dataset (можна замінити на API) -----
const PRODUCTS = [
  { id:1, title:"Ноутбук ProX-15", desc:"Intel i7 · 16GB · 512GB
SSD", price:35000, iot:false, perf:9, battery:6, img:"Ноут.jpg" },
  { id:2, title:"Смартфон Ultra-Z", desc:"8GB · 128GB · Камера
108MP", price:21000, iot:false, perf:7, battery:8, img:"хпери.jpg"
},
  { id:3, title:"IoT Камера SmartView", desc:"HD · Night vision ·
Wi-Fi", price:4500, iot:true, perf:4, battery:7, img:"камера.png" },
  { id:4, title:"Телевізор SmartTV 55\"", desc:"4K · Smart OS ·
HDMI 2.1", price:28000, iot:true, perf:6, battery:0, img:"тв.png" },
  { id:5, title:"Безпроводні навушки AERO", desc:"ANC · 30h
battery", price:5200, iot:false, perf:5, battery:9,
img:"наушки.png" },
  { id:6, title:"Робот-пилосос CleanBot", desc:"Lidar · Smart
Home", price:15000, iot:true, perf:5, battery:8, img:"пилосос.png" }
];

```

```

// ----- Helpers -----
const $ = (sel) => document.querySelector(sel);
const $all = (sel) =>
Array.from(document.querySelectorAll(sel));

function formatPrice(n){ return new Intl.NumberFormat('uk-
UA').format(n) + " грн"; }

// Debounce
function debounce(fn, wait=250){
let t;
return (...args) => { clearTimeout(t); t =
setTimeout(()=>fn(...args), wait); }
}

// Simple similarity: counts keyword matches in title+desc
function textScore(item, query){
if(!query) return 0;
const text = (item.title + " " + item.desc).toLowerCase();
const tokens =
query.toLowerCase().split(/\s+/).filter(Boolean);
let s = 0;
tokens.forEach(tok => { if(text.includes(tok)) s += 1; });
return s / Math.max(1, tokens.length); // normalized
}

// Core scoring function (симуляція контентного аналізу)
function computeScore(item, state){
// weights (можна тонко налаштувати)
const W_TEXT = 0.45;
const W_PRICE = 0.25;
const W_PREF = 0.2;
const W_IOT = 0.1;

// text match (0..1)

```

```

const t = textScore(item, state.query);

// price: closer to budget preferred (score 0..1)
const b = state.budget || 25000;
// if price <= budget => reward, else penalize
let p = 0;
if(item.price <= b) p = 1 - (item.price / Math.max(1, b)); //
cheaper => higher
else p = Math.max(0, 1 - (item.price - b) / b); // over budget
penalized

// pref: if user requires perf/battery -> normalize item.perf
and item.battery (0..1)
let prefScore = 0;
if(state.prefPerf) prefScore += (item.perf || 0) / 10;
if(state.prefBattery) prefScore += (item.battery || 0) / 10;
if(state.prefPerf && state.prefBattery) prefScore /= 2; //
average if both

// IoT matching
const iotScore = state.prefIoT ? (item.iot ? 1 : 0) : 1;

// final
const score = (W_TEXT * t) + (W_PRICE * p) + (W_PREF *
prefScore) + (W_IOT * iotScore);
return Math.max(0, score); // ensure non-negative
}

// ----- State -----
const state = {
  query: '',
  budget: 25000,
  prefPerf: false,
  prefBattery: false,
  prefIoT: false,
  sortBy: 'score'
}

```

```

};

// ----- UI refs -----
const searchInput = document.getElementById('searchInput');
const autoCompleteList =
document.getElementById('autocompleteList');
const budget = document.getElementById('budget');
const budgetValue = document.getElementById('budgetValue');
const applyBtn = document.getElementById('applyFilters');
const resetBtn = document.getElementById('resetFilters');
const productGrid = document.getElementById('productGrid');
const resultCount = document.getElementById('resultCount');
const sortBy = document.getElementById('sortBy');
const themeToggle = document.getElementById('themeToggle');

// persist favorites
const FAV_KEY = 'mvp_favs';
function loadFavs(){ try{ return
JSON.parse(localStorage.getItem(FAV_KEY)) || []; }catch{ return [] }
}

function saveFavs(arr){ localStorage.setItem(FAV_KEY,
JSON.stringify(arr)); }
let favs = loadFavs();

// ----- Rendering -----
const tpl = document.getElementById('cardTpl');

function renderProducts(list){
productGrid.innerHTML = '';
if(!list.length){
productGrid.innerHTML = `<div class="empty">За заданими
критеріями нічого не знайдено.</div>`;
resultCount.textContent = `Знайдено 0 товарів`;
return;
}
resultCount.textContent = `Знайдено ${list.length} товарів`;

```

```

list.forEach(product => {
  const node = tpl.content.cloneNode(true);
  const art = node.querySelector('article');
  const img = node.querySelector('.prod-img');
  const title = node.querySelector('.prod-title');
  const desc = node.querySelector('.prod-desc');
  const rating = node.querySelector('.rating');
  const price = node.querySelector('.price');
  const favBtn = node.querySelector('.btn-fav');

  img.src = product.img;
  img.alt = product.title;
  title.textContent = product.title;
  desc.textContent = product.desc;
  rating.textContent = '★'.repeat(Math.round((product.perf +
product.battery)/2/2)) || '★';
  price.textContent = formatPrice(product.price);

  // fav state
  favBtn.textContent = favs.includes(product.id) ? '♥' : '♡';
  favBtn.addEventListener('click', ()=>{
    if(favs.includes(product.id)) favs =
favs.filter(x=>x!==product.id);
    else favs.push(product.id);
    saveFavs(favs);
    favBtn.textContent = favs.includes(product.id) ? '♥' : '♡';
  });

  // details -> simple alert for demo
  node.querySelector('.btn-details').addEventListener('click',
()=> {
    alert(`${product.title}\n\n${product.desc}\nЦіна:
${formatPrice(product.price)}\nIoT: ${product.iot ? 'Так':'Ні'}`);
  });

```

```

productGrid.appendChild(node);
});
}

// ----- Filtering & Ranking -----
function applyFilters(){
  // compute scores
  let results = PRODUCTS.map(item => {
    const score = computeScore(item, state);
    return { ...item, score };
  });

  // filter out very low scores (threshold)
  results = results.filter(r => r.score > 0.02);

  // sort
  if(state.sortBy === 'price-asc') results.sort((a,b)=>a.price-
b.price);
  else if(state.sortBy === 'price-desc')
results.sort((a,b)=>b.price-a.price);
  else results.sort((a,b)=>b.score - a.score);

  renderProducts(results);
}

// ----- Autocomplete (debounced) -----
const productsListForAuto = PRODUCTS.map(p => p.title);
const showAutocomplete = (items) => {if(!items.length){
autocompleteList.style.display='none';
autocompleteList.innerHTML=''; return; }
  autocompleteList.innerHTML = items.map(i=>`<li
role="option">${i}</li>`).join('');
  autocompleteList.style.display = 'block';
  autocompleteList.setAttribute('aria-hidden','false');
};

```

```

searchInput.addEventListener('input', debounce((e)=>{
  const q = e.target.value.trim();
  state.query = q;
  if(!q){ showAutocomplete([]); return; }
  const matches = productsListForAuto.filter(p =>
p.toLowerCase().includes(q.toLowerCase())).slice(0,6);
  showAutocomplete(matches);
}, 180));

autocompleteList.addEventListener('click', (ev)=>{
  if(ev.target.tagName === 'LI'){ searchInput.value =
ev.target.textContent; state.query = ev.target.textContent;
showAutocomplete([]); }
});

// hide autocomplete if click outside
document.addEventListener('click', (e)=>{
  if(!e.target.closest('.search-block')) showAutocomplete([]);
});

// ----- Controls -----
budget.addEventListener('input', (e)=>{ budgetValue.textContent
= formatPrice(Number(e.target.value)); state.budget =
Number(e.target.value); });
document.getElementById('prefPerf').addEventListener('change',
(e)=> state.prefPerf = e.target.checked);
document.getElementById('prefBattery').addEventListener('change
', (e)=> state.prefBattery = e.target.checked);
document.getElementById('prefIoT').addEventListener('change',
(e)=> state.prefIoT = e.target.checked);
sortBy.addEventListener('change', (e)=> { state.sortBy =
e.target.value; applyFilters(); });

// apply/reset buttons
applyBtn.addEventListener('click', ()=> { applyFilters(); });
resetBtn.addEventListener('click', ()=>{

```

```

    // reset form UI and state
    searchInput.value = ''; state.query=''; budget.value = 25000;
    budgetValue.textContent = formatPrice(25000);
    document.getElementById('prefPerf').checked = false;
    document.getElementById('prefBattery').checked = false;
    document.getElementById('prefIoT').checked = false;
    state.prefPerf = state.prefBattery = state.prefIoT = false;
    state.budget = 25000; state.sortBy='score'; sortBy.value='score';
    applyFilters();
  });

  // reset budget quick
  document.getElementById('resetBudget').addEventListener('click'
, ()=>{
    budget.value = 25000; budget.dispatchEvent(new Event('input'));
  });

  // theme
  themeToggle.addEventListener('change', ()=> {
    document.body.classList.toggle('dark', themeToggle.checked);
  });

  // initial render
  (function init(){
    // set defaults to state
    state.budget = Number(budget.value);
    budgetValue.textContent = formatPrice(state.budget);
    applyFilters();
  })();

```

ДОДАТОК Б СЛАЙДИ ПРЕЗЕНТАЦІЇ

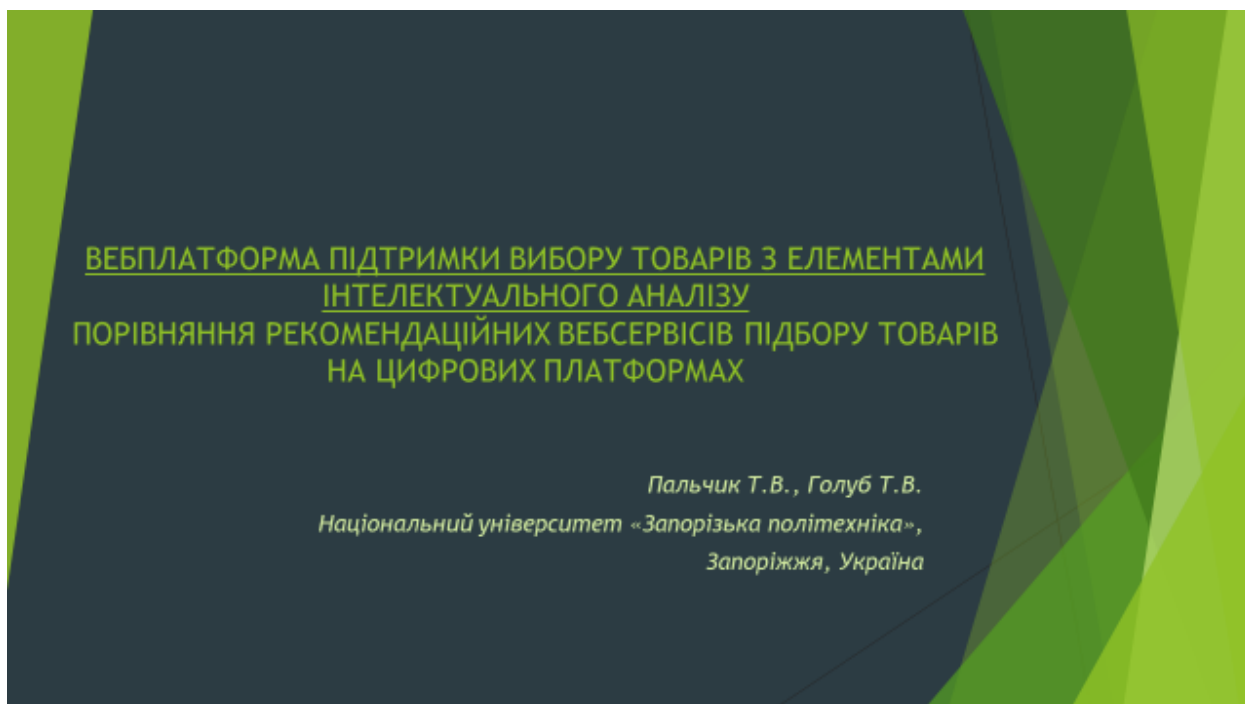


Рисунок Б.1 - Слайд 1

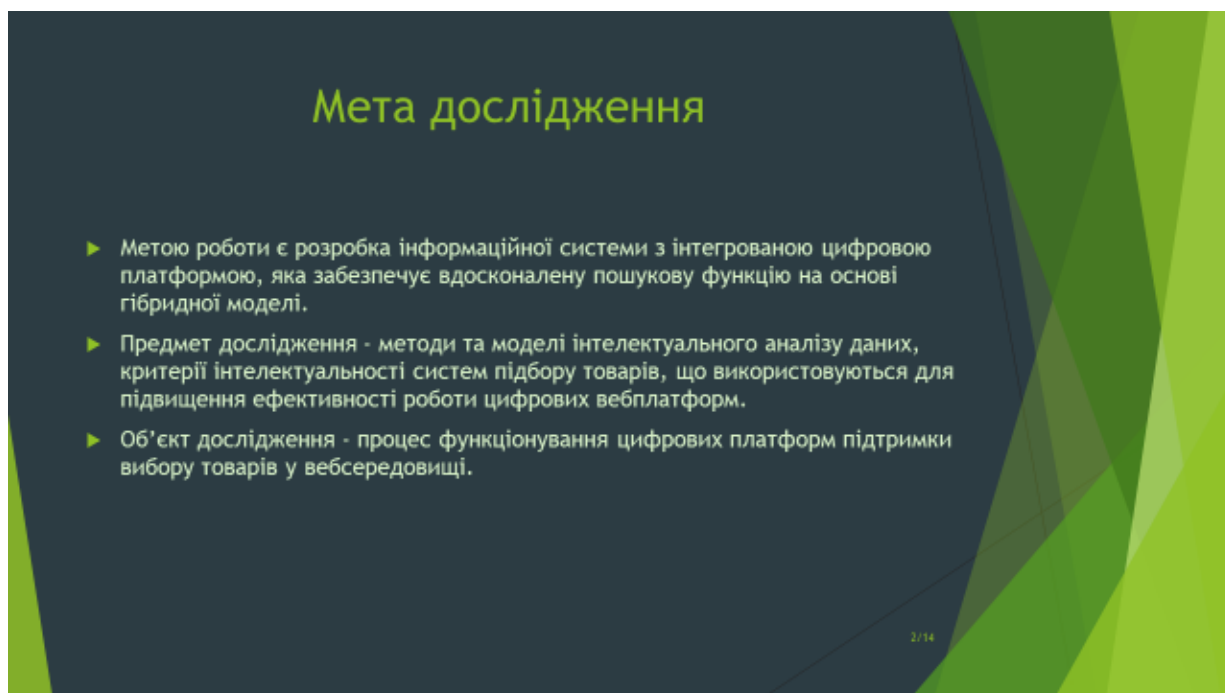


Рисунок Б.2 - Слайд 2



Рисунок Б.3 - Слайд 3

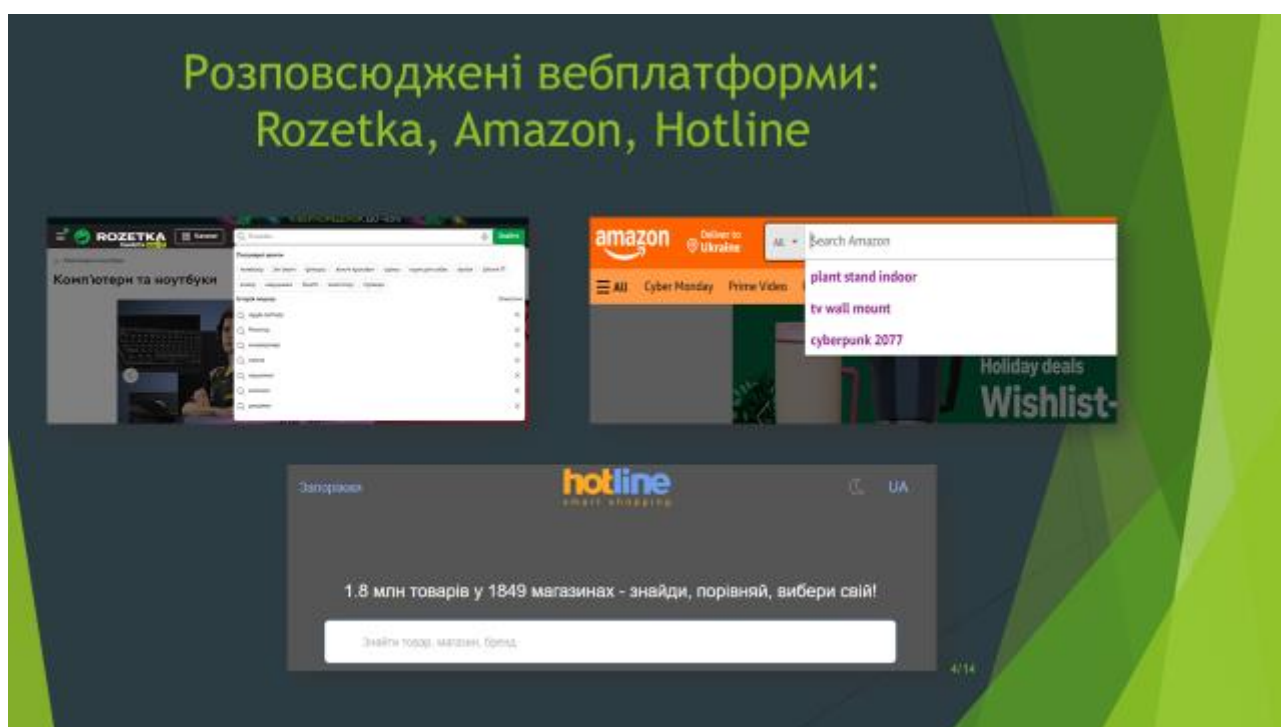


Рисунок Б.4 - Слайд 4

Рекомендаційна (інтелектуальна) система



- Комплекс апаратно-програмних засобів, який на основі накопичених знань, досвіду та алгоритмів машинного навчання здатний самостійно або з мінімальним втручанням людини вирішувати завдання, що раніше вимагали участі експерта.
- Основною метою таких систем є імітація процесів людського мислення - аналізу, узагальнення, прогнозування та прийняття рішень.

5/14

Рисунок Б.5 - Слайд 5

Алгоритми підбору рекомендацій для інтелектуальних систем

- контентний аналіз;
- кластеризація характеристик товарів;
- моделі персоналізації на основі поведінкових патернів.

6/14

Рисунок Б.6 - Слайд 6



Рисунок Б.7 - Слайд 7

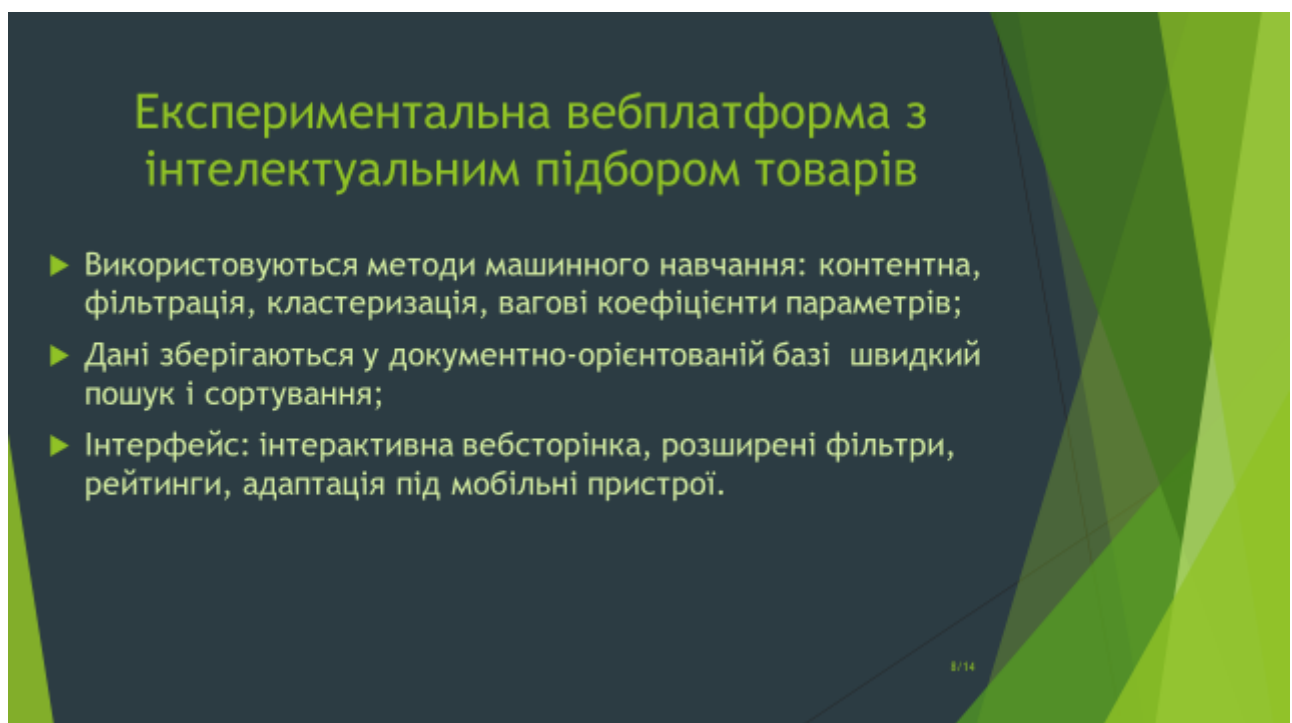


Рисунок Б.8 - Слайд 8

Структура системи модулів з описом функціоналу

Модуль	Основне призначення	Вхідні дані	Вихідні результати	Методи та алгоритми
Модуль кластеризації	Групування користувачів або товарів за подібністю	Поведінкові дані, характеристики товарів	Кластери користувачів / товарів	k-means, DBSCAN, ієрархічна кластеризація
Модуль контентного аналізу	Аналіз властивостей товарів	Опис товарів, атрибути, метадані	Вектор ознак товару	TF-IDF, cosine similarity, embedding
Модуль інтелектуального підбору	Формування рекомендацій	Дані кластеризації + контентні ознаки	Список рекомендованих товарів	Гібридні рекомендаційні моделі
Модуль персоналізації	Адаптація рекомендацій під користувача	Історія дій користувача	Персоналізований рейтинг товарів	Вагові коефіцієнти, профілі користувачів

9/14

Рисунок Б.9 - Слайд 9

Визначення якості рекомендацій

Контент-орієнтоване подання товару

u — користувач;
 i — товар;
 $\vec{a}_i$ — контентний вектор товару;
 $\vec{p}_u$ — персоналізований профіль користувача;
 C_j — кластер, до якого належить товар i ;
 w_d — вагові коефіцієнти складової моделі.
 $\vec{v}_i = [w_{text} \cdot \text{TF-IDF}(d_i), w_{attr} \cdot \vec{a}_i]$
 d_i — текстовий опис товару;
 $\vec{a}_i$ — нормалізований вектор атрибутів;
 $w_{text} + w_{attr} = 1$.

Процедура кластеризації товарів

$$C_j = \text{Cluster}(\vec{v}_i), \quad i \in C_j$$

Персоналізований профіль користувача

$$\vec{p}_u = \frac{1}{\sum_{i \in I_u} \alpha_i} \sum_{i \in I_u} \alpha_i \cdot \vec{v}_i$$

I_u — множина взаємодій користувача;
 α_i — вага типу взаємодії.

Загальна оцінка релевантності рекомендаційної системи

$$R(u, i) = \omega_1 \cdot \cos(\vec{p}_u, \vec{v}_i) + \omega_2 \cdot S_{clust}(u, i) + \omega_3 \cdot S_{pers}(u, i)$$

Кількісна оцінка релевантності товару

Контентна схожість
 $S_{cont}(u, i) = \cos(\vec{p}_u, \vec{v}_i)$
 Кластерна релевантність
 $S_{clust}(u, i) = \begin{cases} 1, & i \in C_u \\ 0, & i \notin C_u \end{cases}$
 Персоналізована відповідність
 $S_{pers}(u, i) = f(\text{pref}_{u,i}, \vec{a}_i)$

10/14

Рисунок Б.10 - Слайд 10



Рисунок Б.11 - Слайд 11



Рисунок Б.12 - Слайд 12

Дослідження розробленої системи

- У дослідженні взяли участь 20 людей.
- Кожен мав знайти однакові групи товарів на різних платформах.

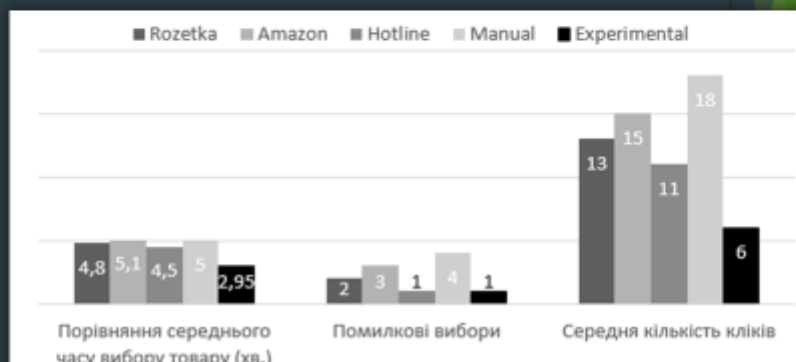


Рисунок Б.13 - Слайд 13

Висновки і перспективи

- При дослідженні було розроблено комплексну структуру гібридної моделі підтримки вибору товарів з елементами інтелектуального аналізу, що поєднує сучасні технології обробки даних, модульність серверної логіки та адаптивні інтерфейсні рішення;
- Контентний аналіз та кластеризація суттєво прискорюють процес вибору товарів, а модель вагових коефіцієнтів покращує аналітичні властивості;
- Алгоритмічний підбір підвищує точність рекомендацій і економить час користувача;
- Моделювання та тестування показали стабільність роботи платформи під навантаженням, відсутність значних затримок обробки запитів, а також коректність реалізації алгоритмів;
- Платформу можна покращити за рахунок Big Data, прогнозування цін, сезонності та інтеграції нових постачальників.

Рисунок Б.14 - Слайд 14

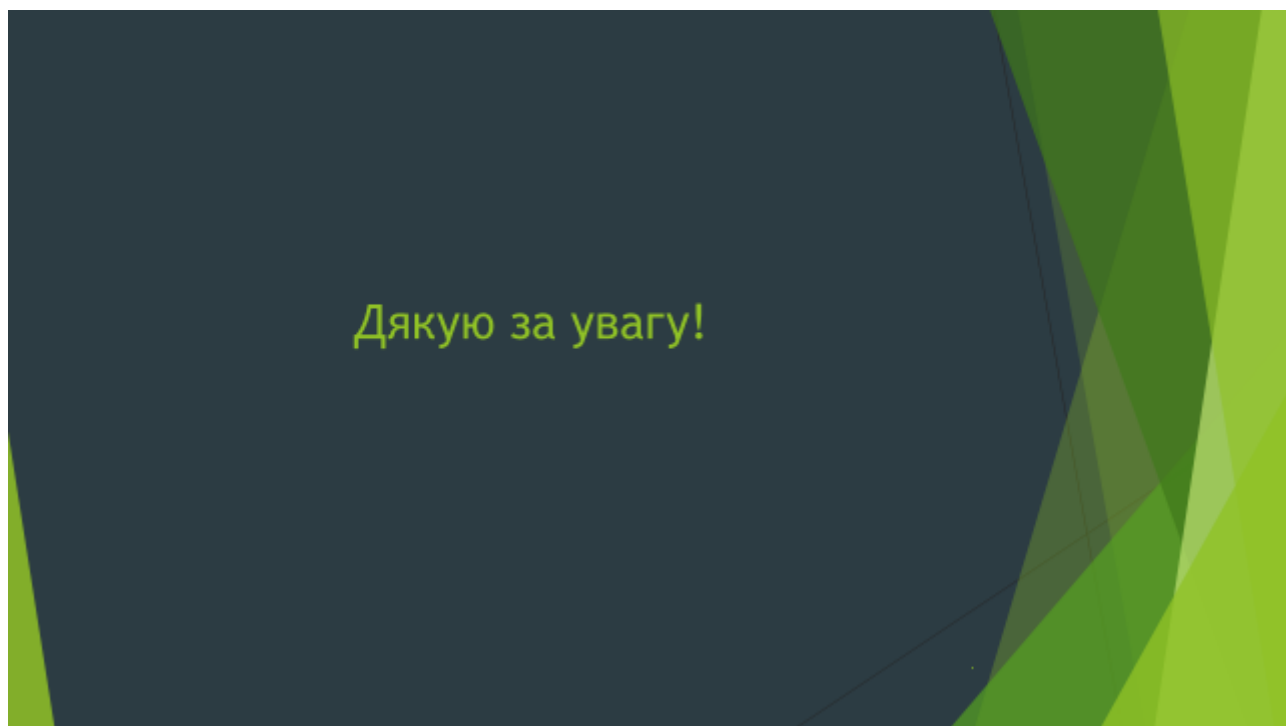


Рисунок Б.15 - Слайд 15