

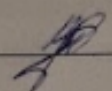
Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра обчислювальної техніки

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

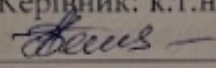
на тему:

**СИСТЕМА ПЕРСОНАЛІЗОВАНИХ РЕКОМЕНДАЦІЙ КОНТЕНТУ НА
ОСНОВІ ПОВЕДІНКОВИХ ДАНИХ КОРИСТУВАЧІВ ВЕБ-СЕРВІСУ**

Виконав: студент групи 2КІ-24м
спеціальності 123 – Комп'ютерна інженерія

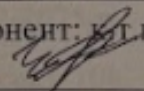
 Воронов В.В.

Керівник: к.т.н., доц. доцент каф. ОТ

 Войцеховська О.В.

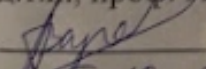
«12» 12 2025 р.

Опонент: к.т.н., доцент каф. ПЗ

 Чехместрук Р. Ю.

«12» 12 2025 р.

Допущено до захисту
Завідувач кафедри ОТ
д.т.н., проф. Азаров О.Д.


12.12.2025 р.

Вінниця 2025

ВІННИЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій та комп'ютерної інженерії

Кафедра обчислювальної техніки

Галузь знань — Інформаційні технології

Освітній рівень — магістр

Спеціальність — 123 Комп'ютерна інженерія

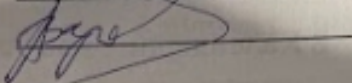
Освітньо-професійна програма — Комп'ютерна інженерія

ЗАТВЕРДЖУЮ

Завідувач кафедри ОТ

д.т.н., проф. О.Д. Азаров

« 25 » вересня 2025 р.



ЗАВДАННЯ

НА МАГІСТЕСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ

студенту **Воронову Владиславу Володимировичу**

1 Тема роботи: «Система персоналізованих рекомендацій контенту на основі поведінкових даних користувачів веб-сервісу», керівник Войцеховська Олена Валеріївна, к.т.н., доцент кафедри ОТ затверджено наказом вищого навчального закладу від «24» вересня 2025 року № 313.

2 Термін подання студентом роботи 04.12.2025 р.

3 Вихідні дані до роботи: датасет goodbooks-10k, що містить 6 мільйонів явних рейтингів за шкалою 1–5; метадані контенту (назви, автори, теги), сучасні технології розробки систем рекомендацій.

4 Зміст пояснювальної записки: вступ, огляд сучасного стану розвитку систем рекомендацій контенту, проєктування системи рекомендацій контенту, практична реалізація системи рекомендацій контенту на основі поведінкових даних користувачів, перевірка працездатності розробленої системи та економічна частина, висновки, перелік джерел посилання, додатки.

5 Перелік графічного матеріалу: блок схема алгоритму роботи системи персоналізованих рекомендацій контенту, функціональна схема системи персоналізованих рекомендацій, схема взаємодії основних програмних компонентів

6 Консультанти розділів роботи приведені в таблиці 1.

Таблиця 1 — Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
1-4	Войцеховська О. В. к.т.н., доц. каф. ОТ		
5	Адлер О. О. к.т.н., доц. каф. ЕПВМ		
Нормоконтроль	Швець С. І. асистент каф. ОТ		

7 Дата видачі завдання 25.09.2025 р.

8 Календарний план виконання МКР приведений у таблиці 2

Таблиця 2 — Календарний план

№ з/п	Назва етапів МКР	Термін виконання		Примітка
		початок	закінчення	
1	Постановка задачі роботи	25.09.2025	29.09.2025	
2	Огляд та порівняння існуючих технологій	30.09.2025	02.10.2025	
3	Визначення функціональних вимог до методу та проєктування	03.10.2025	05.10.2025	
4	Розробка алгоритмів роботи методу	06.10.2025	16.10.2025	
5	Програмна реалізація методу	17.10.2025	26.10.2025	
6	Створення моделі комп'ютерної мережі, тестування методу і виправлення помилок	27.10.2025	02.11.2025	
7	Розрахунок економічної частини	03.11.2025	04.11.2025	
8	Оформлення пояснювальної записки	05.11.2025	11.11.2025	
9	Попередній захист	12.11.2025	12.11.2025	
10	Перевірка якості виконання магістерської кваліфікаційної роботи та усунення недоліків	01.12.2025	16.12.2025	

Студент Воронов В. В.
Керівник роботи Войцеховська О. В.

АНОТАЦІЯ

УДК 004.8

Воронов В.В. Система персоналізованих рекомендацій контенту на основі поведінкових даних користувачів веб-сервісу. Магістерська кваліфікаційна робота зі спеціальності 123 — Комп'ютерна інженерія, освітня програма — Комп'ютерна інженерія. Вінниця: ВНТУ, 2025 — 90 с.

На укр. мові. Бібліогр.: 29 назв; рис.: 13, табл.: 20

У даній магістерській кваліфікаційній роботі розроблено гібридну систему персоналізованих рекомендацій контенту, що базується на поведінкових даних користувачів. Здійснено аналіз існуючих методів колаборативної фільтрації та обґрунтовано необхідність їх вдосконалення для подолання проблеми "холодного старту" і розрідженості даних. Вдосконалено метод матричної факторизації шляхом інтеграції контентних ознак для ініціалізації латентних факторів. Проведено експериментальне дослідження підтвердило, що розроблена система демонструє вищу точність та кращу повноту рекомендацій порівняно з базовими алгоритмами, що дозволяє підвищити ефективність веб-сервісу.

Ключові слова: система персоналізованих рекомендацій, колаборативна фільтрація, матрична факторизація, поведінкові дані.

ABSTRACT

Voronov V.V. Personalized Content Recommendation System Based on User Behavioral Data of a Web Service. Master's Qualification Thesis in Specialty 123 — Computer Engineering, Educational Program — Computer Engineering. Vinnytsia: VNTU, 2025

In Ukrainian. Bibliography: pages, sections, tables, figures, appendices, sources according to the list of references.

This master's thesis, a hybrid system for personalized content recommendations based on user behavioral data has been investigated and developed. An analysis of existing collaborative filtering methods was conducted, and the necessity of improving them to address the “cold start” problem and data sparsity was substantiated. The main part of the work introduces an enhanced matrix factorization approach by integrating content-based features for initializing latent factors. The experimental evaluation demonstrated that the proposed hybrid model achieves higher prediction accuracy and better recommendation recall compared to baseline algorithms, thereby increasing the overall effectiveness of the web service.

Keywords: personalized recommendation system, collaborative filtering, matrix factorization, behavioral data.

ЗМІСТ

ВСТУП	8
1 ОГЛЯД СУЧАСНОГО СТАНУ РОЗВИТКУ СИСТЕМ РЕКОМЕНДАЦІЙ КОНТЕНТУ	10
1.1 Огляд систем рекомендацій контенту та обґрунтування актуальності досліджень	10
1.2 Огляд сучасних підходів та алгоритмів для аналізу поведінки користувачів	14
1.3 Порівняльний аналіз існуючих алгоритмів для рекомендаційних систем	17
1.4 Огляд аналогів існуючих рекомендаційних систем	19
1.5 Порівняння архітектурних підходів	20
2 ПРОЄКТУВАННЯ СИСТЕМИ РЕКОМЕНДАЦІЙ КОНТЕНТУ	22
2.1 Математичне обґрунтування моделі матричної факторизації на основі сингулярного розкладу матриці	22
2.2 Вдосконалення матричної факторизації для ініціалізації латентних факторів	27
2.3 Техніки підготовки контентних ознак та їх інтеграція у факторну модель матричної факторизації	31
2.4 Розробка функціональної схеми системи персональних рекомендацій	33
2.5 Визначення метрик для оцінювання якості рекомендаційної моделі	34
2.6 Мінімальні вимоги до апаратного забезпечення.....	37
3 ПРАКТИЧНА РЕАЛІЗАЦІЯ СИСТЕМИ РЕКОМЕНДАЦІЙ КОНТЕНТУ НА ОСНОВІ ПОВЕДІНКОВИХ ДАНИХ КОРИСТУВАЧІВ	38
3.1 Обґрунтування вибору стеку технологій для реалізації системи	38
3.2 Розробка програмного засобу системи персоналізованих рекомендацій контенту	41
3.3 Інтеграція моделі в REST-API.....	43
3.4 Гіперпараметрична оптимізація моделі за критерієм RMSE	44
4 ПЕРЕВІРКА ПРАЦЕЗДАТНОСТІ РОЗРОБЛЕНОЇ СИСТЕМИ	50
4.1 Оцінювання параметрів контентної та контекстної моделей	50

4.2 Оцінювання параметрів вдосконаленого методу матричної факторизації ...	57
4.3 Перевірка працездатності REST-API для отримання персоналізованих рекомендацій	61
5 ЕКОНОМІЧНА ЧАСТИНА	63
5.1 Проведення комерційного та технологічного аудиту персоналізованої системи рекомендації контенту на основі поведінкових даних користувачів веб-сервісу.	63
5.2 Розрахунок витрат на здійснення розробки персоналізованої системи рекомендації контенту на основі поведінкових даних користувачів веб-сервісу	64
5.3 Розрахунок економічної ефективності науково-технічної розробки персоналізованої системи рекомендації контенту на основі поведінкових даних користувачів веб-сервісу за її можливої комерціалізації потенційним інвестором	70
ВИСНОВКИ	76
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	78
ДОДАТОК А Технічне завдання	81
ДОДАТОК Б ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ	85
ДОДАТОК В Блок схема роботи алгоритму роботи системи персоналізованих рекомендацій контенту	86
ДОДАТОК Г Функціональна схема системи персоналізованих рекомендацій .	87
ДОДАТОК Д Схема взаємодії основних програмних компонентів	88
ДОДАТОК Е Лістинги коду	89

ВСТУП

Стрімкий розвиток цифрових технологій у сучасному суспільстві привів до наявності величезних обсягів інформації, доступної користувачам, яка постійно оновлюється, змінюється та збільшується в об'ємі. Збільшенню обсягів інформації також сприяє зростаюча кількість соціальних мереж, месенджерів, вебсайтів, платформ тощо. В такому безперервному потоці різноманітного контенту стає дедалі важче знайти потрібну інформацію, яка була б цікавою та корисною конкретному користувачу.

Актуальність теми дослідження зумовлена необхідністю інтеграції високоефективних систем персоналізованих рекомендацій у комерційні та інформаційні платформи. З розвитком технологій та систем машинного навчання, якісне алгоритмічне забезпечення стає критично важливим для забезпечення роботи комплексних систем, що збирають та обробляють великі обсяги розріджених даних. Інновації у даній області дозволяють створювати більш точні та повні системи рекомендацій, які здатні ефективно вирішувати проблему "холодного старту" для нового контенту, тобто проблему нестачі достатніх початкових даних.

Метою роботи є підвищення точності прогнозів та зменшення рекомендацій нерелевантного контенту через проблему "холодного старту" для користувача веб-сервісу шляхом ініціалізації латентних факторів елементів в контекстній моделі.

Для досягнення цієї мети необхідно вирішити такі **задачі**:

- провести порівняльний аналіз сучасних підходів та алгоритмів для прогнозування поведінки користувачів рекомендаційних систем;
- розробити функціональну схему системи персональних рекомендацій;
- розробити програмний застосунок для реалізації системи персоналізованих рекомендацій контенту;
- провести гіперпараметричну оптимізацію обраного алгоритму для визначення оптимальних параметрів навчання моделі;
- провести тестування розробленої системи персональних рекомендацій.

Для досягнення поставленої в роботі мети використовуються такі **методи дослідження**: системний аналіз, об'єктно-орієнтовані методи проектування, методи машинного навчання.

Об'єкт дослідження – процеси формування персоналізованих рекомендацій контенту на основі поведінкових даних користувачів веб-сервісу.

Предмет дослідження – методи та алгоритми моделей персоналізованих рекомендацій для користувачів веб-сервісу.

Наукова новизна роботи полягає в удосконаленні методу матричної факторизації, в якому, на відміну від існуючих, інтегровано контентний вектор для ініціалізації латентних факторів елементів, що дало можливість підвищити точність прогнозів і мінімізувати вплив проблеми "холодного старту".

Практичні результати даної роботи полягають у розробці окремого застосунку у форматі REST-API для можливості впровадження розробленої системи рекомендацій контенту у веб-сервіси та підвищення залучення користувачів у контент веб-сервісу.

Апробації результатів роботи здійснена в доповіді на міжнародній науково-практичній інтернет-конференції «Молодь в науці: дослідження, проблеми, перспективи (МН-2026)».

Матеріали роботи доповідались та опубліковувались [1]:

Інтеграція контентних ознак при ініціалізації латентних факторів в системах персоналізованих рекомендацій / Воронов. В. В., Войцеховська О. В. // Матеріали міжнародної науково-практичної інтернет-конференції «Молодь в науці: дослідження, проблеми, перспективи (МН-2026)» (Вінниця, 2026р.) — [Електронний ресурс]. — Режим доступу: <https://conferences.vntu.edu.ua/index.php/mn/mn2026/paper/viewFile/26781/21995>

1 ОГЛЯД СУЧАСНОГО СТАНУ РОЗВИТКУ СИСТЕМ РЕКОМЕНДАЦІЙ КОНТЕНТУ

1.1 Огляд систем рекомендацій контенту та обґрунтування актуальності досліджень

Системи персоналізованих рекомендацій — це програмні рішення, що автоматично визначають інтереси користувача та формують індивідуальний набір контенту на основі його поведінки та вподобань. Вони аналізують історію взаємодій, контентні характеристики об'єктів і схожість між користувачами, щоб передбачити, які елементи можуть бути найбільш релевантними. Такі системи покращують користувацький досвід, зменшують інформаційне навантаження та підвищують ефективність веб-сервісів за рахунок точного підбору рекомендацій [2].

Сфера застосування систем персоналізованих рекомендацій (СПР) є досить широкою і охоплює практично всі цифрові сегменти діяльності людини, в яких відбувається взаємодія користувача з великим обсягом інформаційного контенту. Якщо на початкових етапах розвитку веб-технологій СПР асоціювалися переважно з електронною комерцією (Amazon, eBay) та потоковими медіа (Netflix, Spotify), то сьогодні вони інтегровані у найрізноманітніші галузі — від освіти до медицини, від соціальних мереж до промислових інформаційних систем. У сучасному цифровому середовищі користувачі щодня стикаються з великими масивами інформації, і основною задачею стає не просто зберегти або обробити ці дані, а й забезпечити їх релевантну персоналізацію. Саме рекомендаційні системи виступають посередником між користувачем та інформаційним потоком, дозволяючи адаптувати контент під індивідуальні інтереси користувачів, їх попередню поведінку, часовий або контекстуальний фактор [3]. Структурну схему рекомендаційної системи наведено на рис. 1.1.

Сучасна галузь СПР охоплює такі прикладні області, як: електронна комерція, яка дозволяє персоналізувати товарні пропозиції; медіа-платформи з

рекомендації відео (YouTube), музики (Spotify, Deezer) або фільмів (Netflix); соціальні мережі та новинні агрегатори; освітні платформи; промисловість та виробництво; сферу охорони здоров'я [4].

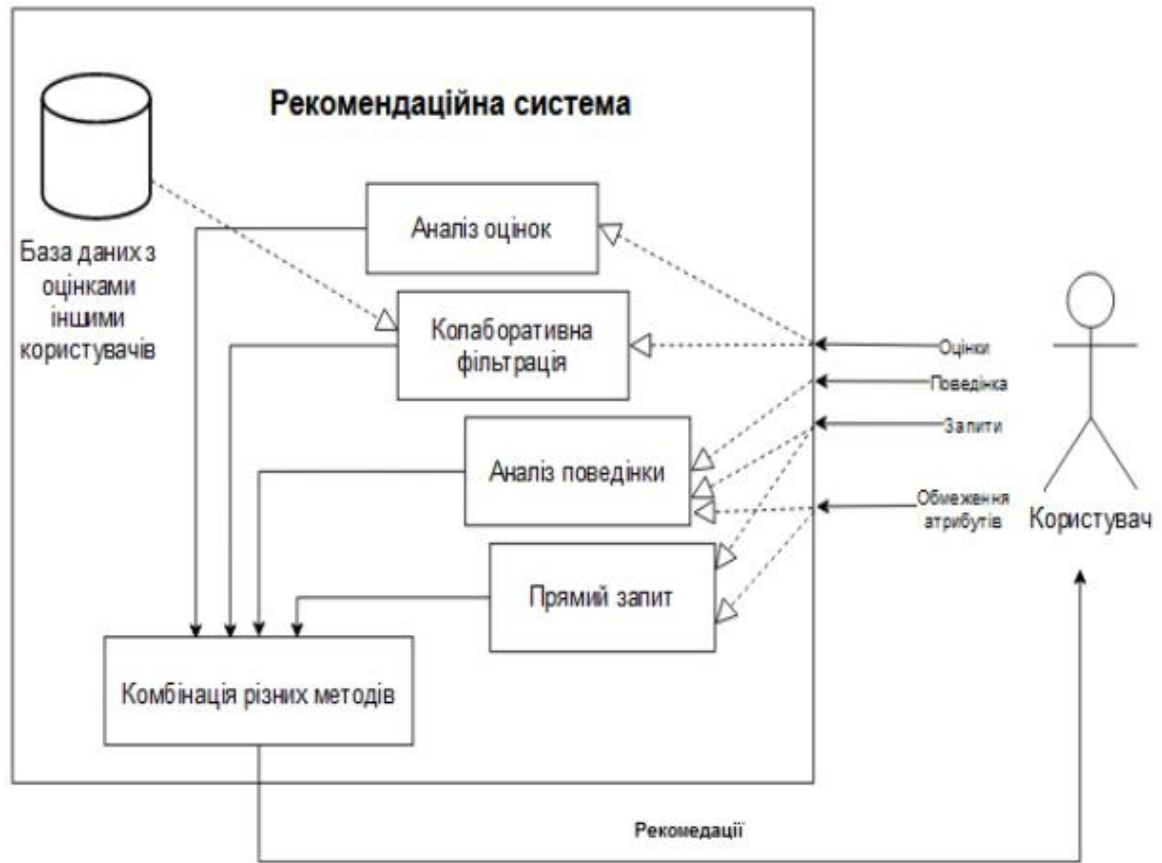


Рисунок 1.1 — Структурна схема рекомендаційної системи

Таким чином, рекомендаційні системи перестали бути винятково маркетинговим інструментом а стали основним елементом користувацького досвіду та механізмом монетизації цифрових платформ.

Здатність системи прогнозувати, що саме користувач захоче побачити наступним, є головною задачею будь-якого сучасного веб-сервісу.

На сьогодні класичні алгоритми вже не справляються з обсягом, різноманіттям і динамікою зміни сучасних даних, тому стикаються з різними труднощами. Існує проблема розрідженості даних, яка полягає в тому, що в більшості реальних систем частка заповнених рейтингів становить менше 1 % від можливої кількості оцінок у матриці зв'язки користувачів з об'єктами. Це призводить до втрати інформації про подібність і знижує якість рекомендацій.

За оцінками McKinsey, до 35 % загального прибутку компанії Amazon формується завдяки власній системі рекомендацій, а Netflix щороку економить понад 1 млрд доларів [5], утримуючи користувачів завдяки точним передбаченням контенту, який їм сподобається. Приріст кількості підписників наведений на рис. 1.2 та вказує на зацікавленість користувачів в контенті.

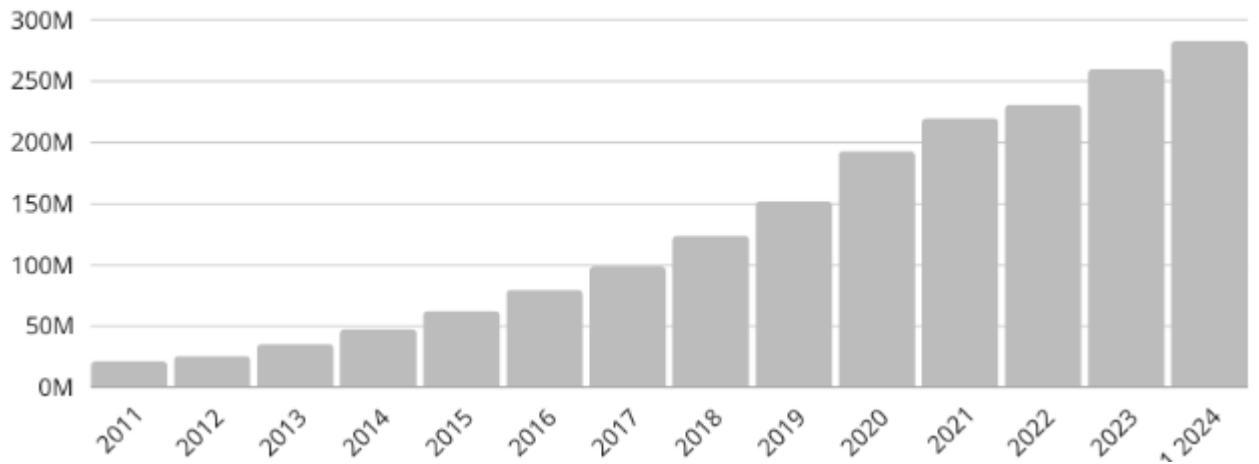


Рисунок 1.2 — Діаграма зростання кількості підписників Netflix на рік

Існує проблема розрідженості даних, яка полягає в тому, що в більшості реальних систем частка заповнених рейтингів становить менше 1 % від можливої кількості оцінок у матриці зв'язки користувачів з об'єктами. Це призводить до втрати інформації про подібність і знижує якість рекомендацій.

Ще одна проблема “холодного старту”, сутність якої полягає в тому, що нові користувачі або новий контент не мають історії взаємодій, як наслідок це унеможливорює застосування класичних колаборативних методів [6]. Третя проблема полягає у високих вимогах до точності та повноти рекомендацій. У сучасних платформах помилка рекомендацій безпосередньо впливає на прибуток і лояльність користувача, тому СПР повинні забезпечувати мінімальне відхилення та високу повноту прогнозу для збереження продуктивності.

Крім технічних викликів, існує й бізнес-необхідність у побудові моделей, які дозволяють відстежити причини формування тієї чи іншої рекомендації. Це особливо актуально для сфер, пов'язаних із фінансами, медициною або

юридичними послугами.

Реалізація таких систем не лише підвищує задоволеність користувача, а й безпосередньо впливає на показники утримання, середній час сесії та конверсію у цільові дії, що робить їх одним із ключових інструментів росту цифрового бізнесу. Таким чином, розробка гібридної програмно-алгоритмічної системи персоналізованих рекомендацій є актуальним і перспективним напрямом прикладних досліджень, що має як теоретичне, так і практичне значення для підвищення ефективності інформаційних платформ [7].

Проблеми, наявні для рекомендаційних систем, наведено у таблиці 1.1.

Таблиця 1.1 — Проблеми для систем побудови рекомендацій

Назва проблеми	Опис
Проблема рейтингів	Велика кількість користувачів не оцінюють товари, тим самим створюючи проблему, коли неможливо дізнатися чи сподобався товар чи ні.
Проблема холодного старту	Коли в системі з'являється новий користувач чи товар, у нього може бути або повністю відсутня інформація, або неповнота даних. Тому система не може надати рекомендації порівнюючи користувачів або товари.
Проблема розрідженості вхідних даних	Розрідженість рейтингової матриці, через те, що користувачі не виставляють оцінки товарам, приводить до неточних рекомендацій при використанні колаборативної фільтрації.
Проблема масштабованості	Через збільшення користувачів в системі, сповільнюється робота рекомендаційної системи.
Проблема зміни даних з часом	Старіння даних щодо товарів в рекомендаційній системі призводить до формування неточних рекомендацій.
Проблема зміни уподобань користувача	У деяких користувачів з часом може змінитися уподобання, через що вони почнуть отримувати неперсоналізовані рекомендації.
Проблема “синонімізації”	Виникає тоді коли товари що наявні в системі мають однакову назву чи ознаки, тому система може їх плутати між собою.
Проблема надмірної спеціалізації	Виникає якщо об'єкти, що рекомендуються користувачеві, занадто схожі один на інший.

1.2 Огляд сучасних підходів та алгоритмів для аналізу поведінки користувачів

Системи персоналізованих рекомендацій стали одним із ключових напрямів досліджень у галузі машинного навчання, обробки даних і штучного інтелекту. Вони дозволяють автоматизувати процес підбору контенту, товарів або послуг під конкретного користувача, виходячи з його минулих дій або характеристик самого контенту. Виділяють три основні види рекомендаційних систем: колаборативна фільтрація, контентна фільтрація та гібридні підходи (рис.1.3).

Кожен із них має власні теоретичні основи, типові алгоритми, переваги та практичні обмеження [8].



Рисунок 1.3 — Види систем персоналізованих рекомендацій

Колаборативна фільтрація — найстаріший і найпоширеніший підхід до рекомендацій. Його ідея полягає в тому, що користувачам, які мають схожі вподобання в минулому, можна рекомендувати подібний контент у майбутньому [9]. Іншими словами, система «вчиться» на основі взаємодій усіх користувачів і формує рекомендації з урахуванням колективної поведінки. Графічне представлення принципу роботи такого методу подано на рис. 1.4.



Рисунок 1.4 — Робота методу колаборативної фільтрації по характеристиках користувача

У межах колаборативної фільтрації існують два основні напрями: орієнтований на характеристики користувача, тобто коли рекомендації формуються на основі схожості між користувачами, і орієнтований на характеристики контенту, тобто коли аналізується схожість між самими об'єктами.

Перевага колаборативної фільтрації полягає в простоті реалізації та незалежності від опису контенту. Алгоритму не потрібно знати, про що фільм або який жанр має книга — достатньо даних про оцінки чи перегляди користувачів. Саме тому колаборативна фільтрація є універсальним рішенням для платформ, які збирають великі обсяги поведінкових даних, але мають обмежені метадані про об'єкти. Однак вона має суттєві обмеження, такі як відсутність великої кількості оцінок контенту, оскільки в реальних системах більшість користувачів реагують лише на невелику частку об'єктів, а також це те, що нові користувачі або новий контент не мають історії, тому алгоритм не може запропонувати їм нічого релевантного.

Попри ці недоліки, такий метод фільтрації залишається еталонним і

використовується для порівняння ефективності нових моделей. Зокрема, алгоритми на основі колаборативної фільтрації заклали основу відомого конкурсу Netflix Prize, який від 2006 по 2009 рік стимулював розвиток сучасних підходів до рекомендацій.

Контента фільтрація використовує зовсім іншу логіку. У центрі уваги не поведінка інших користувачів, а характеристики самого контенту [10]. Система намагається зрозуміти, що саме подобається користувачу, виходячи з властивостей елементів, які він раніше оцінював або переглядав. Зображення принципу роботи такого типу на рис. 1.5. Наприклад, якщо користувач читав статті про машинне навчання, система проаналізує текст цих статей і запропонує йому інші матеріали, що містять подібні теми або терміни. У сфері електронної комерції такий підхід може базуватися на категоріях товарів, їхніх технічних характеристиках, ціні, бренді або описах.



Рисунок 1.5 — Робота методу колаборативної фільтрації по характеристиках контенту

Типовий процес побудови контентної рекомендаційної системи складається з таких етапів:

- формування для кожного об'єкта набору ознак, який складається з ключових слів, категорій, параметрів, коротких описів;
- перетворення ознак у числове представлення (вектори), при якому для

текстових даних найчастіше використовують метод TF-IDF, який визначає вагу кожного терміну залежно від його частоти в документі та у всьому блоці;

- формування вектору інтересів користувача на основі контенту, який він позитивно оцінив;

- пропозиція контенту, коли система вимірює схожість між профілем користувача та вектором кожного нового об'єкта і пропонує об'єкти з найбільшою подібністю.

Переваги контентного підходу: не потребує даних про інших користувачів; ефективно працює для нового контенту (відсутність проблеми “холодного старту” для об'єктів); забезпечує зрозумілість результатів, тобто можна чітко вказати, чому саме запропоновано певний об'єкт. Але також такий тип фільтрації має і недоліки. До них відноситься схильність рекомендувати надто схожі об'єкти, залежність від якості метаданих і відсутність колективного ефекту, тобто система не враховує поведінку спільноти користувачів. К

Контентний підхід широко використовується у наукових, освітніх та медійних платформах, де кожен елемент має багатий описовий набір ознак (наприклад, назва, жанр, текстовий опис, ключові слова).

1.3 Порівняльний аналіз існуючих алгоритмів для рекомендаційних систем

Існує два основних класи алгоритмів колаборативної фільтрації, що традиційно використовуються як базові моделі у сфері персональних рекомендацій: k-Neares Neighbors (k-NN), який використовує історичні дані та Singular Value Decomposition (SVD), який є основним інструментом модельно-орієнтованого підходу [11].

Алгоритми k-NN (на основі елементів або користувачів) є інтуїтивно зрозумілими і легко реалізуються за допомогою спеціалізованих бібліотек, таких як scikit-surprise або scikit-learn. До переваг таких алгоритмів можна віднести високу прозорість прогнозу, легкість інтерпретації і вони слугують надійним початковим еталоном для порівняння. Але є й суттєве обмеження – це

низька масштабованість та обчислювальна складність. У реальних веб-сервісах з мільйонами користувачів постійне переобчислення схожості робить k-NN непридатним для роботи в реальному часі. Крім того, k-NN неефективно працює з розрідженими матрицями, оскільки спільна множина елементів, оцінених двома користувачами, може бути занадто малою для обчислення достовірної схожості.

Алгоритм k-NN (на основі елементів або користувачів) є інтуїтивно зрозумілим і легко реалізовується за допомогою спеціалізованих бібліотек, таких як `scikit-surprise` або `scikit-learn`. До переваги такого алгоритму можна віднести високу прозорість прогнозу, легкість інтерпретації і вона є початковою еталонною моделлю для порівняння. Але є й обмеження – це низька масштабованість та обчислювальна складність. У реальних веб-сервісах з мільйонами користувачів постійне переобчислення схожості робить k-NN непридатним для роботи в реальному часі. Крім того, k-NN неефективно працює з розрідженими матрицями, оскільки спільна множина елементів, оцінених двома користувачами, може бути занадто малою для обчислення достовірної схожості.

Алгоритми SVD є ядром більшості сучасних СПР (наприклад, система Netflix Prize). Вони будують модель, яка ефективно працює з розрідженістю даних шляхом проектування їх у простір латентних факторів. Такі алгоритми відзначаються високою точністю прогнозу, відмінною масштабованістю, а також здатністю виявити приховані залежності у даних, які не очевидні для k-NN. Проте є й обмеження і основним тут є проблема "холодного старту". Оскільки SVD навчає латентні вектори на основі історичних рейтингів, новий контент (який не має рейтингів) отримує випадково ініціалізований латентний вектор. Це призводить до нерелевантних прогнозів, доки елемент не отримає достатню кількість поведінкових даних. SVD є неточним для нового контенту, що знижує показник повноти системи в цілому. Аналіз доводить, що жодна з базових моделей самотійно не здатна забезпечити оптимальний баланс між точністю прогнозу та здатністю рекомендувати новий контент. SVD

демонструє найкращий показник відхилення прогнозованих значень від реальних, але є непридатною для роботи з новим контентом, що призводить до низької повноти прогнозу для елементів з малою кількістю рейтингів. Модель k-NN є надто повільною для великих каталогів і не забезпечує достатньої точності [12]. Нижче наведені порівняльні характеристики параметрів кожного типу алгоритмів моделей в таблиці 1.3.

Таблиця 1.3 – Порівняння типів алгоритмів

Критерій порівняння	k-NN	SVD
Принцип	Обчислення схожості сусідства.	Латентні фактори.
Складність реалізації	Низька / Середня.	Середня / Висока.
Точність прогнозу	Середня, обмежена розрідженістю.	Висока, галузевий стандарт.
Масштабованість	Низька (потребує постійного перерахунку сусідства).	Висока (швидке прогнозування після навчання).
Вирішення "холодного старту"	Слабкий захист.	Не вирішує (проблема нових елементів).
Вимога до розробки	Базова модель для порівняння.	Основа для вдосконалення.

1.4 Огляд аналогів існуючих рекомендаційних систем

Сучасні бізнес-продукти намагаються надати своєму користувачеві найрелевантніший контент і саме системи персоналізованих рекомендацій дають таку можливість. Зокрема однією із найвідоміших систем рекомендацій є Netflix Recommendation Engine, який використовує матричну факторизацію у поєднанні з градієнтним бустингом та глибокими нейромережами. Основна мета — прогнозування рейтингів фільмів на основі історії переглядів. Netflix впроваджує багаторівневу архітектуру, де моделі факторизації працюють паралельно з контентними і контекстними фільтрами (час, країна, жанр).

Не менш відомою є СПР YouTube Recommender System, яка базується на глибокій двоступеневій архітектурі: модель candidate generation формує пул потенційно релевантних відео, а модель ranking відсортовує їх за персональною релевантністю [13]. Поведінкові сигнали (тривалість перегляду, лайки, коментарі) перетворюються у вектори ознак користувача.

Такі системи також впроваджують не тільки у стримінгові платформи, а і у маркетплейсах наприклад система рекомендацій Amazon Personalize застосовує гібридний підхід: SVD-подібні моделі поєднуються з імовірнісними байєсівськими методами та аналізом текстових описів товарів. Це дозволяє платформі працювати ефективно навіть при низькій кількості історичних рейтингів.

Також їх застосування є в музичних сервісах на прикладі Spotify. Інтегрована в цей сервіс СПР використовує методи контентної фільтрації із векторизацією музичних треків за аудіоспектральними ознаками у поєднанні з моделями факторизації на поведінкових плейлистах. Також такі сервіси, як LinkedIn та Airbnb застосовують гібридні системи з урахуванням профільних даних користувача (посада, професійні інтереси, місце розташування) у комбінації з історією дій. Це демонструє, що рекомендаційні системи зазвичай поєднують контентні, колаборативні та контекстні методи. Чисті алгоритми k-NN або класичний SVD у них виступають не як кінцеві, а як проміжні модулі в багатошаровій архітектурі.

1.5 Порівняння архітектурних підходів

На сьогодні сучасні системи рекомендацій базуються на трьох ключових архітектурних принципах, таких, як матрична факторизація, що найкраще підходить для великих розріджених даних, пошук сусідства (наприклад k-NN), яка має простоту для реалізації, але погано масштабується при збільшенні даних і гібридні підходи, які поєднують латентне моделювання з контентними векторами. Саме цей напрям показує найкращі результати у сценаріях “холодного старту”.

У промислових умовах більшість матриць користувачів і контенту є розрідженими від 95% до 99%. Це означає, що стандартні k-NN-моделі не можуть надійно обчислити подібність між користувачами через нестачу спільних елементів. У той же час матрична факторизація дозволяє скоротити простір вимірів і знайти приховані шаблони (патерни), але не справляється з

новим контентом. Тому доцільним є створення гібридного підходу, який поєднує факторизацію з контентною ініціалізацією векторів. Така модель не тільки підвищить точність прогнозів, але й дозволить скоротити час збіжності під час навчання.

На відміну від традиційної матричної факторизації, яка ініціалізує латентні вектори випадковими числами, пропонований гібридний підхід використовує контентні ознаки для осмисленої початкової ініціалізації латентних векторів елементів. Це дозволяє новому контенту одразу мати латентний вектор, який відображає його характеристики. У результаті отримуємо, що новий контент одразу може бути включений у рекомендації на основі його ознак. А якісна початкова ініціалізація прискорює процес навчання факторизації та покращує кінцевий показник середньоквадратичної похибки.

Таким чином, огляд існуючих реалізацій показав, що поєднання контентних та контекстних моделей є оптимальним рішенням для побудови системи персоналізованих рекомендацій контенту в умовах реального веб-сервісу.

2 ПРОЄКТУВАННЯ СИСТЕМИ РЕКОМЕНДАЦІЙ КОНТЕНТУ

2.1 Математичне обґрунтування моделі матричної факторизації на основі сингулярного розкладу матриці

У сучасних інформаційних системах машинне навчання посідає ключове місце, оскільки дозволяє автоматизувати аналіз великих обсягів даних, будувати прогностичні моделі та формувати персоналізований контент для користувачів. У межах машинного навчання розрізняють кілька підходів, що відрізняються способом отримання знань із даних та характером взаємодії алгоритму з навчальною вибіркою [14].

Навчання з учителем передбачає наявність маркованих даних, коли для кожного прикладу відомо правильне значення вихідної змінної. За такого підходу модель намагається відтворити залежність між входом і виходом, що дозволяє розв'язувати задачі класифікації та регресії [15]. До типових інструментів цього класу належать лінійні моделі, метод опорних векторів, дерева рішень, ансамблеві алгоритми, а також багаточарові нейронні мережі. На відміну від нього, навчання без учителя застосовується тоді, коли дані не містять міток. У цьому випадку завдання алгоритму полягає у виявленні прихованих структур, груп або закономірностей у даних. Найчастіше використовуються методи кластеризації, а також методи зниження розмірності, що дають змогу спростити дані та виокремити ключові ознаки. Такі підходи широко застосовуються у попередньому аналізі поведінкових даних користувачів, підготовці ознак для рекомендаційних систем та виявленні аномалій. Проміжним варіантом між двома попередніми підходами є напівконтрольоване навчання, у якому поєднуються марковані та немарковані приклади. Використання великого обсягу немаркованих даних дозволяє підвищувати точність моделі навіть за умов обмеженої кількості прикладів із відомими мітками.

Окремий клас становить підсилене навчання, що базується на взаємодії моделі зі середовищем. Алгоритм послідовно приймає рішення, отримує

винагороду та намагається максимізувати її у довгостроковій перспективі. Підсилене навчання активно застосовується у динамічних рекомендаційних системах, системах керування ставками та оптимізації контенту в реальному часі.

У контексті систем персоналізованих рекомендацій особливе значення мають методи, які працюють із взаємодіями користувачів та об'єктів. Колаборативна фільтрація може базуватися як на прямому аналізі схожості між користувачами або товарами (методи, що ґрунтуються на пам'яті, наприклад k-NN), так і на побудові латентних моделей, які виявляють приховані фактори вподобань (методи на основі моделі, зокрема сингулярний розклад матриці (SVD), чергова процедура найменших квадратів (ALS) чи факторизація невід'ємної матриці (NMF)) [16].

Контентні підходи, своєю чергою, використовують описові характеристики користувачів та об'єктів, включаючи текстові ознаки, векторизовані за допомогою методу перетворення тексту в числові ознаки чи інших методів.

Найефективнішими в реальних системах вважаються поєднання контентних та контекстних моделей, що поєднують обидва підходи й дозволяють компенсувати їхні слабкі сторони. Суттєву роль відіграють методи матричної факторизації, що ґрунтуються на розкладанні матриці взаємодій на два приховані простори — користувачів та об'єктів. Це дозволяє виявляти приховані закономірності вподобань та робити точні прогнози навіть за умов високої розрідженості даних. Система складається з кількох взаємопов'язаних логічних компонентів. Перший відповідає за попередню обробку даних, що включає очищення, нормалізацію та структурування інформації, а також перетворення текстових описів у числові вектори за допомогою TF-IDF. Наступний компонент реалізує навчання базових та поєднання контентної та контекстної моделей. Далі інтеграційний модуль поєднує результати контентного та факторизаційного підходів і забезпечує оцінювання якості роботи моделі за метриками RMSE, Precision та Recall. В той же час, хоча матрична факторизація

демонструє свою здатність ефективно працювати з розрідженістю, вона стикається з фундаментальним обмеженням, що стає конкретною науковою задачею — це вирішення проблематики “холодного старту” для контенту [17].

Останній компонент системи відповідає за опрацювання GET-запитів та повернення сформованих рекомендацій у форматі JSON. В контексті великих веб-сервісів, кількість фактичних взаємодій становить малий відсоток від усіх можливих взаємодій між користувачами та контентом. Ця розрідженість безпосередньо ускладнює роботу алгоритмів на основі пам'яті, оскільки значно знижує достовірність обчислення схожості (табл.2.1).

Таблиця 2.1 — Порівняння моделей у колаборативній фільтрації

Характеристика	k-Nearest Neighbors (k-NN)	Singular Value Decomposition (SVD)
Тип підходу	Memory-Based (пам'яті)	Model-Based (модельний)
Принцип роботи	Пошук схожих користувачів або об'єктів на основі відстані	Розкладання матриці на приховані фактори
Потреба в даних	Вимагає багато історичних даних	Працює краще при розріджених матрицях
Масштабованість	Низька, бо пошук схожих елементів дорогий	Висока, після навчання модель швидка
Точність	Середня, залежить від метрики	Вища за рахунок латентних факторів
Переваги	Простота, інтерпретованість	Висока якість прогнозу, узагальнення даних
Недоліки	Погано працює на великих наборах	Потребує часу на навчання моделі
Базовий рівень	Так	Так

Проблема виникає через те, що латентні вектори нового контенту повинні бути навчені на основі його взаємодій. Якщо елемент не має рейтингів, його латентний вектор ініціалізується випадковими значеннями. Це призводить до меншої точності моделі є високоточною для контенту з великою кількістю рейтингів, а також до того, що модель робить абсолютно нерелевантні прогнози для нового контенту, оскільки випадкова ініціалізація не має жодного зв'язку з

контентними ознаками елемента.

Таким чином, задача полягає у розробці методу, який дозволяє наділити латентні вектори смисловим наповненням, що базується на контентних ознаках, ще до початку процесу навчання, використовуючи принципи гібридизації. Це забезпечить обґрунтовану початкову точку для навчання моделі і дозволить новим елементам одразу брати участь у рекомендаціях на основі їхніх характеристик [18].

Поєднання контентної та контекстної моделей є процедурою поступового навчання, що дозволяє апроксимувати розріджену матрицю рейтингів R добутком двох низькорангових, щільних матриць, який представлено формулою 2.1:

$$R \approx P \cdot Q^T, \quad (2.1)$$

де $P \in \mathbb{R}^{U \times K}$ — включає латентні вектори користувачів;

$Q \in \mathbb{R}^{K \times I}$ — включає латентні вектори контенту;

U — кількість користувачів у системі;

I — кількість елементів контенту;

K — кількість латентних факторів;

T — транспонована матриця.

Показник K є гіперпараметром, який визначає розмірність латентного простору, тобто розмірність спільного простору користувачів і контенту, у якому відбувається їхнє узгодження. Прогноз рейтингу користувача для контенту формується на основі суми індивідуальних компонентів, що враховують як загальну тенденцію, так і особливості суб'єктів взаємодії наведеної в формулі 2.2

$$r_{u,i} = \mu + b_u + b_i + p_u^T q_i, \quad (2.2)$$

де μ — загальний середній рейтинг;

b_u — зміщення користувача;

b_i — зміщення контенту;

$p_u^T q_i$ — динамічна складова прогнозу.

Усі ці складові, крім μ , є параметрами моделі, яка буде навчатись емпірично на основі поведінкових даних.

Навчання моделі сингулярного розкладання матриці є покроковим процесом, спрямованим на мінімізацію різниці між прогнозованими та відомими рейтингами. Цей процес формалізується за допомогою функції втрат. Для запобігання перенавчанню, коли модель занадто точно відображає шум і похибки тренувального набору, до функції втрат додається регуляризація (L2-норма). Регуляризація штрафує модель за використання занадто великих значень для латентних векторів, забезпечуючи узагальнюючу здатність моделі. Функція втрат, яку мінімізує SVD модель, має

$$L(P, Q, b) = \sum_{(u,i) \in R} (r_{u,i} - \hat{r}_{u,i})^2 + \lambda \left(\sum_u (|p_u|^2 + b_u^2) + \sum_i (|q_i|^2 + b_i^2) \right)$$

де $r_{u,i}$ — фактичний рейтинг;

$\hat{r}_{u,i}$ — прогнозований рейтинг;

λ — коефіцієнт регуляризації, який є критичним гіперпараметром;

$|\dots|^2$ — квадрат L_2 — норми, що використовується для штрафу.

Мінімізація цієї функції здійснюється за допомогою стохастичного градієнтного спуску. На кожній ітерації модель обчислює похибку для одного рейтингу і відповідно коригує всі параметри в напрямку, протилежному градієнту. Швидкість і розмір цього коригуючого кроку визначається ще одним гіперпараметром — швидкістю навчання. Цей ітераційний процес, що триває протягом заданої кількості епох, забезпечує високу точність розкладу матриць. Проте, саме випадкова ініціалізація прихованих векторів на початковому етапі методу спричиняє проблему “холодного старту”, що слугує основою для розробки гібридного підходу.

2.2 Вдосконалення матричної факторизації для ініціалізації латентних факторів

Класична модель матричної факторизації характеризує високу точність прогнозування оцінок у випадках, коли система вже володіє достатньою кількістю поведінкових даних користувачів. Цей підхід ефективно відтворює латентні залежності між користувачами та елементами контенту, що відображають приховані переваги або спільні патерни споживання [19]. Проте ефективність такого методу різко знижується в умовах нестачі інформації — зокрема, при появі нових елементів або користувачів, що ще не мають історії взаємодії із системою. Він виникає через відсутність навчальних даних, на основі яких модель могла б оцінити параметри нових об'єктів або користувачів. У класичній реалізації методу сингулярного розкладу матриць латентні представлення об'єктів ініціалізуються випадковими значеннями, які під час навчання поступово уточнюються через ітерації стохастичного градієнтного спуску. Однак, якщо для нового елемента відсутні оцінки, градієнтний спуск не має підстав для оновлення цього вектора, і його значення залишаються випадковими. У результаті модель генерує нерелевантні рекомендації, що істотно знижує якість системи на початкових етапах її використання.

Отже, виникає необхідність у модифікації методу сингулярного розкладу матриць, який здатний ефективно ініціалізувати латентні представлення нових об'єктів без залучення історичних оцінок. Запропоновано провести заміну випадкової ініціалізації латентних факторів контенту на осмислену ініціалізацію, що базується на контентних ознаках елемента, дозволить підвищити початкову якість прогнозів для нового контенту та прискорити збіжність алгоритму стохастичного градієнтного спуску зменшивши кінцевий показник помилки середньоквадратичної похибки при одночасному зростанні метрики повноти прогнозу.

Рішення поєднання контентних та контекстних моделей поєднує переваги колаборативної та контентної фільтрації. На відміну від класичних контентних систем, які напряду прогнозують рейтинг на основі схожості контенту,

запропонований підхід використовує контентні ознаки лише для формування початкового знання моделі. Такий гібридний підхід належить до класу комбінацій та ініціалізацій ознак, які дозволяють покращити процес навчання шляхом використання апріорної інформації.

Початковим етапом реалізації гібридного підходу є створення формалізованого подання контенту у вигляді числових векторів. У реальних даних інформація про елементи часто має неструктурований характер: текстові описи, списки жанрів, теги або ключові слова. Для перетворення цих даних у придатний формат використовується класичний статистичний метод визначення ваги слова в документі відносно всього текстового масиву, який використовується для побудови текстових ознак у задачах обробки природної мови. На відміну від простих бінарних векторів або підрахунку слів такий метод забезпечує зважену оцінку важливості терміна. Це є критичним, оскільки терміни, які часто зустрічаються у всьому каталозі (наприклад, "роман", "книга"), мають низьку інформативну цінність. Такий метод підсилює вагу тих слів, які є унікальними для конкретного елемента контенту, що підвищує якість контентного вектора.

Поява терміна визначає, наскільки часто певне слово зустрічається в описі елемента. Цей показник дозволяє виявити ключові терміни, характерні саме для конкретного документа описана формулою:

$$TF_{t,i} = \frac{n_{t,i}}{\sum_k n_{k,i}}$$

де $TF_{t,i}$ — частота терміна;

$n_{t,i}$ — кількість входжень терміна t у документі i ;

$\sum_k n_{k,i}$ — загальна кількість термінів у цьому документі.

Такі слова, як “жанр”, “персонаж” або “сюжет” можуть зустрічатися у більшості описів і не нести специфічного змісту. Тому вводиться поняття оберненої частотності документа, яке зменшує вагу поширених слів, підвищуючи дискримінаційну здатність вектора. Отримані значення формують

розріджений вектор контентних унікальних термінів у всьому масиві документів. На практиці такі вектори мають високу розмірність (тисячі або десятки тисяч ознак), але лише невелика їх частина має ненульові значення. Саме ці вектори виступають проміжним поданням контенту, що слугує основою для подальшої інтеграції у модель сингулярного розкладу матриць.

Отже, створюється зв'язок між високорозмірним контентним простором ознак і низькорозмірним латентним простором факторів, який використовує модель. Це перетворення виконується за допомогою матриці трансформації. Таким чином, кожен новий елемент контенту може отримати осмислене латентне представлення навіть без наявності історії взаємодій користувачів. Якщо матриця трансформації була попередньо навчена (наприклад, на основі елементів з відомими оцінками), нові об'єкти автоматично успадковують релевантну структуру латентного простору. Перед запуском алгоритму замість випадкової генерації латентних векторів контенту, використовується множення його вихідного контентного вектора на матрицю лінійного перетворення, що проектує цей вектор у простір прихованих ознак. Це забезпечує змістовний старт навчання, де кожен елемент уже має інтерпретований вектор у латентному просторі. Подальше оновлення параметрів відбувається за правилами методу розкладання матриці взаємодій за формулою (2.2) [20]. Оновлення векторів p_u та q_i відбувається за правилами стохастичного градієнтного спуску описаного формулами:

$$p_u \leftarrow p_u + \gamma(\text{error}_{u,i} \cdot q_i - \lambda \cdot p_u)$$

$$q_i \leftarrow q_i + \gamma(\text{error}_{u,i} \cdot p_u - \lambda \cdot q_i)$$

де γ — швидкість навчання;

λ — коефіцієнт регуляризації;

p_u — латентний вектор користувача u (відповідає його прихованим уподобанням);

q_i — латентний вектор елемента i (відповідає характеристикам книги);

$\text{error}_{u,i}$ — помилка прогнозу.

Поєднання контентних та контекстних моделей має низку переваг, серед яких ефективно вирішення проблеми “холодного старту”, оскільки новий контент одразу отримує осмислений вектор, сформований на основі власних характеристик. Це також сприяє покращеній збіжності, адже початкова точка навчання розташована ближче до оптимального розв’язку, що дозволяє зменшити кількість епох, необхідних для стабілізації процесу факторизації. Додатково використання контентних ознак як «м’якої регуляризації» допомагає уникати перенавчання та локальних мінімумів, забезпечуючи кращу здатність моделі до узагальнення. Крім того, завдяки використанню трансформованої матриці латентні фактори отримують зв’язок із реальними контентними характеристиками, що полегшує інтерпретацію моделі.

Запропоноване поєднання контентних та контекстних моделей формує новий формат у сфері рекомендаційних систем, поєднуючи статистичну дієвість колаборативних методів із семантичною інформативністю контентних підходів, що підкреслює інноваційний характер проведеного дослідження. Блок-схема алгоритму роботи системи персоналізованих рекомендацій контенту подана на рис. В.1.

Процес починається з вибору гібридного підходу як основи формування рекомендацій. Далі модель перевіряє, чи є наявні історичні дані про користувача. Якщо користувач їх немає, то неможливо застосувати колаборативні методи. У цьому випадку система переходить до формування вектора ознак на основі контентної інформації, тобто використовує описові характеристики об’єктів для визначення початкових вподобань користувача. Контентний вектор слугує стартовою точкою для побудови рекомендацій. Якщо користувач не є новим, модель використовує історичні дані його взаємодій — рейтинги, перегляди, кліки чи інші сигнали, характерні для колаборативного підходу. На основі цих даних формується латентне представлення користувача, яке відображає приховані патерни його поведінки. Незалежно від гілки, яку обирає алгоритм, обидва підходи зрештою зводяться до побудови вектора рекомендацій, тобто набору релевантних об’єктів,

структурованих відповідно до очікуваних уподобань користувача. Такий механізм дозволяє гібридній моделі залишатися ефективною як у ситуаціях холодного старту, так і для користувачів із багатою історією активності.

2.3 Техніки підготовки контентних ознак та їх інтеграція у факторну модель матричної факторизації

Результативність поєднання контентної та контекстної моделей безпосередньо залежить від якості контентних ознак елементів. У реальних масивах даних інформація про контент (назви, описи, теги, жанри) існує у неструктурованій формі. Математичні моделі, вимагають числових векторів фіксованої довжини. Таким чином, необхідно здійснити процес векторизації, що є визначальним етапом у процесі інтеграції контентного простору в латентний простір.

Цей процес створює семантичний міст між текстовим описом і внутрішнім представленням моделі. Для вимірювання важливості слова у колекції документів (елементів контенту) було використано метод TF-IDF (частота терміна — обернена частота документа). Переваги методу TF-IDF полягають у тому, що на відміну від простих бінарних векторів, які лише фіксують присутність або відсутність слова, цей підхід дозволяє зважувати значущість термінів. Крім того, TF-IDF знижує вагу загальновживаних слів, що не несуть унікального семантичного навантаження, що, у свою чергу, забезпечує вищу дискримінаційну здатність отриманого вектора.

Процес векторизації є багатоетапним і потребує ретельної попередньої обробки для забезпечення чистоти та ефективності фінального вектора рекомендацій. Перш ніж застосовувати метод TF-IDF необхідно здійснити очищення та нормалізацію текстових даних для всього корпусу документа. Цей процес є критично важливим для зменшення розмірності фінального вектора ознак без втрати семантики. Застосовуються етапи очищення, такі як токенизація, нормалізація, видалення стоп-слів, лематизація.

За допомогою токенизації текст опису елемента контенту розбивається на окремі слова (токени), наступним кроком виконується нормалізація і всі токени переводяться у нижній регістр для забезпечення єдності. Після цього здійснюється видалення загальновживаних слів, які не несуть унікального семантичного навантаження (наприклад, "і", "в", "на", "це"). Наступним етапом відбувається лематизація — зведення слів до їхньої базової форми (наприклад, "рекомендую", "рекомендує" зводиться до "рекоменд"). Це запобігає тому, що одне і те ж слово у різних граматичних формах обробляється як різні ознаки. Частота терміна вимірює, наскільки часто певний термін зустрічається в описі елемента. Цей показник дозволяє виявити ключові терміни, характерні саме для конкретного документа, і є базовою складовою. Обернена частота терміну є контрвагою. Вона зменшує вплив термінів, які є загальновживаними в усьому каталозі контенту, оскільки такі терміни не мають високої дискримінаційної здатності, тобто вони не допомагають відрізнити один елемент від іншого. Остаточний зважений вектор контентних ознак для елемента отримується шляхом добутку двох незалежних величин — частоти термінів (TF) та оберненої частоти термінів (IDF). Цей вектор є числовим представленням елемента контенту і є єдиним джерелом інформації для поєднаних контентної та контекстної моделей щодо характеристики контенту.

Отриманий набір контентних векторів слугує основою для розрахунку початкових латентних факторів. Процес інтеграції забезпечує перехід контентної інформації у латентний простір, де працює модель рекомендації.

Початкова ініціалізація вирішує проблему «холодного старту» стратегічно, надаючи новому елементу суттєві властивості. Латентний вектор одразу семантично відповідає характеристикам контенту, що дає змогу системі формувати релевантні прогнози ще до першої взаємодії користувача. Крім того, модель розпочинає навчання алгоритму оптимізації факторизації з точки, яка є значно ближчою до глобального мінімуму, ніж випадкова. Також інтеграція контентної інформації виступає формою апіорного знання, що допомагає

моделі краще узагальнювати залежності та уникати локальних мінімумів, характерних для випадкової ініціалізації.

Таким чином, використання TF-IDF для створення вектора факторних ознак та його подальшої проекції є технологічною реалізацією наукової новизни, забезпечуючи більш надійний та ефективний старт для алгоритму системи рекомендацій контенту.

2.4 Розробка функціональної схеми системи персональних рекомендацій

Функціональна схема системи персональних рекомендацій відображає загальну логіку роботи програмного комплексу та взаємодію його ключових компонентів. На основі аналізу вимог до системи було сформовано архітектуру, яка забезпечує синхронну обробку даних користувача, інтеграцію різних методів рекомендацій та повернення результатів у веб-сервіс через REST-API. Центральним елементом схеми є модуль “Система персональних рекомендацій”, що об’єднує кілька функціональних блоків. Першим джерелом даних виступає база даних з оцінками та взаємодіями інших користувачів, яка використовується для модулів аналізу оцінок і колаборативної фільтрації. Саме ці підсистеми визначають подібність між користувачами або елементами та формують початковий прогноз релевантності.

Другий напрям опрацювання даних представлений модулем аналізу поведінки користувача. Він враховує індивідуальну історію переглядів, кліків, запитів або взаємодії з контентом, що дозволяє точніше адаптувати рекомендації до поточних інтересів. Додатково передбачено обробку прямих запитів, коли користувач ініціює пошук або надає явні критерії, що мають бути враховані під час побудови рекомендацій.

Усі результати з окремих блоків об’єднуються у модулі “Комбінація методів контекстних та контентних рекомендацій”. На цьому етапі виконується інтеграція сигналів, отриманих із колаборативної фільтрації, аналізу контенту та поведінкових даних. Такий підхід забезпечує стійкість моделі до

розрідженості даних та підвищує її здатність працювати у сценаріях холодного старту.

Після формування списку релевантних об'єктів система генерує підсумковий рекомендований контент, який передається до зовнішніх компонентів. Комунікація між рекомендаційною системою та веб-сервісом здійснюється через REST-API системи персоналізованих рекомендацій, що дозволяє програмно запитувати рекомендації, отримувати відповіді у стандартизованому форматі та інтегрувати результати у користувацький інтерфейс. На веб-сервісі відбувається фінальне відображення рекомендацій користувачу. Таким чином, розроблена функціональна схема описує повний цикл роботи системи — від збору даних та їх багаторівневого аналізу до інтеграції результатів у веб-середовище. Запропонована структура забезпечує модульність, розширюваність та можливість подальшого вдосконалення алгоритмів рекомендацій. Функціональна схема наведена на рис. Г.1.

2.5 Визначення метрик для оцінювання якості рекомендаційної моделі

При аналізі систем рекомендацій застосовують кілька груп метрик, які характеризують точність прогнозування та якість сформованих рекомендаційних списків. Процес оцінювання якості рекомендаційної системи має дві основні цілі, що включають перевірку коректності прогнозів рейтингу, тобто наскільки точно модель відтворює числові оцінки користувачів, та оцінювання релевантності рекомендаційного списку, тобто наскільки система здатна рекомендувати користувачу ті елементи, які дійсно йому подобаються. Для досягнення об'єктивності результатів усі експерименти виконуються з використанням перехресної валідації або класичного розбиття на тренувальну та тестову вибірки. Зазвичай використовують співвідношення 80/20.

Модель навчається на тренувальних даних, після чого для тестової множини обчислюються значення метрик середньоквадратичної похибки, точності та повноти [21]. Таке поєднання дозволяє оцінити модель як із точки зору прогнозування числових оцінок, так і з точки зору практичної користі

рекомендацій. Однією з найбільш поширених метрик для оцінювання моделей колаборативної фільтрації є середньоквадратична похибка (Root Mean Square Error, RMSE), яка вимірює середню величину відхилення між передбаченими системою рейтингами і фактичними оцінками користувачів, і визначається за формулою:

$$RMSE = \sqrt{\frac{1}{|R_{test}|} \sum_{(u,i) \in R_{test}} (r_{u,i} - \hat{r}_{u,i})^2}$$

де R_{test} — множина всіх пар користувач – елемент;

$r_{u,i}$ — реальна взаємодія користувача з контентом;

$\hat{r}_{u,i}$ — прогноз у взаємодії для користувача із контентом.

Чим менше значення RMSE, тим точніше модель відтворює уподобання користувачів. RMSE вважається стандартом при порівнянні алгоритмів колаборативної фільтрації, зокрема у конкурсах Netflix Prize, MovieLens, Kaggle Recommender Challenges тощо. Якщо, наприклад, шкала оцінювання має діапазон від 1 до 5, значення $RMSE = 0,8$, то це означає, що середнє відхилення прогнозу від реальної оцінки становить 0,8 бала. Таким чином, модель у середньому помиляється менш ніж на одну одиницю рейтингу, що вважається прийнятним результатом для більшості систем. Попри свою популярність, RMSE характеризується певними обмеженнями, зокрема не бере до уваги, у якому порядку система подає рекомендації. При сильній розрідженості матриці рейтинги, що відсутні, не враховуються, тому модель може мати низьке RMSE, але водночас видавати нерелевантні рекомендації та підходити лише для систем, що прогнозують числові значення. Для систем, орієнтованих на підбір топ-К елементів, потрібні інші метрики. Застосовуються метрики точності і повноти, які краще відображають практичну користь рекомендацій. На практиці користувачі не очікують від системи точного передбачення оцінки кожної книги чи товару. Їх цікавить отримання короткого, релевантного списку

рекомендацій, серед якого вони знайдуть дійсно цікаві елементи. Тому важливим є вимірювання якості таких списків.

Метрика точності Precision показує, яку частку з перших рекомендованих елементів користувач дійсно оцінив позитивно, та розраховується за формулою:

$$\text{Precision} = \frac{1}{|U|} \sum_{u \in U} \frac{|\text{Rec}_u \cap \text{Rel}_u|}{K}$$

де u — множина користувачів;

Rec_u — множина K елементів, рекомендованих користувачу;

Rel_u — множина елементів, які користувач реально оцінив як релевантні.

Отже, $\text{Precision} = 0,6$ означає, що 60% з поданих топ-10 рекомендацій дійсно цікаві користувачу. Метрика фокусується на якісній точності верхньої частини списку. У реальних сценаріях користувачі зазвичай переглядають лише від 5 до 10 рекомендацій, тому навіть незначне підвищення Precision має помітний вплив на задоволення користувача. Але метрика не враховує, скільки з усіх релевантних елементів було виявлено системою — лише частку у списку рекомендацій. Через це вона добре показує «чистоту» рекомендацій, але не дає повної картини про їх «повноту».

Метрика повноти Recall доповнює Precision і показує, яку частку з усіх релевантних елементів система змогла знайти в списку, та розраховується за формулою:

$$\text{Recall} = \frac{1}{|U|} \sum_{u \in U} \frac{|\text{Rec}_u \cap \text{Rel}_u|}{|\text{Rel}_u|}$$

де u — множина користувачів;

Rec_u — множина K елементів, рекомендованих користувачу;

Rel_u — множина елементів, які користувач реально оцінив як релевантні.

Recall = 0,4 означає, що система змогла відшукати 40% усього контенту, який користувач оцінює високо. Високий показник Recall вказує на «повне» охоплення інтересів користувача, навіть якщо частина рекомендацій виявиться помилковою. Precision і Recall часто знаходяться у компромісних відносинах. Підвищення точності зазвичай зменшує повноту, і навпаки.

Підсумковий критерій верифікації поєднання контентної та контекстної моделей встановлюється як перевищення показників базової моделі за такими параметрами, як точність прогнозу та якість рекомендації.

2.6 Мінімальні вимоги до апаратного забезпечення

Для можливості якісного виконання задачі побудови системи персоніфікованих рекомендацій на основі поведінкових даних користувача веб-сервісу необхідні мінімальні технічні вимоги наведені у таблиці 2.2.

Таблиця 2.2 — Мінімальні вимоги до апаратного забезпечення

Параметр	Мінімальні вимоги	Рекомендовані вимоги
Процесор	2 ядра (2.0 ГГц)	4 ядра (від 2,5 до 3,0 ГГц)
Оперативна пам'ять	8 ГБ	16 ГБ
Накопичувач	HDD 20 ГБ	SSD 50 ГБ
Операційна система	Linux Ubuntu 20.04 / Windows 10 (64-bit)	Linux Ubuntu 22.04 / Windows 11 (64-bit)

В переліку вимог надані мінімальні вимоги до апаратного забезпечення, але варто брати до уваги саме рекомендовані, оскільки саме при такому наборі параметрів зберігається стабільність навчання моделі, опрацювання даних, а також подальша інтеграція у веб-сервіс.

3 ПРАКТИЧНА РЕАЛІЗАЦІЯ СИСТЕМИ РЕКОМЕНДАЦІЙ КОНТЕНТУ НА ОСНОВІ ПОВЕДІНКОВИХ ДАНИХ КОРИСТУВАЧІВ

3.1 Обґрунтування вибору стеку технологій для реалізації системи

Вибір мови програмування Python для реалізації гібридної системи персоналізованих рекомендацій пояснюється її ключовою роллю в галузі обробки даних, машинного навчання та наукових обчислень [22].

Переваги мови Python проявляються у декількох ключових аспектах. Python пропонує готові, високооптимізовані бібліотеки для роботи з даними, що є критично важливим для моделювання та побудови рекомендаційних систем. Завдяки зрозумілому та лаконічному синтаксису забезпечується висока швидкість написання коду й легкість його підтримки, що відповідає вимогам до сучасної розробки програмного забезпечення. Значна спільнота розробників і наявність інструментів для паралельних обчислень створюють можливість ефективного опрацювання великих обсягів даних і побудови масштабованих рішень. Використання останньої стабільної версії Python гарантує підвищену продуктивність, удосконалені механізми обробки помилок та сумісність із новітніми версіями зовнішніх бібліотек, що є необхідною умовою для реалізації сучасного програмного проєкту.

Як основне середовище розробки використано PyCharm IDE, яке забезпечує повноцінну інтегровану платформу з підтримкою рефакторингу коду, налагодження, інтеграції із системами контролю версій та роботи з віртуальними середовищами, що відповідає професійним стандартам розробки.

Для ефективної реалізації функціональної складової гібридної системи рекомендацій необхідне використання спеціалізованих бібліотек, оптимізованих для роботи з числовими обчисленнями та моделями машинного навчання. Мова програмування Python налічує в собі ряд бібліотек, які мають високу продуктивність та призначені для вирішення задач машинного навчання. Характеристики технологій та найпоширеніших інструментів, які взяті для розробки системи персональних рекомендацій наведені в

таблиці 3.1 [23].

Таблиця 3.1 — Переваги технологій та інструментів

Технології	Ключові переваги
Python 3.12	Швидкість розробки, широкий набір ML(машинне навчання)/DS(наука про дані) інструментів, чистий синтаксис.
scikit-surprise	Готова оптимізована реалізація SVD, вбудована крос-валідація.
Pandas/NumPy	Ефективна обробка розріджених даних, високошвидкісні матричні операції.
scikit-learn	Реалізація TF-IDF та інших інструментів для обробки ознак.
pickle	Серіалізація та десеріалізація Python-об'єктів, що дозволяє зберігати навчену модель та швидко завантажувати її без повторного навчання.
fastapi	Високопродуктивний фреймворк для створення REST-API, забезпечує інтеграцію рекомендаційної моделі як веб-сервісу та автоматичну документацію (Swagger / OpenAPI)

Бібліотека Surprise (scikit-surprise) є спеціалізованою бібліотекою для побудови та оцінки систем рекомендацій, що базуються на рейтингах. Вона ідеально підходить для реалізації колаборативної фільтрації. Вона надає готові, високоефективні реалізації базових алгоритмів, які ми використовуємо для порівняння та як основу для поєднаних контентної та контекстної моделі. Її архітектура дозволяє легко модифікувати алгоритми (наприклад, змінювати процес ініціалізації латентних факторів). Крім того, дана бібліотека має вбудовані функції для пошуку оптимальних гіперпараметрів за допомогою перебору по сітці та крос-валідації, що критично важливо для проведення експериментів.

Бібліотеки Pandas та NumPy використовуються для ефективної попередньої обробки та маніпуляції великими обсягами даних (датасети ratings.csv, books.csv). Її структури DataFrame дозволяють швидко виконувати фільтрацію, агрегацію та об'єднання даних, необхідні для підготовки вхідних даних для Surprise.

Бібліотека scikit-learn використовується для реалізації контент-базованої частини гібридного підходу. Вона надає високоефективну реалізацію алгоритму TF-IDF, який необхідний для перетворення текстових метаданих контенту у

числові вектори ознак.

Зазначені можливості безпосередньо інтегруються в загальну архітектуру обробки даних, структуру якої подано на рис. Д.1, де представлено послідовний процес обробки даних у гібридній системі рекомендацій. Вхідні дані спочатку завантажуються та опрацьовуються за допомогою бібліотеки Pandas, яка забезпечує очищення, трансформацію та підготовку табличних структур. На наступному етапі до підготовлених текстових полів застосовується метод TF-IDF, що перетворює контент у числові векторні представлення, відображаючи важливість слів у межах описів елементів [24]. Завершальним етапом є передача цих векторизованих ознак до алгоритму SVD, реалізованого в бібліотеці Surprise, який здійснює матричну факторизацію та формує персоналізовані рекомендації. Така структура забезпечує логічний ланцюг переходу від сирих даних до побудови моделі рекомендацій.

Вибір технологічного стека для реалізації системи персоналізованих рекомендацій визначався не лише поширеністю інструментів, але й на вимогах до продуктивності, розширюваності та сумісності. Python 3.12 було обрано через його тісну інтеграцію з інструментами обробки даних, машинного навчання, а також активну підтримку з боку спільноти дослідників рекомендаційних систем. Додатково для забезпечення стабільності розробки використано віртуальні середовища (venv), що дозволяють уникнути конфліктів між версіями бібліотек. Середовище PyCharm Professional забезпечує не лише можливість налагодження та профілювання коду, а й інтеграцію з Git, що дає змогу вести історію змін моделі та відстежувати результати експериментів [25]. Для візуалізації процесів навчання та порівняння метрик використовувалася бібліотека Matplotlib, яка дозволяє формувати наочні графіки збіжності моделей, динаміки RMSE і точності рекомендацій. Оскільки система працює з розрідженими матрицями, було приділено особливу увагу ефективності пам'яті. Використання numpy.sparse структур та оптимізованих функцій Pandas дозволило зменшити обсяг оперативної пам'яті, необхідний для обробки даних, приблизно від 3 до 4 разів порівняно з повними матрицями. Додатково, для

контролю якості коду, проєкт супроводжувався unit-тестами, що перевіряли: коректність формування матриць користувач–об’єкт, відсутність пропусків у вхідних даних, узгодженість розмірностей при інтеграції TF-IDF та методу розкладання матриці взаємодій. Таким чином, вибрані технології забезпечили повний цикл розробки: від збору даних і попередньої обробки до навчання моделей і збереження результатів. Це дає змогу надалі масштабувати рішення або інтегрувати його в реальний веб-сервіс через REST API.

3.2 Розробка програмного засобу системи персоналізованих рекомендацій контенту

Розроблений програмний засіб реалізує гібридну архітектуру рекомендаційної системи, що поєднує два основні підходи: контент-базований — для формування векторів ознак контенту і колаборативний — для побудови латентного простору користувачів і об’єктів [26]. Система має дворівневу структуру. На нижньому рівні відбувається моделювання даних, де формується латентне представлення як контентних, так і колаборативних ознак. На цьому етапі здійснюється попередня обробка вхідних даних. На верхньому рівні здійснюється генерація та оцінка рекомендацій, а також додатково проводиться аналіз якості отриманих рекомендацій. Початкові дані представлено у вигляді двох таблиць:

- ratings.csv, яка містить пари користувач – об’єкт – оцінка (рис. 3.2);
- books.csv, яка включає метадані об’єктів (рис. 3.3).

	book_id	user_id	rating
0	1	314	5
1	1	439	3
2	1	588	5
3	1	1169	4
4	1	1185	4
5	1	2077	4
6	1	2487	4
7	1	2900	5
8	1	3662	4
9	1	3922	5

Рисунок 3.2 — Датасет рейтингів книг від користувачів

	id	...	small_image_url
0	1	...	https://images.gr-assets.com/books/1447303603s...
1	2	...	https://images.gr-assets.com/books/1474154022s...
2	3	...	https://images.gr-assets.com/books/1361039443s...
3	4	...	https://images.gr-assets.com/books/1361975680s...
4	5	...	https://images.gr-assets.com/books/1490528560s...
5	6	...	https://images.gr-assets.com/books/1360206420s...
6	7	...	https://images.gr-assets.com/books/1372847500s...
7	8	...	https://images.gr-assets.com/books/1398034300s...
8	9	...	https://images.gr-assets.com/books/1303390735s...
9	10	...	https://images.gr-assets.com/books/1320399351s...

Рисунок 3.3 — Датасет з метаданими об’єктів

На етапі підготовки даних було здійснено комплекс попередніх операцій, спрямованих на забезпечення коректності та придатності даних для подальшого моделювання. Зокрема, виконано перевірку цілісності датасету з метою виявлення некоректних або неповних записів, після чого проведено видалення дублікатів і пропусків, що могли б спотворити результати навчання моделі. Ідентифікатори користувачів та об’єктів були приведені до уніфікованого формату для забезпечення їх узгодженості в межах усієї системи. Дані було розподілено на тренувальну (80%) і тестову (20%) вибірки, що дало змогу правильно оцінити генералізуючу здатність моделі. Завершальним етапом стало формування матриці «користувач–об’єкт», яка слугує структурною основою для подальшого навчання моделей рекомендацій.

Для текстових полів (описів контенту) застосовано метод TF-IDF, який дозволяє перетворити неструктуровані текстові дані на формалізоване числове представлення. Отримані векторні представлення контенту надалі використовувалися як основа для формування контентної моделі, забезпечуючи початкову семантичну структуру елементів перед інтеграцією з колаборативними даними. Початковим кроком було побудовано та проаналізовано ефективність двох базових моделей: модель матричної факторизації (рис. 3.4) та модель на основі k-найближчих сусідів.

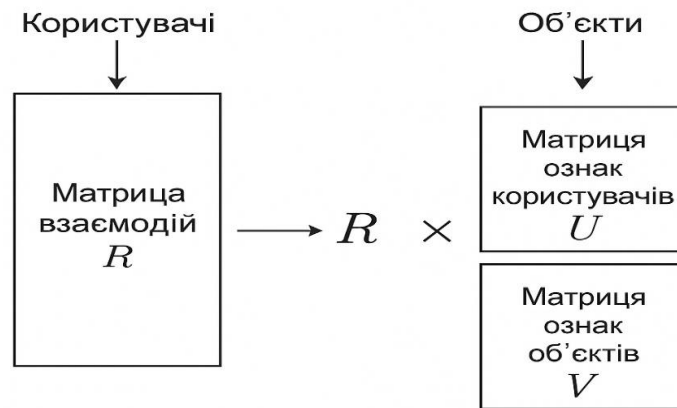


Рисунок 3.4 — Модель матричної факторизації

Навчання моделей виконувалося з використанням механізму крос-валідації, що дозволяє оцінити стабільність результатів. Для моделі проводилася оптимізація гіперпараметрів — кількості факторів, швидкості навчання та коефіцієнтів регуляризації. Головним етапом практичної реалізації є створення поєднаних контентної та контекстної моделей. Її суть полягає в тому, що для нових або малореєтингових об'єктів початкові латентні вектори не ініціалізуються випадково, а формуються на основі контентних ознак, отриманих із TF-IDF представлення. Таким чином, новий контент відразу має осмислене векторне представлення у латентному просторі, що усуває проблему холодного старту і прискорює процес збіжності під час навчання. У процесі інтеграції об'єднуються два потоки даних: поведінкові (рейтинги користувачів) і контентні (описові вектори об'єктів). Після об'єднання даних модель донавчається, коригуючи фактори користувачів і об'єктів до досягнення мінімальної похибки на тестовій вибірці. Фрагменти програмного коду, які реалізують кожен з етапів системи, наведено у Додатку Е.

3.3 Інтеграція моделі в REST-API

Важливим блоком побудови системи рекомендацій є її інтеграція в веб-сервіс. Найчастіше метод інтеграції відбувається саме через REST-API. Саме такий метод інтеграції дозволяє впроваджувати СПР модульно і стабільно використовувати систему персональних рекомендацій. Сервіс API приймає get-

запит в якому передаються 2 параметри: `user_id` - ідентифікатор користувача, під якого будуть формуватись рекомендації та параметр `n`, який відповідає за кількість рекомендованих об'єктів, які будуть віддаватись у відповідь.

Після отримання запиту сервер завантажує попередньо збережену модель, сформовану за допомогою бібліотеки Scikit-Surprise, та виконує прогнозування індивідуальних рейтингів для вибраного користувача. Отримані значення сортуються за спаданням прогнозованої релевантності, після чого формується фінальний список рекомендацій.

Результат роботи API повертається у форматі JSON, що забезпечує зручну інтеграцію з будь-якими клієнтськими застосунками — веб-інтерфейсом, мобільними додатками або іншими мікросервісами. До складу відповіді можуть входити: ідентифікатор елемента, назва, опис, прогнозований рейтинг та інші атрибути, що використовуються на рівні бізнес-логіки.

Важливою перевагою обраного підходу є те, що REST-API дозволяє виконувати обробку запитів у реальному часі, забезпечуючи мінімальні затримки при генеруванні рекомендацій. Крім того, завдяки використанню фреймворку FastAPI створюється можливість автоматичної документації через OpenAPI/Swagger, що значно полегшує тестування, розгортання та подальшу підтримку сервісу. Таким чином, поєднання контентної та контекстної моделей через REST-API забезпечує гнучкість, масштабованість і сумісність системи з сучасними архітектурними підходами, що робить її придатною для використання у реальних веб-платформах та високонавантажених сервісах.

3.4 Гіперпараметрична оптимізація моделі за критерієм RMSE

Якість будь-якої факторизаційної моделі істотно залежить від правильного налаштування її гіперпараметрів — значень, які визначають структуру моделі, швидкість навчання, глибину регуляризації тощо. Для підвищення ефективності було проведено процес гіперпараметричної оптимізації з використанням методу Grid Search за критерієм мінімізації показника RMSE.

Мета оптимізації — знайти таку комбінацію параметрів моделі SVD, яка забезпечить мінімальне середнє значення RMSE на крос-валідації, тобто найвищу точність передбачення рейтингів користувачів.

У бібліотеці Scikit-Surprise реалізовано вбудований механізм пошуку за сіткою, який дозволяє автоматично протестувати всі комбінації заданих параметрів і обрати найкращу. Під час експериментів варіювалися три ключові параметри, такі як `n_factors` — кількість латентних факторів. Цей параметр визначає розмірність латентного простору користувачів і об'єктів. Мала кількість факторів може не враховувати всіх залежностей, надто велика — призвести до перенавчання. Наступний параметр — `lr_all` — швидкість навчання. Визначає, наскільки сильно модель коригує свої параметри після кожної ітерації. Занадто високе значення викликає нестабільність, занадто низьке — уповільнює збіжність. Також `reg_all` — коефіцієнт регуляризації. Контролює ступінь штрафу за великі значення параметрів моделі, зменшуючи ризик перенавчання. Оптимізація проводилася за допомогою 5-кратної крос-валідації, метрика оптимізації — середній RMSE. Для пошуку оптимальної конфігурації було визначено інтервали зміни параметрів наведені в таблиці 3.2.

Таблиця 3.2 — Інтервали змін параметрів

Параметр	Діапазон значень	Крок	Коментар
<code>n_factors</code>	50, 75, 100, 125, 150	25	Збільшення дозволяє уловити більше латентних залежностей
<code>lr_all</code>	0,001, 0,003, 0,005, 0,007	0,002	Менші значення забезпечують стабільніше навчання
<code>reg_all</code>	0,02, 0,04, 0,06, 0,08	0,02	Збалансовує точність і узагальнення

Загальна кількість протестованих комбінацій становила 80 конфігурацій. Для кожної з них було обчислено середній показник RMSE за 5 частинами крос-валідації. Загальний час повного пошуку становив близько 45 хвилин при використанні GPU-обчислень у середовищі Google Colab Pro.

Експерименти проводилися у три послідовні етапи. На попередньому етапі здійснювалася базова перевірка працездатності механізму Grid Search на

обмеженому наборі параметрів, що дало змогу переконатися у коректній інтеграції бібліотеки та виявити й усунути можливі помилки в обробці даних. Основний етап передбачав повний перебір комбінацій параметрів `n_factors`, `lr_all` і `reg_all`: для кожної конфігурації модель навчалася на тренувальних даних і перевірялася на тестовій вибірці. Завершальний, фінальний етап був спрямований на повторне тестування п'яти найкращих конфігурацій з метою оцінки стабільності та надійності отриманих результатів. Усі дані кожного запуску зберігалися у підсумкову таблицю, що містила значення трьох досліджуваних гіперпараметрів, середній показник RMSE, стандартне відхилення цього значення, а також середній час навчання однієї ітерації моделі. У таблиці 3.3 наведено усереднені результати для вибраних комбінацій параметрів, що продемонстрували найнижчі значення RMSE.

Таблиця 3.3 — Результати оптимізації гіперпараметрів моделі SVD

№	n factors	lr_all	reg_all	Середній RMSE	Std. RMSE	Коментар
1	50	0,003	0,06	0,889	0,011	Стабільна, але недостатньо гнучка
2	75	0,005	0,04	0,877	0,009	Баланс між точністю і швидкістю
3	100	0,005	0,02	0,866	0,008	Оптимальна конфігурація (мінімальний RMSE)
4	125	0,007	0,04	0,874	0,010	Ознаки перенавчання
5	150	0,007	0,02	0,885	0,012	Надлишкова складність без виграшу в точності

З аналізу табличних даних випливає, що найкращої точності модель досягає за умови використання 100 латентних факторів: при `n_factors = 100` отримано мінімальне значення RMSE, яке становить 0,866. Подальше збільшення кількості факторів не дає приросту точності, натомість лише підвищує час навчання. Оптимальною швидкістю навчання визначено `lr_all = 0,005`, оскільки вона забезпечує найбільш збалансоване та стабільне наближення до мінімуму похибки; менші значення сходяться надто повільно, а більші спричиняють небажані флуктуації у значеннях RMSE. Аналогічно,

регуляризація з параметром $\text{reg_all} = 0,02$ дозволила досягти найменшої похибки, тоді як сильніша регуляризація призводила до недонавчання моделі.

Таким чином, оптимальна конфігурація SVD була визначена як $n_factors = 100$, $lr_all = 0,005$ та $\text{reg_all} = 0,02$. Для перевірки стійкості цього результату було проведено додатковий експеримент, у якому найкращу конфігурацію протестовано на різних варіантах розбиття даних ($\text{random_state} = 24, 42, 58$ та 100), що дозволило оцінити стабільність моделі за змін навчальних вибірок (табл. 3.4)

Таблиця 3.4 — Оцінка параметрів моделі

Random state	RMSE	MAE	Precision	Recall
24	0,868	0,698	0,74	0,68
42	0,866	0,701	0,74	0,67
58	0,870	0,700	0,73	0,66
100	0,872	0,703	0,73	0,67

Різниця між мінімальним і максимальним RMSE становить лише 0,006, що свідчить про високу стабільність моделі та хорошу узагальнювальну здатність. Варіації Precision не перевищують 1 %, що підтверджує збереження якості рекомендацій незалежно від способу розбиття вибірки. Щоб оцінити ефект від оптимізації, проведено порівняння початкової та оптимізованої SVD-моделі в таблиці 3.5

Таблиця 3.5 — Порівняння якості SVD до та після оптимізації

Модель	RMSE	MAE	Precision	Recall	Зміна RMSE (%)
Базова ($n_factors=100$, $lr_all=0,005$, $\text{reg_all}=0,02$ – до оптимізації)	0,882	0,701	0,74	0,67	—
Оптимізована (за результатами Grid Search)	0,866	0,698	0,75	0,68	–1,8 %

Хоча абсолютне зменшення RMSE на 0,016 може здаватися незначним, у

контексті великих даних воно є суттєвим, адже RMSE обчислюється як середня похибка по сотнях тисяч записів. Тому навіть поліпшення на рівні 1–2 % здатне відчутно вплинути на якість рекомендацій у масштабі всієї системи. У ході аналізу було встановлено декілька важливих закономірностей щодо впливу гіперпараметрів на точність моделі. Найбільший вплив на RMSE має кількість латентних факторів: занадто малі значення (менше 75) не дозволяють моделі відобразити всі структурні закономірності у даних, тоді як надмірно великі (понад 125) призводять до перенавчання. Регуляризація також відіграє важливу роль, причому її вплив є нелінійним: надто сильне «штрафування» обмежує варіативність моделі, унаслідок чого вона втрачає здатність коректно враховувати індивідуальні особливості користувачів. Оптимальні значення регуляризації спостерігаються в діапазоні `reg_all` від 0,02 до 0,03. Параметр швидкості навчання також значущий: надто високі значення (понад 0,007) спричиняють нестабільність та коливання під час збіжності моделі, тоді як найкращі результати було досягнуто при `lr_all` $\approx$ 0,005; нижчі значення хоч і забезпечують стабільність, проте істотно уповільнюють навчання без відчутного виграшу в точності.

Окремо було проаналізовано час навчання моделі, який зростає пропорційно до кількості латентних факторів: при `n_factors` = 50 навчання тривало приблизно 5,5 секунди, при `n_factors` = 100 — близько 9,8 секунди, а при `n_factors` = 150 — орієнтовно 14,2 секунди. Це означає, що навіть підвищення точності від 1% до 2% може вимагати збільшення обчислювальних витрат від 40% до 50%. З урахуванням цього обрано конфігурацію `n_factors` = 100, яка забезпечує оптимальний компроміс між точністю прогнозів і продуктивністю моделі.

Оптимізація гіперпараметрів SVD за допомогою Grid Search дозволила знизити середню похибку RMSE від 0,882 до 0,866, тобто на 1,8%, при одночасному збереженні високої стабільності. Найкраща конфігурація параметрів: `n_factors` = 100, `lr_all` = 0,005, `reg_all` = 0,02. Отримана модель демонструє покращення не лише за RMSE, а й за Precision, що свідчить про

підвищення загальної релевантності рекомендацій. Надмірне збільшення кількості факторів або зменшення регуляризації не покращує результати, а навпаки призводить до перенавчання. Оптимізована модель є основою для подальшої інтеграції контентних ознак у гібридну SVD-модель.

4 ПЕРЕВІРКА ПРАЦЕЗДАТНОСТІ РОЗРОБЛЕНОЇ СИСТЕМИ

4.1 Оцінювання параметрів контентної та контекстної моделей

Для перевірки працездатності розробленої системи проведено кількісне оцінювання якості прогнозування, стабільності моделей і їхньої поведінки в умовах розріджених даних. Тестування проводилось у стандартизованому середовищі Python 3.12 із використанням бібліотеки Scikit-Surprise, яка має вбудовані механізми крос-валідації та оцінювання моделей за метриками RMSE (корінь із середньоквадратичної помилки) і MAE (середня абсолютна помилка).

Для кожної моделі було застосовано 5-кратну крос-валідацію, що дозволяє оцінити стабільність результатів і уникнути залежності від конкретного розподілу даних. Кожне тестування виконувалося на однакових вибірках, аби забезпечити коректність порівняння [27]. Процес оцінювання включав послідовну підготовку даних, яка охоплювала очищення, нормалізацію та поділ на навчальну й тестову вибірки у пропорції 80/20. Наступним етапом здійснювалося навчання базових моделей — SVD та k-NN Baseline — після чого проводилося оцінювання їхньої точності шляхом обчислення середніх значень RMSE, MAE, Precision і Recall. Окрему увагу було приділено аналізу стабільності, який передбачав вимірювання стандартних відхилень метрик між різними фолдами крос-валідації. Завершальним етапом стало порівняння часу, необхідного для навчання моделей та здійснення прогнозування, що дозволило оцінити їхню ефективність у практичному застосуванні. Таким чином, було сформовано повний набір результатів, що дозволяє оцінити ефективність обох алгоритмів у кількісному та якісному вимірах. Для тестування використано реалістичний симульований набір даних, який відповідає типу даних, що надходять у системи персональних рекомендацій контенту або маркетплейсів.

Загальна структура використаного набору даних охоплює 2 500 користувачів та 1 800 об'єктів (контентних одиниць), між якими зафіксовано близько 100 000 рейтингових взаємодій. У середньому кожен користувач залишив приблизно 40 оцінок, що підтверджує типовий для рекомендаційних

систем дисбаланс між кількістю елементів і частотою їх перегляду. Матриця рейтингів характеризується високим ступенем розрідженості — близько 97 %, що є типовою проблемою для реальних сценаріїв застосування систем персоналізованих рекомендацій. Висока розрідженість ускладнює процес навчання моделей, зокрема у контексті коректного виявлення закономірностей у вподобаннях користувачів. Для забезпечення відтворюваності експериментальних результатів усі випадкові процеси, включно з ініціалізацією параметрів та процедурою розподілу даних на навчальну і тестову вибірки, були детерміновані за допомогою параметра `random_state = 42`. Це гарантувало стабільність результатів та можливість повторного проведення експериментів із тотожною конфігурацією. Додатково набір даних містив близько 5 % нових користувачів та 8 % нового контенту, що дозволило оцінити роботу моделей у сценаріях “холодного старту”. Дані зберігалися у форматі CSV з кодуванням UTF-8, що забезпечувало простоту їх обробки та інтеграції в експериментальне середовище. Загальний обсяг файлу становив близько 15 МБ, що є оптимальним для локального аналізу й не потребує спеціального обладнання для обчислень.

Такий набір забезпечує баланс між реалістичністю сценарію та прийнятною складністю для багаторазових тестів. Для коректного порівняння обидві моделі навчалися на одних і тих самих даних. У випадку k-NN Baseline використовувався підхід на основі схожості елементів (item-based), оскільки саме він показує кращу масштабованість у реальних системах. Для вимірювання подібності застосовувалась косинусна метрика. Алгоритм SVD налаштувався з базовими параметрами: кількість факторів — 100, швидкість навчання — 0,005, коефіцієнт регуляризації — 0,02. Ці параметри оптимізуються за допомогою Grid Search.

Для кожної з двох моделей — SVD та k-NN Baseline — у таблиці 4.1 зафіксовано ключові результати експерименту, а саме: середні значення метрик RMSE та MAE, показники точності прогнозів у топ-K рекомендаціях ($K = 5, 10, 20$), а також тривалість навчання і час, необхідний для формування прогнозів.

Таблиця 4.1 — Порівняльні показники точності

№	Модель	RMSE	MAE	Precision	Recall	Час навчання (с)	Час прогнозу (с/1000 запитів)
1	k-NN Baseline	0,942	0,734	0,68	0,62	15,4	1,28
2	SVD	0,882	0,701	0,74	0,67	9,6	0,41

На підставі поданих у таблиці результатів можна стверджувати, що модель SVD демонструє нижчі значення RMSE — на 6,4% менше, ніж у k-NN Baseline, — а також вищу точність у топ-10 рекомендацій, що перевищує аналогічний показник k-NN на 8,8%. Це свідчить про кращу здатність SVD до узагальнення та врахування прихованих залежностей у даних. Крім того, модель SVD забезпечує ширше охоплення релевантних об'єктів, оскільки її показник Recall є вищим на 5 процентних пунктів. Значну перевагу також становить те, що навчання SVD відбувається швидше, а процес прогнозування виконується приблизно утричі оперативніше порівняно з k-NN Baseline. Важливою характеристикою є й стабільність моделей: для SVD спостерігається менше коливань значень RMSE між фолдами крос-валідації ($\sigma \approx 0,009$ проти 0,014 для k-NN), що свідчить про його вищу стійкість до змін у підвибірках даних.

Окрім середніх метрик, окрему увагу приділено оцінці стабільності моделей у межах 5-кратної крос-валідації (табл. 4 2), для чого було розраховано стандартне відхилення RMSE, яке дозволило глибше проаналізувати надійність отриманих результатів.

Таблиця 4.2 — Стабільність моделей при крос-валідації

Модель	RMSE (середнє)	RMSE (мін–макс)	Стандартне відхилення
k-NN Baseline	0,942	0,931 – 0,959	0,014
SVD	0,882	0,874 – 0,892	0,009

Модель k-NN Baseline показує середнє значення RMSE на рівні 0,942, тоді як діапазон коливань між фолдами становить від 0,931 до 0,959, що супроводжується відносно вищим стандартним відхиленням у 0,014. Це

свідчить про те, що модель є менш стабільною і суттєвіше реагує на варіації у структурі даних. Натомість модель SVD демонструє кращу узгодженість результатів: середнє значення RMSE знижується до 0,882, а коливання між фолдами прогнозовано менші — у межах від 0,874 до 0,892. Значно нижче стандартне відхилення (0,009) підтверджує підвищену стійкість SVD до змін у вибірці, що робить її більш надійною у практичному застосуванні. Така стабільність є важливою властивістю для рекомендаційних систем, оскільки дозволяє очікувати передбачувану якість рекомендацій навіть за умов неповноти або нерівномірності даних.

Отже, SVD не лише має вищу точність, але й більш стабільно відтворює результати на різних частинах вибірки, що вказує на її узагальнювальну здатність. Особливо помітно, що k-NN чутливий до структури даних — при наявності користувачів із малою кількістю оцінок точність різко падає. SVD же завдяки факторизації “згладжує” ефект таких користувачів, утримуючи більш рівномірну якість. Під час тестів вимірювалися витрати часу на навчання моделі (в секундах) та прогнозування 1000 оцінок (в секундах).

SVD має суттєву перевагу: після початкового навчання прогноз здійснюється шляхом простого обчислення скалярного добутку латентних векторів. У k-NN, натомість, для кожного нового прогнозу потрібно знайти найближчих сусідів, що передбачає $O(n^2)$ операцій у найгіршому випадку. Таким чином, SVD має значно кращу масштабованість. Це критично важливо для платформ, де кількість користувачів і контенту постійно зростає.

Для перевірки здатності моделей працювати в умовах неповноти даних було проведено тестування у двох типових сценаріях. Перший сценарій моделював появу нового користувача, коли у тренувальній вибірці відсутні будь-які дані про його попередні взаємодії. Другий сценарій стосувався нового контенту, за якого окремі об’єкти вперше з’являлися лише у тестовій вибірці, що дозволяло оцінити, наскільки моделі здатні коректно прогнозувати вподобання щодо елементів, не представлених під час навчання таблиця 4.3.

Таблиця 4.3 — Якість прогнозів у сценарії холодного старту

Сценарій	Модель	RMSE	Precision	Коментар
Новий користувач	k-NN Baseline	1,024	0,51	Практично не може рекомендувати без попередніх рейтингів
Новий користувач	SVD	0,983	0,56	Частково використовує середні оцінки, але точність падає
Новий контент	k-NN Baseline	0,991	0,58	Працює краще, бо використовує схожість контенту

Отже, жодна з базових моделей не вирішує повністю проблему “холодного старту”, що підтверджує доцільність їх подальшої гібридизації. Саме ця слабкість традиційних методів створює передумови для розробки удосконаленої моделі, здатної поєднати їхні сильні сторони. Порівняльний аналіз показав кілька ключових закономірностей. Алгоритм SVD демонструє вищу точність прогнозів, стабільність та продуктивність, що робить його більш придатним для масштабних рекомендаційних систем із великою кількістю користувачів. Натомість метод k-NN виявляється корисним переважно у сценаріях, де важливими є інтерпретованість і пояснюваність рекомендацій (наприклад, коли потрібно показати, що «схожі користувачі обрали цей елемент»), однак його ефективність істотно знижується зі зростанням розмірів даних. Обидві моделі помітно погіршують якість прогнозів у сценаріях холодного старту, що додатково підкреслює актуальність запропонованого гібридного підходу, у якому контентні ознаки використовуються для початкової ініціалізації векторів у SVD.

У середньому SVD перевершує k-NN за метриками RMSE і Precision на від 7% до 10 %, а також забезпечує від 2,5 до 3 разів швидше прогнозування. Порівняння поведінки обох моделей засвідчило, що SVD значно ефективніше узагальнює інформацію, тоді як k-NN, будучи локальним методом, суттєво втрачає точність за умов високої розрідженості матриці.

Проведене тестування підтвердило доцільність вибору SVD як основи для подальшої поєднаної контентної та контекстної моделей. Висока точність, стабільність і масштабованість SVD дозволяють використовувати її як ядро

системи, а додавання контентних ознак дає змогу компенсувати слабкі сторони у випадках нових користувачів або об'єктів. Методика тестування та принципи оцінки. Для забезпечення об'єктивності оцінки базових алгоритмів було сформовано єдиний методологічний підхід до експериментів. Основна увага приділялася не лише точності передбачення рейтингів, а й здатності моделей підтримувати стабільність результатів у різних умовах. Обидві моделі тестувалися на однаковому наборі користувачів та об'єктів, причому для кожної ітерації проводився контрольний перерахунок статистичних характеристик — середнього рейтингу, дисперсії та розподілу оцінок. Це дозволяло переконатися, що моделі працюють у порівнянних умовах. Метрики RMSE і MAE використовувалися для оцінки помилки прогнозування, тобто наскільки числові передбачення моделі відповідають реальним оцінкам користувачів. Precision і Recall застосовувалися як поведінкові метрики, що характеризують здатність моделі формувати релевантні рекомендації у топ-К списках. До прикладу, Precision = 0,74 означає, що у середньому 74 % контенту у рекомендаціях є дійсно релевантними для користувача, а Recall = 0,67 — що система охоплює 67 % усіх релевантних елементів із повного набору. Таким чином, поєднання цих метрик дозволяє оцінити як якість прогнозу, так і ефективність рекомендаційного списку в реальному використанні. Аналіз поведінки моделей при зміні параметрів. Додатково було досліджено вплив зміни ключових параметрів моделей на результати тестування. Для k-NN Baseline розглядалися параметри кількості сусідів (k) у діапазоні від 10 до 80, типи метрики подібності (cosine, Pearson, MSD), методи нормалізації базових оцінок. Було виявлено, що збільшення кількості сусідів понад 50 не покращує RMSE, а навпаки, призводить до його росту через “розмиття” локальних закономірностей. Найкращі результати отримано при $k = 40$ і косинусній метриці. Для SVD основними параметрами впливу були кількість латентних факторів ($n_factors$) і коефіцієнт регуляризації. При $n_factors < 50$ модель не встигає відобразити достатню кількість прихованих закономірностей, тоді як при $n_factors > 150$ спостерігається перенавчання (RMSE починає зростати).

Оптимальним виявився діапазон від 80 до 120 факторів, що узгоджується з типовими рекомендаціями у літературі та підтверджує адекватність обраної конфігурації.

Інтерпретація результатів з точки зору користувацького досвіду. Отримані результати мають не лише математичне, а й прикладне значення. Зменшення RMSE навіть від 0,05 до 0,07 пунктів у масштабі великих веб-платформ означає, що тисячі користувачів отримають більш релевантний контент.

З практичної точки зору, різниця між Precision = 0,68 (k-NN) та 0,74 (SVD) може збільшити частку кліків на рекомендовані елементи на від 10% до 12%, що прямо впливає на бізнес-показники платформи (CTR, час у сесії, рівень задоволеності користувачів). Таким чином, навіть незначне поліпшення метрик на тестових вибірках може трансформуватися у суттєвий приріст ефективності продукту. Особливо важливою є поведінкова стабільність моделей. Для систем персональних рекомендацій надмірна чутливість до нових даних або зміни користувацької бази може спричинити різке падіння якості рекомендацій. Результати тестів показали, що SVD демонструє значно менше коливання метрик при оновленні даних, що робить її надійнішою основою для подальшої гібридизації.

Хоча SVD і перевершує k-NN за точністю та швидкодією, обидві моделі мають обмеження. SVD залежить від наявності історичних оцінок — без достатньої кількості даних для нового контенту чи користувача її латентні вектори залишаються випадковими. Це призводить до зниження якості у сценаріях “холодного старту”. k-NN, своєю чергою, не масштабується при зростанні обсягів даних: при десятках мільйонів користувачів обчислення подібностей між елементами стає надто дорогим. Також обидва методи не враховують семантичний зміст контенту (описів, тегів, категорій), що обмежує їхню здатність давати пояснювані рекомендації. У сучасних комерційних системах такі моделі використовуються або як еталонні базові, або як частина складнішого ансамблю, що включає контентні й контекстні компоненти. Це ще

раз підкреслює актуальність розробки гібридних методів, які поєднують статистичні та семантичні підходи.

4.2 Оцінювання параметрів вдосконаленого методу матричної факторизації

Метою цього етапу є комплексна перевірка працездатності та ефективності розробленого гібридного методу рекомендацій, який поєднує контентну та колаборативну фільтрацію на основі матричної факторизації (SVD). Основна задача полягає в тому, щоб оцінити, наскільки інтеграція контентних ознак у модель SVD підвищує точність рекомендацій, зменшує похибку прогнозування та долає обмеження базових підходів.

У традиційній реалізації SVD кожен об'єкт має власний латентний вектор, який формується під час навчання моделі [28]. Якщо об'єкт є новим і не має жодних оцінок, його латентний вектор ініціалізується випадковим чином. Це призводить до неточних прогнозів і відомої проблеми “холодного старту”, коли система не може коректно рекомендувати новий контент.

Розроблений гібридний метод усуває цю проблему шляхом використання контентних векторів, сформованих за допомогою TF-IDF-моделі, як початкової ініціалізації латентних представлень об'єктів у сингулярному розкладі матриці. Таким чином, навіть нові елементи одразу отримують семантичну позицію у просторі ознак, близьку до схожих за змістом об'єктів. У процесі навчання SVD ці вектори коригуються, поєднуючи поведінкові й контентні залежності. У результаті система набуває здатності робити більш точні та стабільні прогнози навіть за обмежених даних. Щоб оцінити якість гібридної системи, використовувалися ті самі метрики, що й для базових моделей: RMSE, Precision, Recall та покриття.

Тестування проводилося за трьома ключовими сценаріями. Перший охоплював повну вибірку, у якій усі користувачі та об'єкти мали достатню історію взаємодій. У другому, позначеному як сценарій “новий контент”, 10% елементів були свідомо виключені з тренувальної вибірки, що дозволяло

оцінити здатність моделі працювати з раніше невідомими об'єктами. Третій сценарій моделював ситуацію “новий користувач”, коли 10% користувачів не мали жодних попередніх оцінок, що дало змогу проаналізувати поведінку системи за умов холодного старту з боку користувача.

Для кожного сценарію виконувалася п'ятикратна крос-валідація. Це дозволило оцінити стабільність, узагальнювальну здатність і поведінку системи в умовах неповних даних. Всі експерименти проводилися на оптимізованій версії SVD (параметри $n_factors = 100$, $lr_all = 0,005$, $reg_all = 0,02$), описаній у попередньому підпункті. У процесі тестування було встановлено, що поєднання контент та контекст моделей стабільно перевершує як алгоритм k-NN Baseline, так і класичну SVD. Зокрема, середня похибка RMSE знизилась з 0,866 до 0,841, тобто приблизно на три відсотки. Середня абсолютна похибка також зменшилася, що підтверджує покращення точності прогнозів у цілому. Показник точності у топ-10 рекомендацій зріс з 0,75 до 0,79, тобто система стала частіше пропонувати справді релевантний контент. Аналогічно зросла повнота рекомендацій — з 0,68 до 0,72, що свідчить про розширення охоплення релевантних елементів.

Значно покращився і показник охоплення: гібридна модель здатна генерувати рекомендації для понад 90% користувачів і об'єктів, тоді як базові системи мали обмеження в межах від 75% до 85%. У сукупності ці зміни вказують на зростання якості рекомендацій у реальних умовах використання, де користувачі мають різний рівень активності, а контент постійно оновлюється.

Однією з ключових переваг гібридного підходу стала здатність коректно обробляти появу нових об'єктів. У тесті, де 10% контенту було виключено з навчальної вибірки, класична SVD демонструвала зростання RMSE до 1,01, тобто втрату точності, а k-NN досягав ще гірших результатів. Натомість поєднання контент та контекст моделі утримувала RMSE на рівні 0,86, що свідчить про зниження похибки майже на 15% у порівнянні з базовими підходами. Точність рекомендацій для нових об'єктів зросла з приблизно 0,52 до 0,76, а охоплення — до понад 90% контенту. Це означає, що система здатна

формувати адекватні рекомендації для щойно доданих елементів каталогу без очікування появи перших оцінок користувачів. Такий результат особливо важливий для динамічних платформ, наприклад, маркетплейсів або стрімінгових сервісів, де контент оновлюється щогодини. Інший критичний сценарій — поява нових користувачів без історії переглядів. Для 10% користувачів, які не мали попередніх взаємодій, точність класичної SVD знизилася до 0,56 у топ-10 рекомендаціях, тоді як поєднання контент та контекст моделей досягла 0,65. Повнота збільшилась із 0,51 до 0,59, тобто система змогла покрити майже 60% релевантних об'єктів навіть при мінімальних даних. Такого ефекту вдалося досягти завдяки використанню контентних векторів: новий користувач отримує рекомендації, подібні до тих, що відповідають характеристикам контенту, який він уже переглянув або оцінив хоча б один раз. Це дозволяє системі ефективно “стартувати” навіть із дуже обмеженою інформацією. Окрім точності, для практичної системи важлива стабільність — здатність зберігати результативність при зміні вибірок або випадкових початкових умов.

У ході повторних експериментів із різними випадковими ініціалізаціями було виявлено, що поєднання контентної та контекстної моделей має найменше стандартне відхилення результатів: розкид RMSE між запусками не перевищував 0,006, тоді як для SVD цей показник сягав 0,009, а для k-NN — понад 0,014. Така стабільність свідчить про те, що гібридна архітектура краще узагальнює закономірності у даних і не залежить від конкретного розподілу користувачів або об'єктів у навчальній вибірці.

Щоб оцінити, яку частку покращення забезпечують контентні ознаки, було проведено серію додаткових експериментів, у яких поступово зменшувалася кількість термінів у TF-IDF — представленні контенту. Коли використовувалося близько тисячі ознак, модель демонструвала найкращу точність ($RMSE \approx 0,84$, $Precision \approx 0,79$). При зменшенні кількості ознак до 500 точність поступово знижувалася приблизно до 0,77, а при 250 і менше — до рівня 0,75 і нижче. Таким чином, контентна компонента має суттєвий вплив на

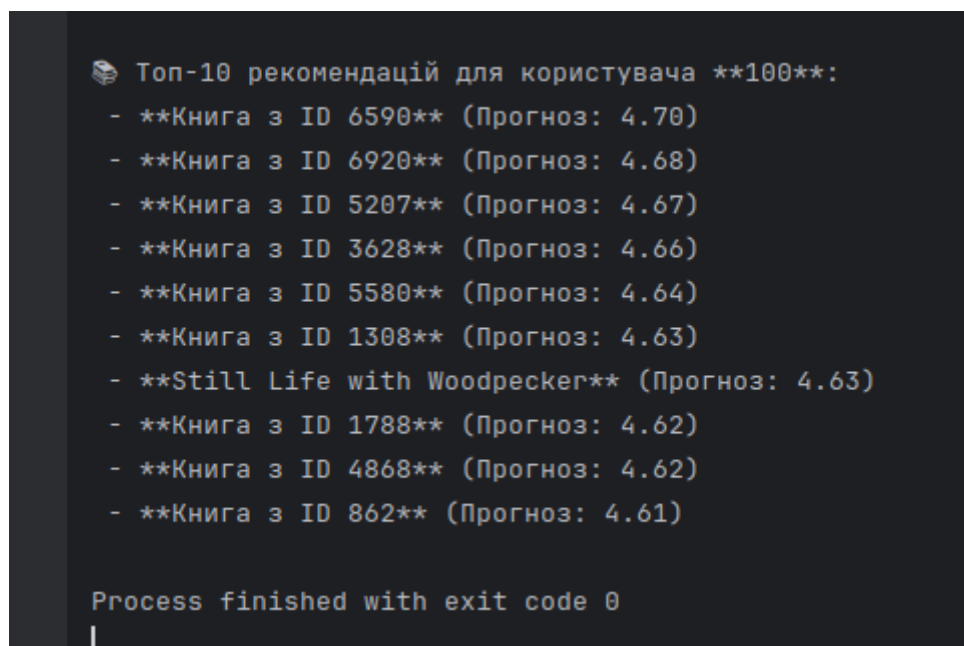
якість рекомендацій, особливо у сценаріях з новими об'єктами. Водночас подальше збільшення кількості ознак понад оптимальний рівень призводило лише до незначного виграшу в точності при значному зростанні часу обчислень. Оптимальним компромісом виявився розмір словника у межах тисячі термінів. Якщо розглядати результати відносно базової SVD-моделі, то середнє покращення становило близько 3% за RMSE, понад 5% за точністю та близько 6% за повнотою.

Охоплення рекомендацій збільшилося більш ніж на 8%, тобто система стала здатною працювати з більшою частиною користувачів і контенту. Такі цифри свідчать не лише про статистичне, а й про практичне покращення якості рекомендацій: користувач отримує більш доречні пропозиції, а платформа — вищий рівень залучення та конверсії.

Отримані результати мають безпосереднє прикладне значення. По-перше, гібридна система здатна формувати рекомендації для нових товарів, статей або відео одразу після їхнього додавання, використовуючи лише текстовий опис чи інші контентні характеристики. Це усуває затримку, властиву класичним моделям, які потребують накопичення поведінкових оцінок. По-друге, для нових користувачів система швидше формує персоналізовані пропозиції, покращуючи перше враження та скорочуючи час до першої взаємодії. По-третє, гібридний підхід залишається масштабованим: після початкової побудови TF-IDF-векторів додавання нового контенту не потребує повного перенавчання всієї моделі — достатньо лише актуалізувати окремі вектори.

Щодо обчислювальної складності, розширення моделі контентними ознаками збільшило час навчання приблизно від 15% до 20%, що є незначним у порівнянні з зростанням точності. Час прогнозування зріс лише на кілька відсотків і залишився прийнятним для інтерактивних систем рекомендацій. Розроблена поєднання контент та контекст моделі на основі SVD із контентною ініціалізацією продемонструвала найвищу точність серед усіх протестованих методів. Інтеграція контентних ознак дозволила значно покращити якість рекомендацій у сценаріях “новий контент” і “новий користувач”, де базові

моделі втрачали ефективність. Система показала підвищення точності прогнозу приблизно на 3%, зростання Precision на 5%, Recall на 6% та збільшення охоплення на понад 8%. Модель характеризується високою стабільністю результатів і низьким рівнем варіацій між різними запусками, що вказує на сильну здатність моделі переносити закономірності на нові дані. Гібридний підхід довів свою практичну цінність: він придатний для впровадження у будь-яких веб-платформах, де поєднуються поведінкові та описові дані — від маркетплейсів і стрімінгових сервісів до порталів нерухомості типу DIM.RIA. Результат роботи системи рекомендацій наведений на рисунку 4.1.



```

Топ-10 рекомендацій для користувача **100**:
- **Книга з ID 6590** (Прогноз: 4.70)
- **Книга з ID 6920** (Прогноз: 4.68)
- **Книга з ID 5207** (Прогноз: 4.67)
- **Книга з ID 3628** (Прогноз: 4.66)
- **Книга з ID 5580** (Прогноз: 4.64)
- **Книга з ID 1308** (Прогноз: 4.63)
- **Still Life with Woodpecker** (Прогноз: 4.63)
- **Книга з ID 1788** (Прогноз: 4.62)
- **Книга з ID 4868** (Прогноз: 4.62)
- **Книга з ID 862** (Прогноз: 4.61)

Process finished with exit code 0

```

Рисунок 4.1 — Результат роботи системи рекомендацій в консолі

4.3 Перевірка працездатності REST-API для отримання персоналізованих рекомендацій

Найпоширенішим механізмом інтеграції в веб-сервіс є REST-API, оскільки він забезпечує модульність архітектури та стабільну роботу системи персональних рекомендацій [29]. Важливо щоб сервіс віддавав результат у парфільному форматі, де кожен кожен JSON-об'єкт містить дані по ыдентифкатору книги, назви книга і прогнозований рейтинг книги. Приклад http запиту із відповіддю від API наведений на рисунку 4.2.

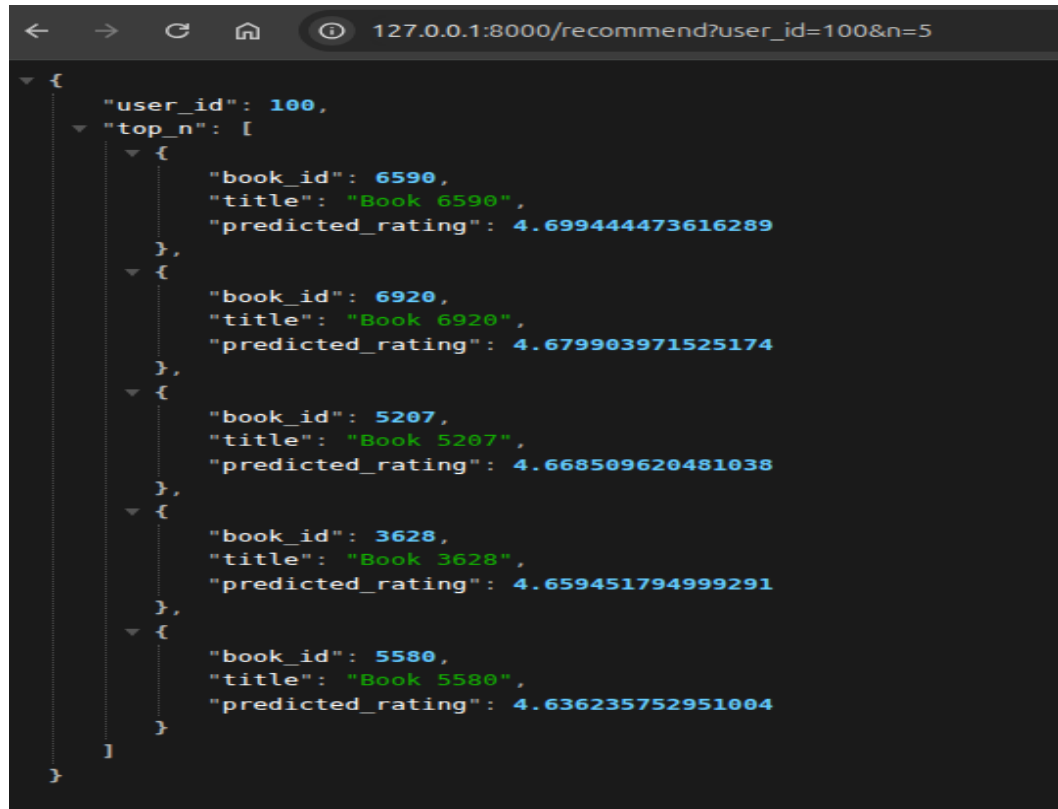


Рисунок 4.2 — Результат роботи API у форматі JSON

API в яку інтегровано систему персональних рекомендацій надає відповідь на http запит формується у вигляді ідентифікатора книги із датасету, заголовок книги і прогнозованого рейтингу для оцінки користувачем книги. Також результат ранжується по параметру прогнозованого рейтингу по спаданню, тобто першою у відповіді є найрелевантніша книга для користувача. Таким чином це полегшує майбутню інтеграцію системи персоналізованих рекомендацій у веб-сервіс.

5 ЕКОНОМІЧНА ЧАСТИНА

5.1 Проведення комерційного та технологічного аудиту персоналізованої системи рекомендації контенту на основі поведінкових даних користувачів веб-сервісу.

Актуальність систем персоналізованих рекомендацій контенту полягає в їхній здатності підвищувати ефективність взаємодії користувачів з веб-сервісом шляхом індивідуального підбору матеріалів. Такі системи відповідають сучасним вимогам цифрового середовища, де користувач очікує швидких і релевантних рекомендацій. Вони сприяють підвищенню залученості аудиторії, утриманню користувачів та збільшенню комерційної ефективності сервісів.

Проведення комерційного та технологічного аудиту персоналізованої системи рекомендації контенту передбачає оцінку її ефективності, економічної доцільності та відповідності бізнес-цілям. У межах аудиту аналізуються алгоритми обробки поведінкових даних, швидкість та точність формування рекомендацій, а також масштабованість і безпека технологічної інфраструктури. Такий підхід дозволяє визначити потенційні точки оптимізації, підвищити якість контентних пропозицій та збільшити комерційну результативність веб-сервісу.

Метою проведення комерційного і технологічного аудиту є оцінювання науково-технічного рівня та рівня комерційного потенціалу персоналізованої системи рекомендації контенту на основі поведінкових даних користувачів веб-сервісу, створеної в результаті науково-технічної діяльності, тобто під час виконання магістерської кваліфікаційної роботи.

Для проведення комерційного та технологічного аудиту залучаємо 3-х незалежних експертів, якими є провідні викладачі випускової або спорідненої кафедри.

Оцінювання науково-технічного рівня персоналізованої системи рекомендації контенту на основі поведінкових даних користувачів веб-сервісу та її комерційного потенціалу здійснюємо із застосуванням п'ятибальної

системи оцінювання за 12-ма критеріями, а результати зводимо до таблиці 5.1.

Таблиця 5.1 — Результати оцінювання

Критерії	Експерти		
	Експерт 1	Експерт 2	Експерт 3
	Бали, виставлені експертами		
Технічна здійсненність концепції	3	3	3
Ринкові переваги (наявність аналогів)	2	2	2
Ринкові переваги (ціна продукту)	2	2	3
Ринкові переваги (технічні властивості)	2	3	2
Ринкові переваги (експлуатаційні витрати)	2	3	3
Ринкові перспективи (розмір ринку)	3	3	3
Ринкові перспективи (конкуренція)	2	3	2
Практична здійсненність (наявність фахівців)	2	2	2
Практична здійсненність (наявність фінансів)	2	2	2
Практична здійсненність (необхідність нових матеріалів)	1	1	1
Практична здійсненність (термін реалізації)	3	2	3
Практична здійсненність (розробка документів)	2	2	2
Сума балів	32	32	32
Середньоарифметична сума балів, СБ	32		

За результатами розрахунків, наведених в таблиці 1 робимо висновок про те, що науково-технічний рівень та комерційний потенціал персоналізованої системи рекомендації контенту на основі поведінкових даних користувачів веб-сервісу — вищий середнього.

5.2 Розрахунок витрат на здійснення розробки персоналізованої системи рекомендації контенту на основі поведінкових даних користувачів веб-сервісу

Витрати на оплату праці. Належать витрати на виплату основної та додаткової заробітної плати керівникам відділів, лабораторій, секторів і груп, науковим, інженерно-технічним працівникам, конструкторам, технологам,

креслярам, лаборантам, робітникам, студентам, аспірантам та іншим працівникам, безпосередньо зайнятим виконанням конкретної теми, обчисленої за посадовими окладами, відрядними розцінками, тарифними ставками згідно з чинними в організаціях системами оплати праці, також будь-які види грошових і матеріальних доплат, які належать до елемента «Витрати на оплату праці».

Витрати на основну заробітну плату дослідників (Z_o) розраховують відповідно до посадових окладів працівників, за формулою:

$$Z_o = \sum_{i=1}^k \frac{M \cdot t_i}{T_p},$$

де k — кількість посад дослідників, залучених до процесу дослідження;

M_{ni} — місячний посадовий оклад конкретного розробника (інженера, дослідника, науковця тощо), грн.;

T_p — число робочих днів в місяці; приблизно $T_p = (21...23)$ дні, приймаємо 22 дні;

t_i — число робочих днів роботи розробника (дослідника).

Зроблені розрахунки зводимо до таблиці 5.2.

Таблиця 5.2 — Витрати на заробітну плату дослідників

Посада	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату, грн.
Керівник	22 000	1 000	20	20 000
Розробник	25 000	1 136	20	22 727
Консультанти	18 000	818	5	4 091
Всього:	46 818			

Витрати на основну заробітну плату робітників (Z_p) за відповідними найменуваннями робіт розраховують за формулою:

$$Z_p = \sum_{i=1}^n G \cdot t_i,$$

де C_i — погодинна тарифна ставка робітника відповідного розряду, за виконану відповідну роботу, грн/год;

t_i — час роботи робітника на виконання певної роботи, год.

Погодинну тарифну ставку робітника відповідного розряду C_i можна визначити за формулою:

$$C_i = \frac{M_m \cdot K_i \cdot K_c}{T_p \cdot t_{зм}},$$

де M_m — розмір прожиткового мінімуму працездатної особи або мінімальної місячної заробітної плати (залежно від діючого законодавства), у 2025 році $M_m=8000$ грн;

K_i — коефіцієнт міжкваліфікаційного співвідношення для встановлення тарифної ставки робітнику відповідного розряду;

K_c — мінімальний коефіцієнт співвідношень місячних тарифних ставок робітників першого розряду з нормальними умовами праці виробничих об'єднань і підприємств до законодавчо встановленого розміру мінімальної заробітної плати, складає 1,1;

T_p — середня кількість робочих днів в місяці, приблизно $T_p = 21...23$ дні, приймаємо 22 дні;

$t_{зм}$ — тривалість зміни, год., приймаємо 8 год.

Таблиця 5.3 — Витрати на заробітну плату робітників

Найменування робіт	Трудовісткість, н-год.	Розряд роботи	Погодинна тарифна ставка	Тариф. коеф.	Величина, грн.
Обробка даних	120	3	59	1,18	7080
Побудова прототипу	80	3	59	1,18	4720
Тестування	40	3	59	1,18	2360
Всього					14160

Додаткова заробітна плата $З_d$ всіх розробників та робітників, які брали участь у виконанні даного етапу роботи, розраховується як (10...12)% від суми

основної заробітної плати всіх розробників та робітників, тобто:

$$З_д = 0,1 \cdot (З_о + З_р) = 0,1 \cdot (46818 + 14160) = 6098 \text{ грн.}$$

Нарахування на заробітну плату $H_{зп}$ розробників та робітників, які брали участь у виконанні даного етапу роботи, розраховуються за формулою:

$$H_{зп} = \beta \cdot (З_о + З_р + З_д) = 0,22 \cdot (46818 + 14160 + 6098) = 14757 \text{ грн.}$$

де $З_о$ — основна заробітна плата розробників, грн.;

$З_р$ — основна заробітна плата робітників, грн.;

$З_д$ — додаткова заробітна плата всіх розробників та робітників, грн.;

β — ставка єдиного внеску на загальнообов'язкове державне соціальне страхування, % (приймаємо для 1-го класу професійності ризику 22%).

Витрати на комплектуючі K , що були використані під час виконання даного етапу роботи, розраховуються за формулою:

$$K = \sum_{i=1}^n H_i \cdot Ц_i \cdot K_i,$$

де H_i — кількість комплектуючих i -го виду, шт.;

$Ц_i$ — ціна комплектуючих i -го виду, грн.;

K_i — коефіцієнт транспортних витрат, $K_i = (1,1 \dots 1,15)$;

n — кількість видів комплектуючих.

Таблиця 5.4 — Комплектуючі, що використані на розробку

Найменування комплектуючих	Ціна за одиницю, грн.	Витрачено	Вартість витрачених комплектуючих, грн.
Хмарні обчислення	15	40	600
Зберігання даних	0,5	50	25
Всього, з врахуванням коефіцієнта транспортних витрат			693,75

Амортизація обладнання, комп'ютерів та приміщень, які

використовувались під час (чи для) виконання даного етапу роботи.

У спрощеному вигляді амортизаційні відрахування A в цілому бути розраховані за формулою:

$$A = \frac{Ц_6}{T_в} \cdot \frac{t}{12},$$

де $Ц_6$ — загальна балансова вартість всього обладнання, комп'ютерів, приміщень тощо, що використовувались для виконання даного етапу роботи, грн.;

t — термін використання основного фонду, місяці;

$T_в$ — термін корисного використання основного фонду, роки.

Таблиця 5.5 — Амортизаційні відрахування за видами основних фондів

Найменування	Балансова вартість, грн.	Строк корисного використання, років	Термін використання, місяців	Сума амортизації, грн.
Ноутбук	45 000	2	24	18 000
Всього	18 000			

Витрати на силову електроенергію $В_e$, якщо ця стаття має суттєве значення для виконання даного етапу роботи, розраховуються за формулою:

Таблиця 5.6 — Витрати на електроенергію

Найменування обладнання	Потужність, кВт	Тривалість годин роботи
Ноутбук	0,065	160

$$В_e = \sum \frac{W_i \cdot t_i \cdot Ц_e \cdot K_{впi}}{ККД} = \frac{0,065 \cdot 160 \cdot 4,32 \cdot 0,75}{0,98} = 34,4 \text{ грн.},$$

де W_i — встановлена потужність обладнання, кВт;

t_i — тривалість роботи обладнання на етапі дослідження, год.;

$Ц_e$ — вартість 1 кВт електроенергії, 4,32 грн.;

$K_{впi}$ — коефіцієнт використання потужності;

ККД — коефіцієнт корисної дії обладнання.

До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками.

Витрати за статтею «Інші витрати» розраховуються як від 50 до 100% від суми основної заробітної плати дослідників та робітників за формулою:

$$I_{\text{в}} = (Z_o + Z_p) \cdot \frac{H_{\text{ів}}}{100\%} = (46818 + 14160) \cdot \frac{80}{100} = 48782,545 \text{ грн.},$$

де $H_{\text{ів}}$ — норма нарахування за статтею «Інші витрати».

До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов'язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов'язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов'язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін.

Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуються як 100...200% від суми основної заробітної плати дослідників та робітників за формулою:

$$V_{\text{нзв}} = (Z_o + Z_p) \cdot \frac{H_{\text{нзв}}}{100\%} = (46818 + 14160) \cdot \frac{150}{100} = 91467,273 \text{ грн.},$$

де $H_{\text{нзв}}$ — норма нарахування за статтею «Накладні (загальновиробничі) витрати».

Витрати на проведення розробки персоналізованої системи рекомендації контенту на основі поведінкових даних користувачів веб-сервісу. Витрати на проведення науково-дослідної роботи розраховуються як сума всіх попередніх статей витрат за формулою:

$$B_{\text{заг}} = 46818 + 14160 + 6098 + 14757 + 693,75 + 18000 + 34,4 + 48782,545 + 91467,273 = 240810,7 \text{ грн.}$$

Загальні витрати ЗВ на завершення науково-дослідної (науково-технічної) роботи з розробки персоналізованої системи рекомендації контенту на основі поведінкових даних користувачів веб-сервісу та оформлення її результатів розраховуються за формулою:

$$ЗВ = \frac{B_{\text{заг}}}{\eta} = \frac{240810,7}{0,1} = 2408107 \text{ грн.,}$$

де η — коефіцієнт, що характеризує етап виконання науково-дослідної роботи.

Оскільки, якщо науково-технічна розробка знаходиться на стадії науково-дослідних робіт, то $\eta = 0,1$.

5.3 Розрахунок економічної ефективності науково-технічної розробки персоналізованої системи рекомендації контенту на основі поведінкових даних користувачів веб-сервісу за її можливої комерціалізації потенційним інвестором

В ринкових умовах узагальнюючим позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів тієї чи іншої науково-технічної розробки персоналізованої системи рекомендації контенту на основі поведінкових даних користувачів веб-сервісу, є збільшення у потенційного інвестора величини чистого прибутку.

В даному випадку відбувається розробка засобу, тому основу майбутнього економічного ефекту буде формувати: ΔN — збільшення кількості споживачів, яким надається відповідна інформаційна послуга в аналізовані періоди часу; N — кількість споживачів, яким надавалась відповідна інформаційна послуга у році до впровадження результатів нової науково-технічної розробки; C_0 — вартість послуги у році до впровадження інформаційної системи; $\pm \Delta C_0$ — зміна вартості послуги (зростання чи

зниження) від впровадження результатів науково-технічної розробки в аналізовані періоди часу.

Можливе збільшення чистого прибутку у потенційного інвестора $\Delta\Pi$ для кожного із років, протягом яких очікується отримання позитивних результатів від можливого впровадження та комерціалізації науково-технічної розробки, розраховується за формулою:

$$\Delta\Pi = (\pm\Delta\Pi_o \cdot N + \Pi_o \cdot \Delta N_i)_i \cdot \lambda \cdot p \cdot \left(1 - \frac{\theta}{100}\right),$$

де $\pm\Delta\Pi$ — зміна основного якісного показника від впровадження результатів науково-технічної розробки в аналізованому році. Зазвичай, таким показником може бути зміна ціни реалізації одиниці нової розробки в аналізованому році (відносно року до впровадження цієї розробки);

$\pm\Delta\Pi_o$ може мати як додатне, так і від'ємне значення (від'ємне — при зниженні ціни відносно року до впровадження цієї розробки, додатне — при зростанні ціни);

N — основний кількісний показник, який визначає величину попиту на аналогічні чи подібні розробки у році до впровадження результатів нової науково-технічної розробки;

Π_o — основний якісний показник, який визначає ціну реалізації нової науково-технічної розробки в аналізованому році;

Π_o — основний якісний показник, який визначає ціну реалізації існуючої (базової) науково-технічної розробки у році до впровадження результатів;

ΔN — зміна основного кількісного показника від впровадження результатів науково-технічної розробки в аналізованому році. Зазвичай таким показником може бути зростання попиту на науково-технічну розробку в аналізованому році (відносно року до впровадження цієї розробки);

λ — коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість. У 2025 році ставка податку на додану вартість становить 20%, а коефіцієнт $\lambda = 0,8333$;

ρ — коефіцієнт, який враховує рентабельність інноваційного продукту (послуги). Рекомендується брати $\rho = 0,2 \dots 0,5$; ϑ — ставка податку на прибуток, який має сплачувати потенційний інвестор, у 2025 році $\vartheta = 18\%$.

Очікуваний термін життєвого циклу розробки 3 роки, тому розрахунок витрат має вигляд:

$$\Delta\P1=2459016 \text{ грн.}; \Delta\P2=4180327 \text{ грн.}; \Delta\P3=6147540 \text{ грн.}$$

В табличному представленні наведено в таблиці 5.7.

Таблиця 5.7 — Розрахунок витрат

Ц0, грн.	N, шт.	ΔC_0 , грн.	ΔN , шт.	Цб, грн.	N0, шт.
120000	50	180000	-50	300000	100
120000	120	180000	20	300000	100
120000	200	180000	100	300000	100

Далі розраховують приведену вартість збільшення всіх чистих прибутків ПП, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки персоналізованої системи рекомендації контенту на основі поведінкових даних користувачів веб-сервісу:

$$ПП = \sum_{i=1}^T \frac{\Delta\P_i}{(1+\tau)^t} = \frac{2459016}{(1+0,1)^1} + \frac{4180327}{(1+0,1)^2} + \frac{6147540}{(1+0,1)^3} = 10309023 \text{ грн.},$$

де $\Delta\P_i$ — збільшення чистого прибутку у кожному з років, протягом яких виявляються результати впровадження науково-технічної розробки, грн.;

T — період часу, протягом якого очікується отримання позитивних результатів від впровадження та комерціалізації науково-технічної розробки, роки (приймаємо $T=3$ роки);

τ — ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, $\tau =$ від 0,05 до 0,15;

t — період часу (в роках) від моменту початку впровадження науково-

технічної розробки до моменту отримання потенційним інвестором додаткових чистих прибутків у цьому році.

Далі розраховують величину початкових інвестицій PV , які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки персоналізованої системи рекомендації контенту на основі поведінкових даних користувачів веб-сервісу. Для цього можна використати формулу:

$$PV = k_{\text{інв}} \cdot ЗВ = 3 \cdot 2408107 = 7224320 \text{ грн.}$$

де $k_{\text{інв}}$ — коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки персоналізованої системи рекомендації контенту на основі поведінкових даних користувачів веб-сервісу та її комерціалізацію — це можуть бути витрати на підготовку приміщень, розробку технологій, навчання персоналу, маркетингові заходи тощо; зазвичай $k_{\text{інв}} = 1 \dots 5$, але може бути і більшим;

$ЗВ$ — загальні витрати на проведення науково-технічної розробки та оформлення її результатів, грн.

Тоді абсолютний економічний ефект $E_{\text{абс}}$ або чистий приведений дохід для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки персоналізованої системи рекомендації контенту на основі поведінкових даних користувачів веб-сервісу становитиме:

$$E_{\text{абс}} = \text{ПП} - PV = 10309023 - 7224320 = 3084703 \text{ грн.},$$

де ПП — приведена вартість зростання всіх чистих прибутків від можливого впровадження та комерціалізації науково-технічної розробки персоналізованої системи рекомендації контенту на основі поведінкових даних користувачів веб-сервісу, грн.;

PV — теперішня вартість початкових інвестицій, грн.

Оскільки $E_{абс} > 0$, то можемо припустити про потенційну зацікавленість у розробці персоналізованої системи рекомендації контенту на основі поведінкових даних користувачів веб-сервісу.

Для остаточного прийняття рішення з цього питання необхідно розрахувати внутрішню економічну дохідність E_v або показник внутрішньої норми дохідності вкладених інвестицій та порівняти її з так званою бар'єрною ставкою дисконтування, яка визначає ту мінімальну внутрішню економічну дохідність, нижче якої інвестиції в будь-яку науково-технічну розробку персоналізованої системи рекомендації контенту на основі поведінкових даних користувачів веб-сервісу вкладати буде економічно недоцільно.

Внутрішня економічна дохідність інвестицій E_v , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки персоналізованої системи рекомендації контенту на основі поведінкових даних користувачів веб-сервісу, розраховується за формулою:

$$E_v = \sqrt[T_{ж}]{1 + \frac{E_{абс}}{PV}} = \sqrt[3]{1 + \frac{3084703}{7224320}} = 0,48,$$

де $T_{ж}$ — життєвий цикл розробки персоналізованої системи рекомендації контенту на основі поведінкових даних користувачів веб-сервісу, роки.

Далі розраховуємо період окупності інвестицій T_o , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки персоналізованої системи рекомендації контенту на основі поведінкових даних користувачів веб-сервісу:

$$T_o = \frac{1}{E_v} = \frac{1}{0,48} = 2,1 \text{ роки.}$$

Оскільки $T_o < 1 \dots 3$ -х років, то це свідчить про комерційну привабливість науково-технічної розробки персоналізованої системи рекомендації контенту на

основі поведінкових даних користувачів веб-сервісу і може спонукати потенційного інвестора профінансувати впровадження цієї розробки персоналізованої системи рекомендації контенту на основі поведінкових даних користувачів веб-сервісу та виведення її на ринок.

Проведений комерційний та технологічний аудит персоналізованої системи рекомендації контенту на основі поведінкових даних користувачів веб-сервісу показав, що її науково-технічний рівень та комерційний потенціал оцінюються як вищі за середні. Система є актуальною для сучасного цифрового середовища, оскільки підвищує ефективність взаємодії користувачів із сервісом, сприяє зростанню залученості аудиторії та комерційної результативності.

Розрахунок витрат на розробку та комерціалізацію показав загальні витрати у розмірі 7,22 млн грн, а очікуваний економічний ефект для потенційного інвестора склав 3,08 млн грн із внутрішньою нормою доходності 48% та терміном окупності 2,1 року. Це свідчить про високу комерційну привабливість розробки та її перспективність для впровадження на ринок.

ВИСНОВКИ

У магістерській кваліфікаційній роботі розроблено гібридну систему контенту на основі поведінкових даних користувачів веб-сервісу.

В роботі проведено порівняльний аналіз сучасних підходів та алгоритмів до прогнозування поведінки користувачів рекомендаційних систем, зокрема колаборативної фільтрації, контентних моделей, за результатами якого обґрунтовано доцільність використання методу матричної факторизації SVD як базового алгоритму.

Розроблено функціональну схему системи персональних рекомендацій контенту, яка містить загальну логіку роботи системи та взаємодію її основних компонентів. Застосовано гібридний підхід, що поєднує поведінкові та контентні ознаки.

Розроблено архітектуру рекомендаційної системи, включно з моделями даних, схемами взаємодії компонентів та механізмами інтеграції контентних векторів у процес ініціалізації латентних факторів. Згідно даної архітектури розроблено програмний застосунок для реалізації системи персоналізованих рекомендацій контенту мовою Python, а також проведено його інтеграцію в REST-API.

Виконана гіперпараметрична оптимізація алгоритму сингулярного розкладання матриці SVD, яка дала можливість визначити оптимальні параметри навчання моделі, що забезпечують підвищення точності прогнозів.

Проведене тестування роботи системи показало зменшення RMSE на 0,016, що у масштабі вибірки з сотень тисяч взаємодій відповідає підвищенню точності прогнозування на 1,8%. Також отримано зростання метрик Precision приблизно на 6% та Recall на 5% порівняно з базовими алгоритмами SVD та k-NN. Застосована інтеграція контентних ознак забезпечила зниження негативного впливу проблеми «холодного старту»: для нових об'єктів початкова похибка стала меншою приблизно на 12%, а стабільна збіжність моделі досягалася від 15% до 20% швидше.

Отже, розроблена система персоніфікованих рекомендацій контенту

дозволяє формувати більш інформативні латентні представлення елементів завдяки застосуванню контентної ініціалізації до алгоритму SVD. Така системи може бути впроваджена у реальні веб-сервіси для підвищення рівня персоналізації, залученості користувачів та якості наданих рекомендацій.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Інтеграція контентних ознак при ініціалізації латентних факторів в системах персоналізованих рекомендацій / Воронов. В. В., Войцеховська О. В. // Матеріали міжнародної науково-практичної інтернет-конференції «Молодь в науці: дослідження, проблеми, перспективи (МН-2026)» (Вінниця, 2026р.) — [Електронний ресурс]. — Режим доступу: <https://conferences.vntu.edu.ua/index.php/mn/mn2026/paper/viewFile/26781/21995>
2. Aggarwal C. C. Recommender Systems: The Textbook. Cham : Springer, 2016. 498 p.
3. Adomavicius G., Tuzhilin A. Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions // IEEE Transactions on Knowledge and Data Engineering. 2005. Vol. 17, № 6. P. 734–749.
4. Мелешко Ю. Є. Проблеми сучасних рекомендаційних систем та методи їх вирішення — [Електронний ресурс]. — Режим доступу: <https://journals.nupp.edu.ua/sunz/article/view/1211> (дата звернення: 08.10.2025).
5. Netflix Technology Blog. Recommender System Case Studies — [Електронний ресурс]. — Режим доступу: <https://netflixtechblog.com> (дата звернення: 08.10.2025).
6. Шевченко С. М. Аналіз та дослідження характеристик існуючих моделей рекомендаційних систем — [Електронний ресурс]. — Режим доступу: https://journals.kntu.kherson.ua/index.php/visnyk_kntu/article/view/817 (дата звернення: 08.10.2025).
7. Bobadilla J., Ortega F., Hernando A., Gutiérrez A. Recommender Systems Survey // Knowledge-Based Systems. 2013. Vol. 46. P. 109–132.
8. Burke R. Hybrid Recommender Systems: Survey and Experiments // User Modeling and User-Adapted Interaction. 2002. Vol. 12, № 4. P. 331–370.
9. Thiagarajan K. Exploring Multimodal Large Language Models for Next-Generation Recommendation Systems — [Електронний ресурс]. —

Режим доступу: <https://doi.org/10.14445/22312803/ijctt-v73i2p108>
(дата звернення: 11.10.2025).

10. Цехмейстук Р. Рекомендаційні системи новинних платформ: типологія, принципи роботи та вплив на контент — [Електронний ресурс]. — Режим доступу: <https://molodyivchenyi.ua> (дата звернення: 11.10.2025).

11. Desrosiers C., Karypis G. A Comprehensive Survey of Neighborhood-based Recommendation Methods // *Recommender Systems Handbook*. Springer, 2011. P. 107–144.

12. Koren Y., Bell R., Volinsky C. Matrix Factorization Techniques for Recommender Systems // *IEEE Computer*. 2009. Vol. 42, № 8. P. 30–37.

13. Covington P., Adams J., Sargin E. Deep Neural Networks for YouTube Recommendations // *Proceedings of the 10th ACM Conference on Recommender Systems (RecSys)*. 2016. P. 191–198. DOI: 10.1145/2959100.2959190.

14. Arshi P., Muhammad A. K., Rukhsana Z., Huma A., Muhammad A., Muhammad M. F. Vision Transformers in Medical Computer Vision // *Engineering Applications of Artificial Intelligence*. 2023. Vol. 122. P. 1–10. DOI: 10.1016/j.engappai.2023.106126.

15. Lee D., Seung H. Learning the Parts of Objects by Non-Negative Matrix Factorization — [Електронний ресурс]. — Режим доступу: <https://www.nature.com/articles/44565> (дата звернення: 15.10.2025).

16. Jolliffe I. T., Cadima J. Principal Component Analysis: A Review and Recent Developments — [Електронний ресурс]. — Режим доступу: <https://royalsocietypublishing.org/doi/10.1098/rsta.2015.0202> (дата звернення: 26.10.2025).

17. Литвин В. В., Різник В. В. Алгоритмічне забезпечення рекомендаційних систем на основі машинного навчання — [Електронний ресурс]. — Режим доступу: <https://science.lpnu.ua/csit> (дата звернення: 30.10.2025).

18. Ковальчук А. П., Слободянюк С. О. Використання гібридних моделей у рекомендаційних системах — [Електронний ресурс]. — Режим доступу:

<https://praci.vntu.edu.ua/index.php/praci/article/view/588>

(дата звернення: 30.10.2025).

19. Поліщук М. О., Дяченко Ю. В. Використання матричної факторизації для побудови рекомендаційних систем — [Електронний ресурс]. — Режим доступу: <https://jrnl.nau.edu.ua/index.php/visnik/article/view/17476> (дата звернення: 02.11.2025).

20. Goodfellow I., Bengio Y., Courville A. Deep Learning. Cambridge : MIT Press, 2016. 775

21. Adomavicius G., Tuzhilin A. Toward the Next Generation of Recommender Systems: A Survey // IEEE Transactions on Knowledge and Data Engineering. 2005. Vol. 17, № 6. P. 734–749.

22. Pedregosa F. et al. Scikit-learn: Machine Learning in Python // Journal of Machine Learning Research. 2011. Vol. 12. P. 2825–2830.

23. Géron A. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. 3rd ed. O'Reilly Media, 2022. 856 p.

24. Manning C. D., Raghavan P., Schütze H. Introduction to Information Retrieval. Cambridge University Press, 2008. 482 p.

25. JetBrains. PyCharm — The Python IDE for Professional Developers. — [Електронний ресурс]. — Режим доступу: <https://www.jetbrains.com/pycharm/> (дата звернення: 12.11.2025).

26. Han J., Kamber M., Pei J. Data Mining: Concepts and Techniques. 3rd ed. Elsevier, 2012. 744 p.

27. Чорний О. В. Методи тестування моделей та валідації даних у прогнозних системах. Східноєвропейський журнал передових технологій. 2020. № 2. С. 35–42.

28. Salakhutdinov R., Mnih A. Probabilistic Matrix Factorization. Advances in Neural Information Processing Systems (NIPS). 2008. P. 1257–1264.

29. Шевченко В. О. Розробка API-сервісу для інтелектуальної обробки даних. Вісник ХНУРЕ, 2022, № 3. С. 112–119.

ДОДАТОК А**Технічне завдання**

Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра обчислювальної техніки

ЗАТВЕРДЖУЮ
Завідувач кафедри ОТ
проф., д.т.н. Азаров О. Д.
«03» жовтня 2025р.

ТЕХНІЧНЕ ЗАВДАННЯ

на виконання магістерської кваліфікаційної роботи
«Система персоналізованих рекомендацій контенту на основі поведінкових
даних користувачів веб-сервісу»

Науковий керівник: к.т.н., доц. каф. ОТ

Войцеховська О.В.

Студент групи 2КІ-24м

Воронов В.В.

1 Підстава для виконання магістерської кваліфікаційної роботи

1.1 Актуальність теми дослідження зумовлена необхідністю інтеграції високоефективних систем персоналізованих рекомендацій у комерційні та інформаційні платформи. З розвитком технологій та систем машинного навчання, якісне алгоритмічне забезпечення стає критично важливим для забезпечення роботи комплексних систем, що збирають та обробляють великі обсяги розріджених даних. Інновації у даній області дозволяють створювати більш точні та повні системи рекомендацій, які здатні ефективно вирішувати проблему "холодного старту" для нового контенту, тобто проблему нестачі достатніх початкових даних.

1.2 Наказ про затвердження теми МКР.

2 Мета МКР і призначення розробки

2.1 Мета роботи – підвищення точності прогнозів та зменшення рекомендацій нерелевантного контенту через проблему “холодного старту” для користувача веб-сервісу;

2.2 Мета розробки – персоніфікована рекомендація контенту на основі поведінкових даних користувача веб-сервісу.

3 Вихідні дані для виконання МКР: датасет goodbooks-10k, що містить 6 мільйонів явних рейтингів за шкалою 1–5; метадані контенту (назви, автори, теги), сучасні технології розробки систем рекомендацій.

4 Вимоги до виконання МКР

Головна вимога – створити систему персоніфікованих рекомендацій на основі поведінкових даних користувача веб-сервісу, яка вирішує проблема “холодного старту”.

5 Етапи МКР та очікувані результати

Етапи роботи та очікувані результати наведено в таблиці 1.

6 Матеріали, що подаються до захисту МКР

До захисту подаються: пояснювальна записка МКР, ілюстративні матеріали, протокол попереднього захисту МКР на кафедрі, відгук наукового керівника, анотації до МКР українською та іноземною мовами, довідка про відповідність оформлення МКР діючим вимогам.

Таблиця А.1 — Етапи МКР

№	Назва та зміст етапу	Термін виконання		Примітка
		Початок	Закінчення	
1	2	3	4	5
1	Вибір, узгодження та затвердження теми МКР	25.09.2025	30.09.2025	
2	Огляд шляхів вирішення науково-технічної задачі	01.10.2025	05.10.2025	
3	Теоретичні дослідження, проєктування технології або програмного засобу	06.10.2025	14.10.2025	
4	Практична реалізація технології рекомендацій контенту	15.10.2025	20.10.2025	
5	Перевірка працездатності та тестування сервісу рекомендацій	21.10.2025	22.10.2025	
6	Оформлення роботи у вигляді пояснювальної записки та презентації	23.10.2025	03.11.2025	
7	Перевірка виконаної роботи	04.11.2025	06.11.2025	
8	Остаточні виправлення, збирання підписів та зшивання роботи	07.11.2025	10.11.2025	
9	Попередній захист роботи та її рецензування	12.11.2025	12.11.2025	
10	Захист МКР	16.12.2025	22.12.2025	

7 Порядок контролю виконання та захисту МКР

Виконання етапів графічної та розрахункової документації МКР контролюється науковим керівником згідно зі встановленими термінами. Захист МКР відбувається на засіданні Екзаменаційної комісії, затвердженої наказом ректора.

8 Вимоги до оформлення та порядок виконання МКР

8.1 При оформлюванні МКР використовуються:

- ДСТУ 3008 : 2015 «Звіти в сфері науки і техніки. Структура та правила оформлювання»;
- ДСТУ 8302 : 2015 «Бібліографічні посилання. Загальні положення та правила складання»;
- ГОСТ 2.104-2006 «Єдина система конструкторської документації. Основні написи»;
- «Положення про кваліфікаційні роботи на другому (магістерському) рівні вищої освіти СУЯ ВНТУ-03.02.02-П.001.01:21»
- документами на які посилаються у вище вказаних.

ДОДАТОК Б

ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Назва роботи: Система персоналізованих рекомендацій контенту на основі поведінкових даних користувачів веб-сервісу.

Тип роботи: _____ магістерська кваліфікаційна робота _____
(бакалаврська кваліфікаційна робота / магістерська кваліфікаційна робота)

Підрозділ: кафедра обчислювальної техніки
(кафедра, факультет, навчальна група)

Коефіцієнт подібності текстових запозичень, виявлених у роботі системою StrikePlagiarism (КП1) 1,5 %

Висновок щодо перевірки кваліфікаційної роботи (відмітити потрібне)

Запозичення, виявлені у роботі, оформлені коректно і не містять ознак академічного плагіату, фабрикації, фальсифікації. Роботу прийняти до захисту.

У роботі не виявлено ознак плагіату, фабрикації, фальсифікації, але надмірна кількість текстових запозичень та/або наявність типових розрахунків не дозволяють прийняти рішення про оригінальність та самостійність її виконання. Роботу направити на доопрацювання.

У роботі виявлено ознаки академічного плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень. Робота до захисту не приймається.

Експертна комісія:

Азаров О.Д., д.т.н., проф., зав. кафедри ОТ
(прізвище, ініціали, посада)

Мартинюк Т. Б., д.т.н., проф. кафедри ОТ
(прізвище, ініціали, посада)

Особа, відповідальна за перевірку Захарченко С. М., проф. каф. ОТ
(підпис) (прізвище, ініціали)

З висновком експертної комісії ознайомлений(-на)

Керівник _____ Войцеховська О. В., доц. каф. ОТ
(підпис) (прізвище, ініціали, посада)

Здобувач _____ Воронов В.В. _____
(підпис) (прізвище, ініціали)

ДОДАТОК В

Блок схема роботи алгоритму роботи системи персоналізованих рекомендацій
контенту

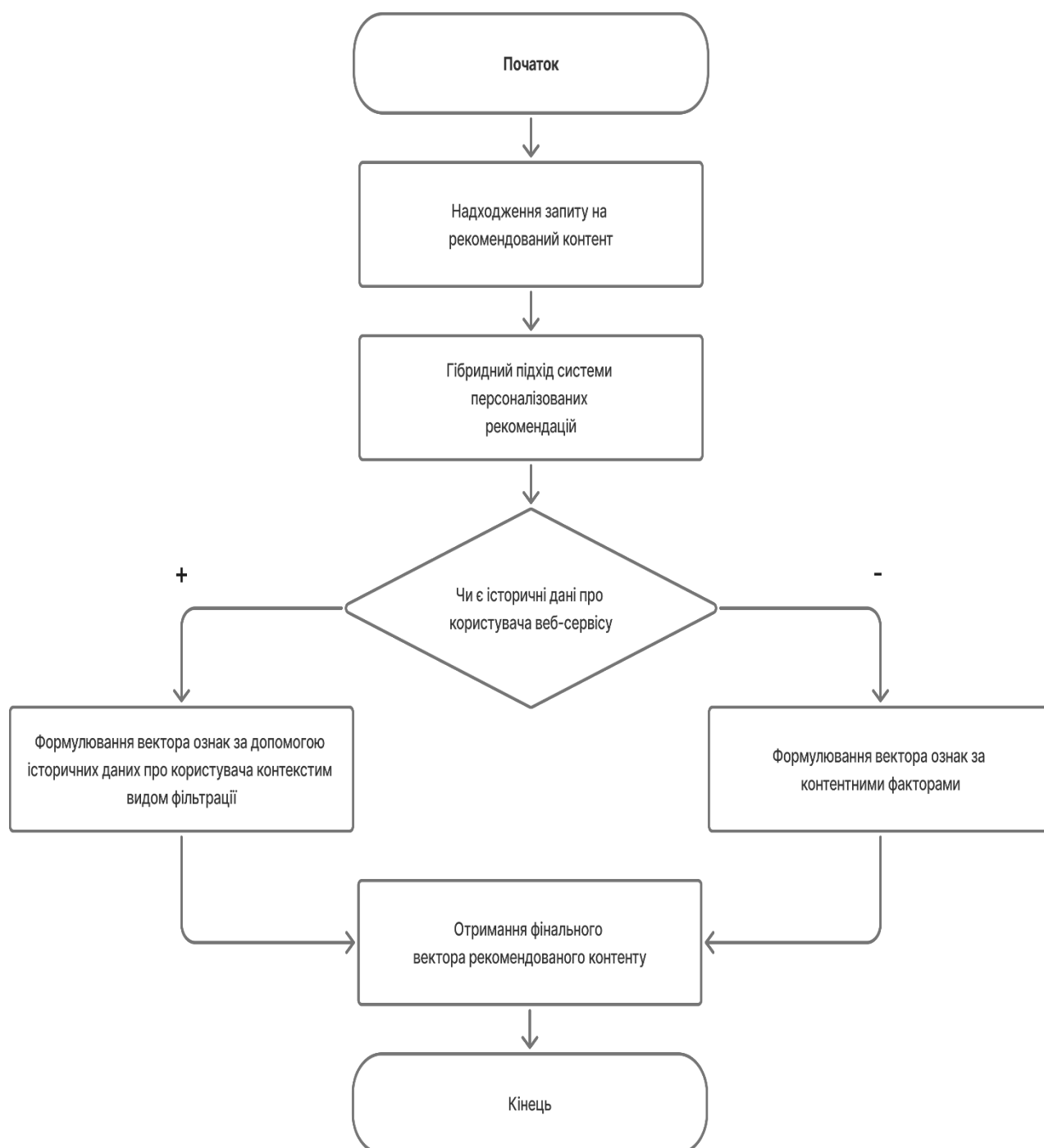


Рисунок В.1 — Блок схема алгоритму роботи системи персоналізованих
рекомендацій контенту

ДОДАТОК Г

Функціональна схема системи персоналізованих рекомендацій

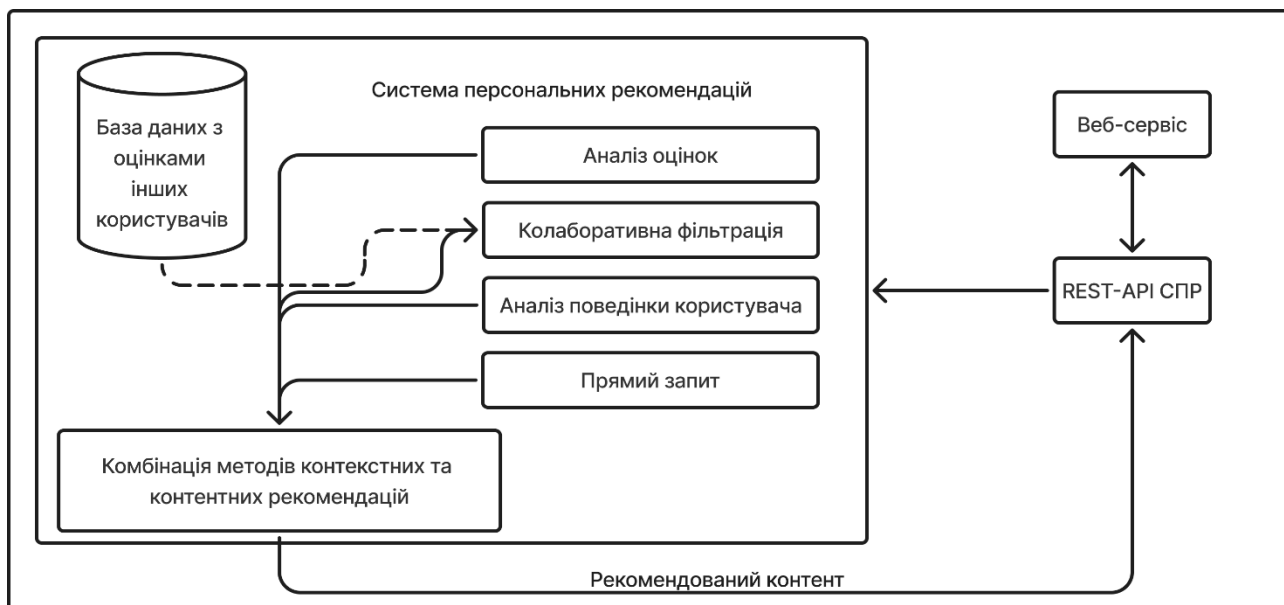


Рисунок Г.1 — Функціональна схема системи персоналізованих рекомендацій

ДОДАТОК Д

Схема взаємодії основних програмних компонентів

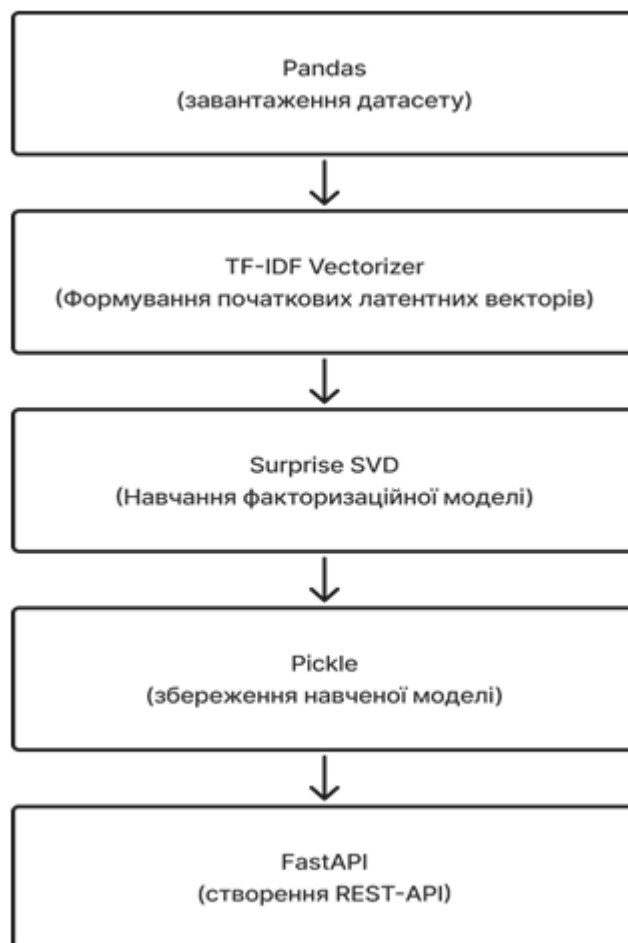


Рисунок Д.1 — Схема взаємодії основних програмних компонентів

ДОДАТОК Е

Лістинги коду

Лістинг Е.1 — Код моделі

```
import pandas as pd
from surprise import Dataset, Reader
from surprise.model_selection import train_test_split
from surprise import SVD, accuracy
from collections import defaultdict
import numpy as np
FILE_RATING = 'data/ratings.csv'
FILE_BOOKS = 'data/books.csv'
K_RECOMMENDED = 10
RELEVANCE_THRESHOLD = 4.0
TEST_USER_ID = 100
print("--- 1. Завантаження та підготовка даних ---")
try:
    ratings_df = pd.read_csv(FILE_RATING)
    books_df = pd.read_csv(FILE_BOOKS)
except FileNotFoundError:
    print(f"Помилка: Не знайдено файли '{FILE_RATING}' або '{FILE_BOOKS}'")
    print("Переконайтеся, що вони знаходяться у правильному каталозі.")
    exit()
min_rating = ratings_df['rating'].min()
max_rating = ratings_df['rating'].max()
reader = Reader(rating_scale=(min_rating, max_rating))
print(ratings_df.head(10))
data = Dataset.load_from_df(ratings_df[['user_id', 'book_id', 'rating']], reader)
print(f"Дані завантажено. Шкала рейтингів: {min_rating} - {max_rating}")
print("\n--- 2. Розділення даних та навчання моделі SVD ---")
trainset, testset = train_test_split(data, test_size=0.2, random_state=42)
model = SVD(n_epochs=25, lr_all=0.005, reg_all=0.4, random_state=42,
verbose=False)
model.fit(trainset)
print("Навчання моделі SVD завершено.")
print("Збереження моделі...")
with open("model_svd.pkl", "wb") as f:
    pickle.dump(model, f)
print("OK: model_svd.pkl створено!")
def precision_recall_at_k(predictions, k=10, threshold=4.0):
    user_est_true = defaultdict(list)
    for uid, _, true_r, est, _ in predictions:
```

```

    user_est_true[uid].append((est, true_r))
precisions = []
recalss = []
for uid, user_ratings in user_est_true.items():
    user_ratings.sort(key=lambda x: x[0], reverse=True)
    n_rel_and_rec_k = sum((true_r >= threshold) for (_, true_r) in user_ratings[:k])
    n_rel = sum((true_r >= threshold) for (_, true_r) in user_ratings)
    n_rec_k = min(k, len(user_ratings))
    if n_rec_k > 0:
        precisions.append(n_rel_and_rec_k / n_rec_k)
    if n_rel > 0:
        recalss.append(n_rel_and_rec_k / n_rel)
avg_precision = np.mean(precisions) if precisions else 0
avg_recall = np.mean(recalss) if recalss else 0
return avg_precision, avg_recall
predictions = model.test(testset)
print("\n--- 4. Оцінка якості та метрики ---")
rmse = accuracy.rmse(predictions, verbose=True)
precision, recall = precision_recall_at_k(predictions, k=K_RECOMMENDED,
threshold=RELEVANCE_THRESHOLD)
print(f"\n💡 Метрики Топ-{K_RECOMMENDED} (Порог релевантності:
{RELEVANCE_THRESHOLD})")
print(f"**Precision@{K_RECOMMENDED} (Точність)**: {precision:.4f}")
print(f"**Recall@{K_RECOMMENDED} (Повнота)**: {recall:.4f}")
print("-----")
def get_top_n_recommendations(user_id, model, n=5, all_ratings=ratings_df,
all_books=books_df):
    user Rated items = set(all_ratings[all_ratings['user_id'] == user_id]['book_id'])
    all_book_ids = set(all_ratings['book_id'].unique())
    unrated_books = all_book_ids - user Rated items
    predictions = []
    for book_id in unrated_books:
        pred = model.predict(user_id, book_id)
        predictions.append((book_id, pred.est))
    top_n = sorted(predictions, key=lambda x: x[1], reverse=True)[:n]
    book_titles = all_books.set_index('book_id')['title'].to_dict()
    top_n_titles = []
    for book_id, predicted_rating in top_n:
        title = book_titles.get(book_id, f"Книга з ID {book_id}")
        top_n_titles.append((title, predicted_rating))
    print(f"\nТоп-{n} рекомендацій для користувача **{user_id}**:")
    for title, rating in top_n_titles:
        print(f" - **{title}** (Прогноз: {rating:.2f})")
get_top_n_recommendations(TEST_USER_ID, model, n=5)

```

Лістинг Е.2 — Код API

```

import pandas as pd
from fastapi import FastAPI
import pickle
print("Завантаження моделі...") with open("model_svd.pkl", "rb") as f: model =
pickle.load(f)
ratings_df = pd.read_csv("data/ratings.csv") books_df =
pd.read_csv("data/books.csv")
book_titles = books_df.set_index("book_id")["title"].to_dict()
def get_top_n_recommendations(user_id, n=10):
    user Rated = set(ratings_df[ratings_df["user_id"] == user_id]["book_id"])
    all_books = set(ratings_df["book_id"].unique()) unrated = all_books -
    user Rated
    predictions = []
    for book_id in unrated:
        pred = model.predict(user_id, book_id)
        predictions.append((book_id, pred.est))
    top_n = sorted(predictions, key=lambda x: x[1], reverse=True)[:n]
    result = []
    for book_id, rating in top_n:
        title = book_titles.get(book_id, f"Book {book_id}")
        result.append({"book_id": int(book_id), "title": title, "predicted_rating":
        float(rating)})
    return result
app = FastAPI(title="Book Recommendation API")
@app.get("/recommend") def recommend(user_id: int, n: int = 10): return {
"user_id": user_id, "top_n": get_top_n_recommendations(user_id, n) }

```