

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Розробка інтелектуальної інформаційної системи аналізу та
прогнозування даних із використанням методів машинного навчання та
оптимізації моделей

Виконав: студент IV курсу, групи СНС-41

спеціальності 122 Комп'ютерні науки

(шифр і назва спеціальності)

Заяць Е.І.

(підпис)

(прізвище та ініціали)

Керівник

Марценко С.В.

(підпис)

(прізвище та ініціали)

Нормоконтроль

Шимчук Г.В.

(підпис)

(прізвище та ініціали)

Завідувач кафедри

Боднарчук І.О.

(підпис)

(прізвище та ініціали)

Рецензент

(підпис)

(прізвище та ініціали)

Тернопіль
2026

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Боднарчук І.О.
(підпис) (прізвище та ініціали)

« » червня 2026 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня Бакалавр
(назва освітнього ступеня)

за спеціальністю 122 Комп'ютерні науки
(шифр і назва спеціальності)

Студенту Заяць Ельвірі Ігорівні
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка інтелектуальної системи аналізу та прогнозування даних із використанням методів машинного навчання та оптимізації моделей

Керівник роботи Марценко Сергій Володимирович, к.т.н., доцент кафедри КН
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «14» травня 2026 року № 4/9-237

2. Термін подання студентом завершеної роботи 22 червня 2026 р.

3. Вихідні дані до роботи Методичні вказівки до виконання кваліфікаційної роботи, наукові праці з питань побудови рекомендаційних систем і застосування методів машинного навчання.

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ. 1. Аналіз методів побудови інтелектуальних рекомендаційних систем. 2. Проєктування інтелектуальної інформаційної системи аналізу читацьких уподобань. 3. Програмна реалізація та дослідження розробленої системи. 4. Безпека життєдіяльності, основи охорони праці. Висновки. Перелік джерел.

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

Титульний слайд. 2. Актуальність теми дослідження. 3. Мета та завдання дослідження. 4.

Аналіз предметної області книжкових рекомендаційних систем. 5. Порівняльна

характеристика методів формування рекомендацій. 6. Архітектура інтелектуальної

інформаційної системи аналізу читацьких уподобань. 7. Структура та підготовка набору даних

для машинного навчання. 8. Алгоритм пошуку книжкового двійника користувача. 9.

Структура бази даних інформаційної системи. 10. Вебінтерфейс інформаційної системи. 11.

Результати експериментального оцінювання рекомендаційних моделей. 12. Висновки. 13.

Завершальний слайд.

АНОТАЦІЯ

Розробка інтелектуальної системи аналізу та прогнозування даних із використанням методів машинного навчання та оптимізації моделей // Кваліфікаційна робота освітнього ступеня «Бакалавр» // Заяць Ельвіра Ігорівна // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних наук, група СНс-41 // Тернопіль, 2026 // С. 70, рис. – 15, табл. – 6, кресл. – 13, додат. – 0, бібліогр. – 45.

Ключові слова: рекомендаційна система, машинне навчання, книжкові рекомендації, колаборативна фільтрація, змістова фільтрація, SVD, книжковий двійник, персоналізація.

Кваліфікаційна робота присвячена дослідженню та розробленню інтелектуальної інформаційної системи аналізу читацьких уподобань і формування персоналізованих рекомендацій книг.

В першому розділі кваліфікаційної роботи описано предметну область книжкових рекомендаційних систем. Розглянуто основні підходи до формування рекомендацій. Проаналізовано переваги та недоліки колаборативної, змістової та гібридної фільтрації.

В другому розділі кваліфікаційної роботи досліджено вимоги до інформаційної системи, спроєктовано її архітектуру, структуру бази даних та процес машинного навчання. Подано алгоритм пошуку книжкового двійника користувача та методику експериментального оцінювання моделей рекомендацій.

В третьому розділі кваліфікаційної роботи описано програмну реалізацію системи рекомендації книг. Проаналізовано особливості реалізації алгоритмів персоналізації та вебінтерфейсу. Проведено експериментальне оцінювання рекомендаційних моделей і функціональне тестування розробленої системи.

ANNOTATION

Development of an Intelligent Information System for Analysis and Forecasting of Data Using Machine Learning Methods and Model Optimization // Qualification work of the educational level «Bachelor» // Zaiats Elvira Ihorivna // Ternopil Ivan Pulyu National Technical University, Computer and Information Systems and Software Engineering Faculty, Computer Sciences Department, group SNs-41 // Ternopil, 2026 // P. 70, fig. – 15, tabl. – 6, chair. – 13, annexes. – 0, references – 45.

Keywords: recommendation system, machine learning, book recommendations, collaborative filtering, content-based filtering, SVD, book twin, personalization.

The qualification work is devoted to the research and development of an intelligent information system for analyzing readers' preferences and generating personalized book recommendations.

The first chapter describes the subject area of book recommendation systems. The main approaches to generating recommendations are considered. The advantages and disadvantages of collaborative filtering, content-based filtering and hybrid filtering are analyzed.

The second chapter investigates the requirements for the information system and presents the design of its architecture, database structure and machine learning process. An algorithm for finding a user's book twin and a methodology for the experimental evaluation of recommendation models are proposed.

The third chapter describes the software implementation of the book recommendation system. The features of implementing personalization algorithms and the web interface are analyzed. Experimental evaluation of recommendation models and functional testing of the developed system are carried out.

ПЕРЕЛІК СКОРОЧЕНЬ І ТЕРМІНІВ

API (англ. Application Programming Interface) – програмний інтерфейс застосунку.

MAE (англ. Mean Absolute Error) – середня абсолютна похибка.

RMSE (англ. Root Mean Square Error) – середньоквадратична похибка.

SVD (англ. Singular Value Decomposition) – сингулярний розклад матриці.

TF (англ. Term Frequency) – частота входження терміна в документ.

DF (англ. Document Frequency) – кількість документів, що містять певний термін.

TF-IDF (англ. Term Frequency – Inverse Document Frequency) – метод визначення ваги термінів у текстових документах.

CSV (англ. Comma-Separated Values) – формат зберігання табличних даних.

SQLite – вбудована реляційна система керування базами даних.

Streamlit – програмний фреймворк для створення вебзастосунків мовою Python.

Scikit-learn – бібліотека машинного навчання для мови Python.

SVD-модель – модель рекомендацій, побудована на основі сингулярного розкладу матриці оцінок.

RMSE-метрика – показник якості прогнозування, що характеризує середньоквадратичну похибку моделі.

MAE-метрика – показник якості прогнозування, що характеризує середню абсолютну похибку моделі.

ШІ – штучний інтелект.

ЗМІСТ

ВСТУП	8
РОЗДІЛ 1. АНАЛІЗ МЕТОДІВ ПОБУДОВИ ІНТЕЛЕКТУАЛЬНИХ РЕКОМЕНДАЦІЙНИХ СИСТЕМ	10
1.1 Аналіз предметної області книжкових рекомендаційних систем	10
1.2 Огляд наявних систем рекомендації книг	13
1.3 Методи побудови книжкових рекомендацій	15
1.4 Постановка завдання дослідження	20
1.5 Висновок до першого розділу	24
РОЗДІЛ 2. ПРОЄКТУВАННЯ ІНТЕЛЕКТУАЛЬНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ АНАЛІЗУ ЧИТАЦЬКИХ УПОДОБАНЬ	25
2.1 Формування вимог до інформаційної системи	25
2.2 Проєктування архітектури системи	28
2.3 Характеристика та підготовка набору даних	30
2.4 Проєктування процесу машинного навчання	32
2.5 Проєктування алгоритму пошуку книжкового двійника	36
2.6 Проєктування бази даних і локальних профілів	39
2.7 Методика експериментального оцінювання	40
2.8 Висновок до другого розділу	42
РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ДОСЛІДЖЕННЯ РОЗРОБЛЕНОЇ СИСТЕМИ	43
3.1 Обґрунтування вибору програмних засобів	43
3.2 Реалізація підготовки даних і моделей	44
3.3 Реалізація алгоритмів персоналізації	47
3.4 Реалізація локальних профілів і збереження даних	48
3.5 Розроблення вебінтерфейсу інформаційної системи	50
3.6 Експериментальне оцінювання рекомендаційних моделей	54
3.7 Функціональне тестування та аналіз результатів	55
3.8 Висновок до третього розділу	55

РОЗДІЛ 4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ	57
4.1 Характеристика життєдіяльності людини у системі «людина – машина – середовище існування».....	57
4.2 Інформація в системі управління охороною праці	59
4.3 Висновок до четвертого розділу	60
ВИСНОВКИ.....	62
ПЕРЕЛІК ДЖЕРЕЛ.....	65

ВСТУП

Актуальність теми. У сучасних умовах стрімкого зростання обсягів цифрового контенту користувачам дедалі складніше швидко знаходити інформаційні об'єкти, що відповідають їхнім індивідуальним потребам. Особливо актуальною ця проблема є у сфері електронних бібліотек, книжкових платформ та інтернет-книгарень, де кількість доступних творів постійно збільшується. Традиційний пошук за назвою, автором або жанром не завжди є достатнім, оскільки читач часто не має чітко сформованого запиту, а лише прагне знайти книгу, що відповідатиме його читацьким уподобанням.

Одним із ефективних способів розв'язання цієї проблеми є використання рекомендаційних систем. Такі системи дають змогу аналізувати попередні оцінки, жанрові вподобання, метадані книг та інші характеристики для формування персоналізованих рекомендацій. Застосування методів машинного навчання у книжкових рекомендаційних системах дозволяє підвищити точність добору творів, автоматизувати процес аналізу читацької поведінки та забезпечити зручніший вибір книг для користувача. Тому розроблення інтелектуальної інформаційної системи аналізу читацьких уподобань і рекомендації книг є актуальним завданням у галузі комп'ютерних наук.

Мета і задачі дослідження. Метою даної кваліфікаційної роботи освітнього рівня «Бакалавр» є розроблення інтелектуальної інформаційної системи аналізу читацьких уподобань та формування персоналізованих рекомендацій книг із використанням методів машинного навчання.

Для досягнення поставленої мети потрібно виконати ряд завдань, зокрема:

- Проаналізувати предметну область книжкових рекомендаційних систем;
- Розглянути наявні сервіси рекомендації книг та визначити їхні переваги й обмеження;
- Дослідити основні методи формування рекомендацій, зокрема колаборативну, змістову та гібридну фільтрацію;

- Сформувати вимоги до інтелектуальної інформаційної системи аналізу читацьких уподобань;
- Спроекувати архітектуру системи, структуру бази даних і процес машинного навчання;
- Розробити алгоритм пошуку книжкового двійника користувача;
- Реалізувати програмний застосунок для формування книжкових рекомендацій;
- Провести експериментальне оцінювання рекомендаційних моделей та функціональне тестування розробленої системи.

Об’єкт дослідження – процес формування персоналізованих книжкових рекомендацій на основі аналізу читацьких уподобань користувачів.

Предмет дослідження – методи та алгоритми побудови рекомендаційних систем, засоби машинного навчання та програмні компоненти інформаційної системи рекомендації книг.

Практичне значення одержаних результатів полягає у створенні програмної системи, яка дає змогу користувачеві отримувати персоналізовані рекомендації книг, знаходити книжкового двійника зі схожими читацькими вподобаннями та переглядати результати через вебінтерфейс. Розроблена система може бути використана як основа для подальшого вдосконалення книжкових сервісів, електронних бібліотек або навчальних проєктів, пов’язаних із рекомендаційними алгоритмами та аналізом користувацьких даних.

РОЗДІЛ 1. АНАЛІЗ МЕТОДІВ ПОБУДОВИ ІНТЕЛЕКТУАЛЬНИХ РЕКОМЕНДАЦІЙНИХ СИСТЕМ

1.1 Аналіз предметної області книжкових рекомендаційних систем

Стрімке зростання кількості цифрового контенту істотно ускладнює пошук інформаційних об'єктів, що відповідають індивідуальним потребам користувача. Ця проблема особливо помітна в електронних бібліотеках, інтернет-книгарнях і книжкових соціальних мережах, де користувачеві необхідно вибрати одну або декілька книг із великого каталогу. Традиційний пошук за назвою, автором або жанром є ефективним лише тоді, коли читач заздалегідь знає, який саме твір він бажає знайти. Якщо ж метою є відкриття нової книги, яка відповідає сформованим читацьким уподобанням, звичайних засобів пошуку недостатньо.

Для розв'язання проблеми інформаційного перевантаження застосовують рекомендаційні системи. Вони являють собою програмні засоби, призначені для автоматизованого формування пропозицій щодо об'єктів, які потенційно можуть бути корисними або цікавими конкретному користувачеві [1].

Такими об'єктами можуть бути товари, музичні композиції, фільми, освітні матеріали, новини або книги. На відміну від пошукових систем, рекомендаційна система не лише реагує на сформульований запит, а й прогнозує інтереси користувача на основі доступної інформації про його попередню поведінку.

Основними даними, які використовуються у книжкових рекомендаційних системах, є оцінки користувачів, історія прочитаних книг, списки бажаного читання, автори, жанри, тематичні теги, текстові описи та інші метадані. Оцінки відображають явний зворотний зв'язок, оскільки користувач безпосередньо вказує рівень свого ставлення до книги. Додавання твору до списку читання, перегляд сторінки або вибір обкладинки можуть розглядатися як неявний зворотний зв'язок. Явні оцінки є інформативнішими, однак їх кількість зазвичай

обмежена. Неявні дії фіксуються частіше, проте не завжди однозначно свідчать про позитивне ставлення користувача.

У загальному випадку процес формування книжкових рекомендацій включає збирання даних, побудову профілю користувача, визначення відповідності між користувачем і книгами, ранжування кандидатів та формування кінцевого списку. Узагальнену послідовність цього процесу подано на рисунку 1.1.



Рисунок 1.1 – Узагальнений процес формування персональних книжкових рекомендацій

На першому етапі система отримує дані про книги та взаємодії користувачів із ними. Після очищення й перетворення даних формується профіль, який відображає явні та приховані читацькі інтереси. Далі алгоритм визначає книги-кандидати, оцінює ймовірний рівень зацікавленості користувача та впорядковує результати. Із ранжованого переліку вилучаються вже прочитані або оцінені твори, після чого користувачеві демонструється обмежена кількість найрелевантніших рекомендацій.

Такий підхід дає змогу не лише сформувати перелік потенційно цікавих книг, а й упорядкувати їх відповідно до очікуваного рівня зацікавленості користувача. У межах розроблюваної системи рекомендація розглядається як результат зіставлення профілю читача з множиною доступних книжкових об'єктів.

Формалізовано рекомендаційну задачу можна подати як прогноз функції корисності:

$$u: U \times I \rightarrow R, \quad (1.1)$$

де U – множина користувачів; I – множина інформаційних об'єктів, якими у межах цієї роботи є книги; R – множина можливих значень корисності; u – функція, що визначає очікуваний рівень відповідності книги конкретному користувачеві. Практичним результатом роботи системи є пошук для кожного користувача таких елементів множини I , для яких прогнозоване значення функції u є найбільшим.

Джерелом складності є розрідженість матриці взаємодій. Навіть активний читач оцінює лише незначну частину загального каталогу, тому для більшості пар «користувач – книга» явне значення оцінки відсутнє. Іншою проблемою є холодний старт, який виникає для нового користувача без історії оцінювання або для нової книги, щодо якої ще не накопичено взаємодій. Також рекомендаційні системи повинні підтримувати баланс між точністю та різноманітністю. Якщо алгоритм пропонує лише дуже схожі між собою твори, рекомендації можуть бути формально релевантними, але не сприяти відкриттю нових авторів і тематичних напрямів.

У сучасних системах застосовують колаборативну, змістову та гібридну фільтрацію. Колаборативні методи використовують закономірності в оцінках багатьох користувачів і можуть виявляти неочевидні зв'язки між книгами без детального аналізу їхнього змісту. Змістові методи порівнюють характеристики книг із профілем користувача. Гібридні моделі поєднують декілька джерел інформації та дозволяють частково компенсувати обмеження окремих підходів [2], [3].

Окремим напрямом персоналізації є пошук користувачів зі схожими вподобаннями. У розроблюваній системі цей підхід реалізовано як функцію «книжкового двійника». Двійником вважається користувач, чий оцінки найбільш

узгоджуються з оцінками активного читача на множині спільно прочитаних книг.

Предметна область розроблюваної системи охоплює зберігання книжкових метаданих, опрацювання користувацьких оцінок, формування індивідуального профілю, прогнозування оцінок непрочитаних книг, пошук книжкового двійника, створення рекомендацій та антирекомендацій, а також пояснення отриманих результатів. Важливою вимогою є можливість роботи не лише з користувачами навчального набору даних, а й із новими профілями, створеними безпосередньо в інформаційній системі. Таким чином, книжкова рекомендаційна система є інтелектуальним засобом підтримки прийняття рішень, який допомагає користувачеві орієнтуватися у великому каталозі та зменшує витрати часу на вибір наступної книги. Специфіка предметної області потребує врахування як колективних закономірностей в оцінках, так і змістових характеристик творів. Це обґрунтовує доцільність застосування гібридного підходу, доповненого пошуком користувача зі схожим читацьким профілем.

1.2 Огляд наявних систем рекомендації книг

Під час проектування інтелектуальної інформаційної системи доцільно проаналізувати наявні програмні рішення, призначені для пошуку, обліку та рекомендації книжок. Такий аналіз дає змогу визначити поширені підходи до персоналізації, виявити функціональні переваги аналогів і сформулювати вимоги до розроблюваної системи. У межах дослідження розглянуто сервіси Goodreads, The StoryGraph, LibraryThing і BookBub, які реалізують різні моделі взаємодії користувача з книжковим контентом.

Goodreads є книжковою соціальною платформою, що поєднує ведення особистої бібліотеки, оцінювання та рецензування творів, формування тематичних полиць і отримання персоналізованих рекомендацій. Соціальна складова платформи дає змогу переглядати активність інших читачів, брати участь у тематичних обговореннях і знайомитися з рецензіями [5].

The StoryGraph орієнтована на детальніший опис читацьких уподобань. Для формування персональних рекомендацій сервіс пропонує користувачеві вказати жанри, характерні настрої, бажаний темп оповіді. Платформа підтримує оцінювання з використанням дробових значень [6].

LibraryThing насамперед призначена для створення та ведення особистого книжкового каталогу. Система дає змогу додавати книги з бібліотечних і торговельних каталогів, редагувати метадані, використовувати власні теги та різні класифікаційні системи [7].

BookBub реалізує інший підхід до книжкової рекомендації. Сервіс орієнтований на пошук нових творів, спеціальних пропозицій та електронних видань. Рекомендації можуть надходити від авторів, редакторів платформи, друзів та інших користувачів, думці яких довіряє читач [8]. Розглянуті сервіси демонструють різні підходи до організації рекомендаційного процесу та використання даних користувачів.

Для узагальнення результатів аналізу в таблиці 1.1 наведено порівняння основних функціональних можливостей досліджених сервісів.

Таблиця 1.1 – Порівняння функціональних можливостей наявних книжкових сервісів

Система	Дані для персоналізації	Переваги	Обмеження
Goodreads	Оцінки, жанри, книжкові полиці	Рецензії, соціальні функції, рекомендації	Відсутнє детальне пояснення рекомендацій
The StoryGraph	Оцінки, жанри, настрої, темп	Гнучкі оцінки, статистика, фільтри	Потребує детального налаштування профілю
LibraryThing	Бібліотека, теги, оцінки	Каталогізація та робота з метаданими	Потребує наповненої особистої бібліотеки
BookBub	Жанри, автори, соціальні сигнали	Пошук новинок і спеціальних пропозицій	Орієнтація переважно на придбання книг

Порівняння показало, що наявні сервіси підтримують облік прочитаних книг, персоналізовані добірки та соціальну взаємодію, однак відрізняються джерелами даних і рівнем пояснюваності.

Отже, аналіз наявних книжкових сервісів показує, що персоналізація рекомендацій може ґрунтуватися на різних типах даних: оцінках користувачів, жанрових уподобаннях, тематичних тегах, книжкових полицях і соціальних взаємодіях. Для розроблюваної системи важливо не лише повторити базові можливості таких сервісів, а й забезпечити пояснюваність результатів та можливість пошуку користувача зі схожим читацьким профілем. Тому наступним етапом є розгляд методів побудови книжкових рекомендацій, які можуть бути використані для реалізації такої системи.

1.3 Методи побудови книжкових рекомендацій

Формування персоналізованих книжкових рекомендацій передбачає визначення ступеня відповідності між читацькими вподобаннями користувача та характеристиками доступних творів.

Колаборативна фільтрація ґрунтується на припущенні, що користувачі, які подібно оцінювали об'єкти в минулому, з певною ймовірністю матимуть схожі вподобання й надалі. Для формування рекомендацій використовують матрицю взаємодій, у рядках якої подано користувачів, у стовпцях – книги, а на перетині – поставлені ними оцінки. Більшість елементів такої матриці залишаються незаповненими, оскільки один користувач взаємодіє лише з невеликою частиною загального каталогу.

Розрізняють колаборативну фільтрацію на основі користувачів і на основі об'єктів. У першому випадку для активного користувача визначають читачів зі схожою історією оцінювання, після чого рекомендації формують на основі книг, позитивно оцінених його найближчими сусідами [9]. У другому випадку встановлюють подібність між книгами за характером отриманих оцінок, а

користувачеві пропонують об'єкти, схожі на ті, які він уже оцінив позитивно [10].

Для визначення подібності можуть застосовуватися коефіцієнт кореляції Пірсона, косинусна подібність та інші міри. Косинусна подібність між векторами оцінок двох користувачів визначається за формулою:

$$\text{sim}(u, v) = \frac{\sum_i r_{ui} r_{vi}}{\sqrt{\sum_i r_{ui}^2} \sqrt{\sum_i r_{vi}^2}}, \quad (1.2)$$

де $\text{sim}(u, v)$ – значення косинусної подібності між користувачами u та v ; r_{ui} та r_{vi} – оцінки, поставлені користувачами u та v книзі i відповідно.

Для зменшення впливу індивідуальних шкал оцінювання доцільно центрувати значення, віднімаючи від кожної оцінки середню оцінку відповідного користувача. Крім самого коефіцієнта подібності, необхідно враховувати кількість спільно оцінених книг: висока подібність, отримана за одним або двома збігами, є менш надійною, ніж результат, підтверджений значною кількістю спільних оцінок. У системі «Book Twin» цей принцип використовується під час пошуку книжкового двійника – користувача з найбільш узгодженим читацьким профілем.

Модельна колаборативна фільтрація використовує факторизацію матриці оцінок. Її мета полягає у поданні користувачів і книг у спільному просторі латентних ознак, які автоматично виявляються зі структури взаємодій. Одним із поширених методів є сингулярний розклад матриці SVD. Наближене подання матриці оцінок має вигляд:

$$R_k \approx U_k \Sigma_k V_k^T, \quad (1.3)$$

де R_k – наближена матриця оцінок; U_k – матриця латентних представлень користувачів; Σ_k – діагональна матриця сингулярних значень; V_k^T –

транспонована матриця латентних представлень книг; k – кількість збережених компонент.

Зменшення розмірності дає змогу компактно подати розріджені дані та відфільтрувати частину випадкового шуму. Прогнозована оцінка книги визначається на основі взаємного розташування латентних векторів користувача і твору. Матрична факторизація ефективно виявляє приховані закономірності у великих наборах оцінок [11], однак не використовує назву, автора, жанри або текстовий опис книги. Тому для нових творів без накопиченої історії взаємодій її можливості обмежені.

Змістова фільтрація, навпаки, використовує характеристики самих книг: авторство, жанри, тематичні теги, назву та інші метадані. Профіль користувача формується на основі ознак творів, яким він поставив високі оцінки, після чого система знаходить книги зі схожим представленням [12]. Для перетворення текстових характеристик у числові вектори застосовують метод TF-IDF, який підвищує вагу термінів, характерних для конкретного документа, і зменшує вплив слів, поширених у всій колекції [13].

$$\text{TF-IDF}(t, d) = \text{TF}(t, d) \cdot \log\left(\frac{N}{\text{DF}(t)}\right), \quad (1.4)$$

де $\text{TF}(t, d)$ – частота терміна t у документі d ; $\text{DF}(t)$ – кількість документів, у яких трапляється цей термін; N – загальна кількість документів у колекції. У книжковій системі документом є об'єднане текстове представлення книги. Подібність між книгою та профілем користувача визначається за допомогою косинусної міри.

Змістова модель може працювати з новими книгами за наявності їхніх метаданих і дає змогу пояснювати результат через спільні авторські, жанрові або тематичні ознаки. Її основним обмеженням є надмірна спеціалізація, коли користувачеві пропонуються переважно твори, дуже схожі на вже прочитані. Для компенсації обмежень окремих підходів використовують гібридні

рекомендаційні системи [14]. У книжкових рекомендаційних системах гібридний підхід також може поєднувати змістові характеристики творів із додатковими ознаками, зокрема авторським ранжуванням [4].

У розроблюваній системі підсумковий результат формується шляхом зваженого об'єднання колаборативної та змістової складових:

$$S_{\text{hybrid}}(u, i) = \alpha S_{\text{SVD}}(u, i) + (1 - \alpha) S_{\text{content}}(u, i), \quad (1.5)$$

де $S_{\text{hybrid}}(u, i)$ – підсумкова оцінка відповідності книги i користувачеві u ; $S_{\text{SVD}}(u, i)$ – нормалізований прогноз SVD-моделі; $S_{\text{content}}(u, i)$ – результат змістової моделі; α – ваговий коефіцієнт у межах від 0 до 1. Значення α визначається експериментально за результатами оцінювання якості рекомендацій.

Під час вибору методу формування рекомендацій необхідно враховувати не лише точність прогнозування, а й вимоги до вхідних даних, здатність працювати з новими користувачами та книгами, а також можливість пояснення отриманого результату. Колаборативні методи ефективно виявляють приховані зв'язки між оцінками, змістова фільтрація забезпечує більшу прозорість рекомендацій, а SVD дає змогу компактно подати розріджену матрицю взаємодій. Поєднання цих підходів у гібридній моделі дозволяє частково компенсувати їхні окремі обмеження та підвищити стійкість персоналізації.

У межах розроблюваної системи доцільно використовувати не один окремий метод, а комбінацію кількох підходів, оскільки кожен із них розв'язує лише частину загальної задачі персоналізації. Колаборативна складова враховує поведінку інших читачів, змістова модель аналізує характеристики самих книг, а гібридне поєднання цих результатів дає змогу сформувати більш збалансований список рекомендацій.

Таке порівняння дає змогу визначити, які алгоритмічні складові є найбільш доцільними для реалізації в системі «Book Twin». Крім того, воно допомагає обґрунтувати вибір гібридної моделі як основного підходу до формування персоналізованих книжкових рекомендацій.

Особливості розглянутих методів та їхню роль у системі «Book Twin» узагальнено в таблиці 1.2.

Таблиця 1.2 – Порівняння методів формування книжкових рекомендацій

Метод	Вхідні дані	Переваги	Обмеження	Роль у системі
Колаборативна фільтрація	Оцінки користувачів	Виявлення неочевидних зв'язків між читачами та книгами	Розрідженість даних і холодний старт	Пошук книжкового двійника
SVD	Матриця оцінок	Виявлення латентних факторів і компактне подання даних	Залежність від накопичених взаємодій	Прогнозування персональних оцінок
Змістова фільтрація	Автори, жанри, теги та інші метадані	Робота з новими книгами та зрозумілі пояснення	Надмірна спеціалізація рекомендацій	Визначення змістової подібності
Гібридна модель	Оцінки та метадані	Компенсація обмежень окремих моделей	Необхідність налаштування ваг	Формування підсумкового рейтингу

Порівняння показує, що жоден із розглянутих методів окремо не забезпечує повного розв'язання завдання персоналізації. Колаборативні моделі ефективно використовують закономірності в оцінках, але залежать від історичних взаємодій. Змістова модель враховує характеристики книг, однак може звужувати рекомендації до вже знайомих користувачеві напрямів.

Важливим етапом роботи гібридної моделі є приведення результатів окремих алгоритмічних складових до спільного масштабу. Така нормалізація необхідна, оскільки оцінки, сформовані SVD-моделлю, змістовою складовою та модулем пошуку книжкового двійника, мають різну природу й діапазони значень. Після нормалізації отримані сигнали зважуються та об'єднуються в єдину підсумкову оцінку, яка використовується для ранжування книг.

Завдяки такій організації система може враховувати як колективні закономірності в оцінках користувачів, так і змістові характеристики самих книг.

Це підвищує гнучкість рекомендаційного процесу та дає змогу формувати результати навіть у випадках, коли окремий метод має обмежену кількість вхідних даних.

Узагальнену взаємодію складових моделі наведено на рисунку 1.2.



Рисунок 1.2 – Схема формування рекомендацій гібридною моделлю

Колаборативна складова прогнозує оцінки непрочитаних книг, змістова визначає відповідність твору профілю, а модуль книжкового двійника додає сигнал від найбільш схожого читача. Після нормалізації результати об'єднуються, книги ранжуються, а користувачеві надаються рекомендації, антирекомендації та пояснення.

Антирекомендація не означає низьку якість книги, а лише її низьку прогнозовану релевантність для конкретного користувача. Отже, поєднання SVD, TF-IDF, гібридного ранжування та книжкового двійника дає змогу враховувати як оцінки, так і змістові характеристики книг.

1.4 Постановка завдання дослідження

За результатами аналізу предметної області, наявних книжкових сервісів і методів персоналізації сформульовано завдання розроблення інтелектуальної інформаційної системи «Book Twin». Система має забезпечувати аналіз історії читацьких оцінок, прогнозування індивідуального ставлення до непрочитаних книг і формування персоналізованих рекомендацій із поясненням причин отриманого результату. На відміну від сервісів, орієнтованих переважно на популярність або жанрову схожість, розроблюване рішення повинно одночасно

враховувати закономірності колективного оцінювання, змістові характеристики книжок і подібність між читацькими профілями.

Метою дослідження є розроблення та експериментальне обґрунтування інтелектуальної інформаційної системи аналізу й прогнозування читацьких уподобань із використанням методів машинного навчання та оптимізації параметрів гібридної рекомендаційної моделі. Досягнення поставленої мети передбачає не лише побудову алгоритмів прогнозування, а й створення завершеного програмного засобу, придатного для роботи з реальними користувацькими профілями.

Вхідними даними системи є метадані книжок, користувацькі оцінки та локальні профілі читачів. До метаданих належать ідентифікатор, назва, автор, рік видання, тематичні ознаки та інші доступні характеристики. Історія оцінювання подається у вигляді множини взаємодій «користувач – книга – оцінка». Для нового користувача початковий профіль формується на основі самостійно доданих оцінок, після чого система повинна оновлювати його представлення без повного перенавчання всіх моделей.

Вихідними результатами є впорядкований список рекомендованих книг, перелік антирекомендацій, прогнозовані значення персональної релевантності, інформація про знайденого книжкового двійника та короткі пояснення, сформовані на основі внеску окремих складових моделі. Антирекомендація повинна трактуватися як низька прогнозована відповідність твору конкретному користувачеві, а не як оцінювання об'єктивної якості книги.

Розроблювана система має поєднати всі зазначені функції в межах єдиного робочого процесу. Це забезпечить послідовне опрацювання даних від створення профілю користувача до отримання персоналізованого результату.

Окремі етапи роботи системи мають бути пов'язані між собою та виконуватися в чіткій послідовності. Це дасть змогу контролювати якість підготовки даних, роботу моделей і коректність формування кінцевих рекомендацій.

Для реалізації поставленої мети необхідно розв'язати сукупність взаємопов'язаних завдань. Їх узагальнення та очікувані результати наведено в таблиці 1.3.

Таблиця 1.3 – Основні завдання дослідження та очікувані результати

Завдання дослідження	Очікуваний результат
Підготовка та структуризація даних про книги й оцінки	Узгоджений набір даних, придатний для побудови моделей машинного навчання
Побудова колаборативної моделі на основі матричної факторизації	Прогнозування оцінок непрочитаних книг за прихованими закономірностями взаємодій
Формування змістового представлення книжок	Векторний опис творів за назвами, авторами, тегами та іншими метаданими
Розроблення гібридного алгоритму та оптимізація його ваг	Об'єднання колаборативного й змістового сигналів із найкращими значеннями метрик якості
Реалізація пошуку книжкового двійника	Визначення користувача з найбільш узгодженим профілем оцінювання та врахування його вподобань
Реалізація рекомендацій, антирекомендацій і пояснень	Формування зрозумілого для користувача персоналізованого результату
Створення інформаційної системи та експериментальне оцінювання	Працездатний веб-застосунок із локальними профілями, базою даних і підтвердженою ефективністю моделей

Під час підготовки даних необхідно вилучити дублікати, опрацювати пропущені значення, узгодити ідентифікатори книжок і сформувати матрицю користувацьких оцінок. Колаборативна складова повинна реалізовувати факторизацію цієї матриці та підтримувати отримання латентного представлення нового користувача. Змістова складова має перетворювати книжкові метадані у векторний простір і визначати їхню відповідність профілю читача.

Підсумковий рейтинг книги необхідно формувати шляхом зваженого об'єднання прогнозів моделей. Значення ваг мають визначатися не довільно, а за результатами експериментального пошуку на незалежній вибірці користувачів. Для оцінювання доцільно застосувати ранжувальні метрики $Precision@K$, $Recall@K$, $HitRate@K$ і $NDCG@K$, а отримані результати порівняти з колаборативною моделлю та базовою стратегією рекомендації популярних книг.

Функціональна частина системи повинна підтримувати створення й вибір користувацького профілю, пошук книжок, додавання, зміну та видалення оцінок, отримання рекомендацій і антирекомендацій, а також перегляд знайденого книжкового двійника. Збереження локальних профілів та історії оцінювання необхідно реалізувати у постійній базі даних. Інтерфейс має забезпечувати послідовну взаємодію з функціями системи та не вимагати від користувача знання внутрішньої будови моделей.

Для нового користувача система повинна формувати початковий профіль на основі введених ним оцінок. Після кожної наступної взаємодії профіль має уточнюватися, а результати рекомендації – оновлюватися. Збереження оцінок і налаштувань у базі даних дасть змогу продовжувати роботу з профілем під час наступних запусків застосунку.

Окремою вимогою є зрозуміле подання результатів. Для рекомендованих і нерекондованих книг доцільно відображати прогнозовану оцінку, внесок окремих моделей і дані про книжкового двійника. Це дозволить користувачеві зрозуміти причину появи конкретної книги у сформованій добірці.

Експериментальна перевірка повинна охоплювати коректність підготовки даних, роботу окремих моделей, алгоритм об'єднання прогнозів, збереження профілів і поведінку вебінтерфейсу. Результатом виконання поставленого завдання має стати програмно реалізована система, у якій теоретично обґрунтовані методи доведено до практичного використання та підтверджено кількісними показниками якості.

Окремо необхідно перевірити, як система реагує на додавання та зміну користувацьких оцінок. Після оновлення профілю рекомендації мають перераховуватися коректно, а збережені дані – залишатися доступними під час наступних запусків застосунку.

Таким чином, поставлене завдання охоплює повний цикл роботи системи – від підготовки даних до оцінювання якості сформованих рекомендацій. Це дає змогу перейти до проєктування архітектури, структури даних і взаємодії основних програмних модулів.

1.5 Висновок до першого розділу

У першому розділі проведено аналіз предметної області книжкових рекомендаційних систем і визначено основні проблеми персоналізованого вибору літератури. Встановлено, що велика кількість доступних видань, розрідженість користувацьких оцінок і проблема холодного старту ускладнюють формування точних рекомендацій на основі одного джерела даних.

Проаналізовано функціональні можливості наявних книжкових сервісів Goodreads, The StoryGraph, LibraryThing і BookBub. Виявлено, що вони підтримують ведення читацьких профілів, оцінювання книг і персоналізовані добірки, однак здебільшого не поєднують прогнозування оцінок, пошук користувача зі схожими вподобаннями, антирекомендації та пояснення отриманого результату.

Важливою вимогою визначено пояснюваність сформованого результату. Користувачеві недостатньо показати лише перелік рекомендованих творів, оскільки без додаткової інформації складно зрозуміти причини їх появи.

Розглянуто колаборативну й змістову фільтрацію, матричну факторизацію SVD, метод TF-IDF і гібридний підхід. Колаборативні методи дають змогу враховувати закономірності в оцінках користувачів, а змістова модель використовує метадані книг і частково компенсує нестачу історичних взаємодій.

За результатами аналізу сформульовано завдання розроблення інформаційної системи «Book Twin». Система повинна забезпечувати підготовку книжкових даних, створення локальних користувацьких профілів, прогнозування оцінок, формування рекомендацій та антирекомендацій, пошук книжкового двійника, пояснення результатів і експериментальне оцінювання якості моделей. Сформульовані вимоги є основою для подальшого проектування архітектури та програмної реалізації системи.

РОЗДІЛ 2. ПРОЄКТУВАННЯ ІНТЕЛЕКТУАЛЬНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ АНАЛІЗУ ЧИТАЦЬКИХ УПОДОБАНЬ

2.1 Формування вимог до інформаційної системи

Проектування інформаційної системи «Book Twin» розпочато з уточнення її меж, основних користувацьких сценаріїв і вимог до результату. Система призначена не для заміни читацького рішення, а для підтримки вибору наступної книги на основі історії оцінювання, характеристик творів і закономірностей, виявлених у поведінці інших користувачів. Тому в центрі проектування перебуває не окрема модель машинного навчання, а повний процес: від створення профілю і введення оцінок до ранжування кандидатів, пояснення результату та збереження даних між запусками.

Функціональні вимоги сформовано з урахуванням того, що користувач повинен мати змогу почати роботу без попередньої присутності у навчальному наборі даних. Після створення локального профілю він знаходить книги у каталозі, виставляє оцінки за шкалою від 1 до 5, редагує або видаляє їх. На основі накопичених оцінок система формує рекомендації й антирекомендації, визначає книжкового двійника та показує короткі причини включення книги до певної добірки. Такий сценарій усуває залежність інтерфейсу від ідентифікаторів користувачів початкового набору та дає змогу перевірити моделі на новому профілі.

До нефункціональних вимог належать відтворюваність розрахунків, прийнятний час відповіді, цілісність локальних даних, зрозумілість інтерфейсу й можливість повторного використання навчених моделей. Обчислювально витратні операції, зокрема побудова факторів SVD і TF-IDF-матриці, мають виконуватися заздалегідь, а під час взаємодії користувача завантажуватися як готові артефакти. Це зменшує час очікування і відокремлює етап навчання від етапу використання системи.

Архітектурний опис має відображати не лише перелік компонентів, а й їхню відповідальність та взаємодію. Такий підхід відповідає загальним принципам опису архітектури програмних систем, де різні подання використовуються для пояснення структури, поведінки й потоків даних [16]. Для фіксації користувацьких сценаріїв і взаємодії компонентів застосовано нотацію UML 2.5.1 [17].

Сформовані функціональні та нефункціональні вимоги до інформаційної системи наведено в таблиці 2.1.

Таблиця 2.1 – Основні вимоги до інформаційної системи

Група вимог	Вимога	Обґрунтування
Функціональні	Створення, вибір і видалення локального профілю	Забезпечує роботу з користувачами, відсутніми у початковому наборі даних
Функціональні	Пошук книг і керування оцінками	Формує індивідуальну історію взаємодій, необхідну для персоналізації
Функціональні	Рекомендації, антирекомендації та книжковий двійник	Надає не лише позитивний, а й попереджувальний персоналізований результат
Функціональні	Пояснення сформованого результату	Підвищує прозорість і дає змогу перевірити внесок окремих сигналів
Нефункціональні	Збереження профілів і оцінок між запусками	Унеможлиблює втрату введених користувачем даних
Нефункціональні	Використання попередньо навчених моделей	Зменшує час відповіді та навантаження під час роботи вебзастосунку
Нефункціональні	Стабільність і відтворюваність	Забезпечує однакову логіку розрахунку для однакових вхідних даних

Основним учасником взаємодії з інформаційною системою є користувач. Після запуску застосунку він створює новий локальний профіль або обирає раніше збережений, знаходить книги за назвою чи автором і формує власну історію оцінювання. На основі введених оцінок система будує персональне представлення читацьких уподобань, яке надалі використовується для прогнозування ставлення до непрочитаних книг.

Центральним сценарієм є отримання персоналізованої добірки. Перед формуванням результату система перевіряє достатність даних профілю, вилучає вже оцінені книги та об'єднує прогнози рекомендаційних моделей. Окремо користувач може переглянути антирекомендації, знайденого книжкового двійника й коротке пояснення причин появи кожної книги у списку. Після додавання, зміни або видалення оцінки профіль уточнюється, тому наступний запит може сформулювати інше ранжування. Основні сценарії взаємодії узагальнено на рисунку 2.1. Діаграма не деталізує внутрішні алгоритми, а фіксує дії, доступні користувачеві, та межу відповідальності системи.



Рисунок 2.1 – Діаграма варіантів використання інформаційної системи «Book Twin»

Усі сценарії пов'язані з профілем користувача. Рекомендації не формуються як статичний список: кожна зміна оцінки змінює профіль, після чого

результати можуть бути перераховані. Пояснення, антирекомендації та інформація про книжкового двійника розглядаються як частини одного результату, а не як окремі незалежні функції.

2.2 Проєктування архітектури системи

Для «Book Twin» обрано модульну архітектуру, у якій інтерфейс, прикладна логіка, моделі та сховище даних мають окремі зони відповідальності. Це рішення спрощує тестування, повторне використання моделей і подальшу заміну окремих алгоритмів. Наприклад, зміна способу обчислення книжкового двійника не потребує перебудови таблиць бази даних або логіки відображення карток книг.

Вебінтерфейс реалізується засобами Streamlit. Фреймворк орієнтований на інтерактивні застосунки для даних і машинного навчання та працює за моделлю повторного виконання сценарію після взаємодії з елементами інтерфейсу [18]. Через це ресурси, завантаження яких є дорогим, не повинні створюватися під час кожного повторного запуску. У проєкті дані, фактори SVD, TF-IDF-матриця та відображення ідентифікаторів завантажуються у функції, позначеній кешуванням ресурсів. Офіційна документація Streamlit рекомендує `'st.cache_resource'` для моделей машинного навчання та інших спільних об'єктів [19].

Рівень прикладної логіки координує роботу профілів, моделей і формування результату. Він перетворює локальні оцінки у формат, придатний для SVD та змістової моделі, викликає пошук двійника, нормалізує прогнози та формує підсумкову таблицю кандидатів. Відображення інтерфейсу не містить логіки навчання: воно лише передає вибір користувача та виводить отримані структури даних.

Рівень даних складається з очищених CSV-файлів, збережених масивів і розріджених матриць моделей, а також локальної бази SQLite. SQLite є вбудованою, самодостатньою та серверною-незалежною транзакційною СУБД

[20], тому відповідає масштабу локального демонстраційного застосунку. Доступ до неї здійснюється через стандартний модуль `sqlite3` Python [21].

Компонентну структуру системи наведено на рисунку 2.2.

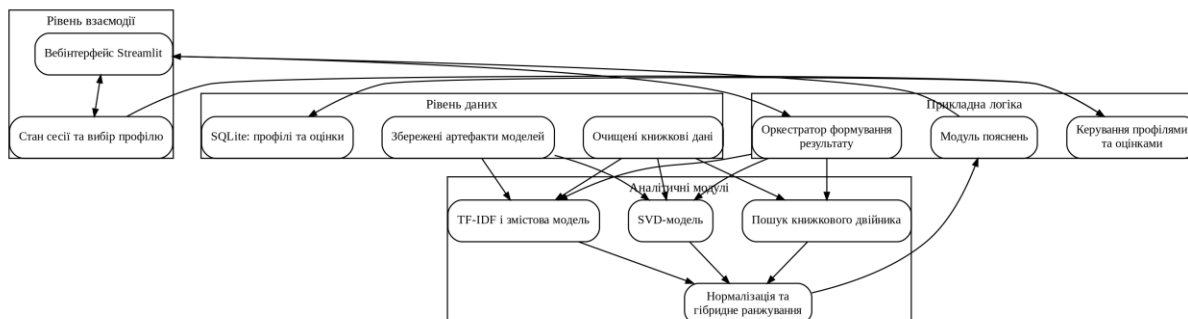


Рисунок 2.2 – Компонентна архітектура інформаційної системи «Book Twin»

Вебінтерфейс забезпечує взаємодію з користувачем, модуль керування профілями та база даних відповідають за збереження оцінок, а рекомендаційні моделі виконують прогнозування й формування підсумкового рейтингу.

Розподіл відповідальності між основними модулями інформаційної системи наведено в таблиці 2.2.

Таблиця 2.2 – Відповідальність основних модулів системи

Модуль	Основні функції	Вхідні та вихідні дані
SVD-модель	Прогноз персональних оцінок за латентними факторами	Оцінки профілю; прогноз для спільного каталогу
Змістова модель	Побудова профілю за метаданими й оцінювання змістової подібності	TF-IDF-вектори книг; змістові оцінки
Пошук двійника	Порівняння профілю з користувачами набору даних	Центровані оцінки; ідентифікатор і показники схожості
Гібридне ранжування	Нормалізація, зважене об'єднання, фільтрація оцінених книг	Прогнози моделей; впорядкований список кандидатів
Модуль пояснень	Формування коротких текстових причин	Окремі складові прогнозу; пояснення для картки книги

Програмна реалізація системи має переважно функціонально-модульну структуру та не передбачає розвиненої ієрархії доменних класів. Опрацювання

даних виконується за допомогою таблиць pandas, масивів NumPy та розріджених матриць SciPy.

2.3 Характеристика та підготовка набору даних

Основою експериментальної частини є набір Goodbooks-10k, що містить дані про десять тисяч популярних книг і приблизно шість мільйонів користувацьких оцінок, а також метадані, теги та зв'язки «книга–тег» [15]. Його вибір пояснюється поєднанням явних оцінок і характеристик книг, необхідних для побудови одночасно колаборативної та змістової складових.

Початкові файли мають різне призначення і використовують кілька систем ідентифікаторів. Після очищення формуються `books_clean.csv` і `ratings_clean.csv`, які використовуються в усіх наступних ноутбуках і в застосунку.

Структуру джерел даних наведено в таблиці 2.3.

Таблиця 2.3 – Використані файли набору даних

Файл	Основні поля	Призначення
books.csv / books_clean.csv	book_id, title, authors, original_publication_year, image_url та ін.	Каталог і метадані книг
ratings.csv / ratings_clean.csv	user_id, book_id, rating	Матриця явних оцінок користувачів
book_tags.csv	goodreads_book_id, tag_id, count	Частота використання тегів для книг
tags.csv	tag_id, tag_name	Текстові назви тегів
Артефакти моделей	фактори SVD, TF-IDF-матриця, масиви ідентифікаторів	Швидке використання навчених моделей у застосунку

Попередня обробка даних передбачала перевірку ідентифікаторів, вилучення дублікатів, опрацювання пропущених значень, нормалізацію текстових полів і формування розрідженої матриці користувацьких оцінок.

Окрему частину взаємодій було відкладено для подальшого оцінювання якості рекомендацій, щоб тестові дані не використовувалися під час навчання моделей.

Матриця «користувач – книга» є розрідженою, оскільки навіть активний читач оцінює лише невелику частину каталогу. Тому її доцільно зберігати у форматі CSR, у якому фіксуються лише ненульові елементи. Такий формат сумісний із засобами нормалізації та TruncatedSVD і дає змогу зменшити витрати пам'яті.

Для змістової моделі формується текстове поле `content_text`, що об'єднує назву, автора та найбільш інформативні теги книги. Використання кількох джерел метаданих зменшує ризик втрати тематично близьких творів через відмінності у формулюваннях жанрів і тегів.

Після очищення даних відображення ідентифікаторів користувачів і книг зберігаються окремо від коду інтерфейсу. Це забезпечує однаковий каталог для SVD-моделі, змістової моделі, пошуку книжкового двійника та вебзастосунку.

Фрагмент формування центрованої розрідженої матриці наведено в лістингу 2.1.

Лістинг 2.1 – Формування центрованої розрідженої матриці оцінок

```

user_ids = np.sort(train["user_id"].unique())
book_ids = np.sort(train["book_id"].unique())

user_to_index = {
    int(user_id): index
    for index, user_id in enumerate(user_ids)
}
book_to_index = {
    int(book_id): index
    for index, book_id in enumerate(book_ids)
}

user_means = train.groupby("user_id")["rating"].mean()
train["user_index"] = train["user_id"].map(user_to_index)
train["book_index"] = train["book_id"].map(book_to_index)
train["rating_centered"] = (
    train["rating"].astype("float32")
    - train["user_id"].map(user_means).astype("float32")
)

train_matrix = csr_matrix(

```

```
(
    train["rating_centered"].to_numpy(dtype=np.float32),
    (
        train["user_index"].to_numpy(dtype=np.int32),
        train["book_index"].to_numpy(dtype=np.int32),
    ),
),
shape=(len(user_ids), len(book_ids)),
dtype=np.float32,
)
```

У наведеному фрагменті ідентифікатори користувачів і книг перетворюються на послідовні індекси, після чого оцінки центруються відносно середнього значення відповідного користувача. Отримані значення записуються до розрідженої CSR-матриці, яка використовується під час подальшого навчання SVD-моделі.

2.4 Проектування процесу машинного навчання

Процес машинного навчання у системі поділено на два режими: офлайн-підготовку моделей і онлайн-формування результату для конкретного профілю. Офлайн-режим виконується в ноутбуках і створює артефакти, які не змінюються під час звичайної роботи застосунку. Онлайн-режим використовує ці артефакти разом із локальними оцінками користувача.

Колаборативна складова навчається на центрованій матриці оцінок. Для кожного користувача від його оцінок віднімається середнє значення, завдяки чому модель описує не загальну схильність ставити високі або низькі бали, а відхилення від індивідуальної норми. Для зниження розмірності використано `TruncatedSVD`, який працює безпосередньо з розрідженими матрицями [22]. Кількість компонент дорівнює 40: цього достатньо для компактного латентного представлення і водночас не створює надмірної складності для демонстраційного набору.

Змістова складова будується за допомогою `TfidfVectorizer`, який перетворює текстові описи книг у розріджену матрицю TF-IDF-ознак [23].

Максимальна кількість ознак встановлена на рівні 5000. Додатково використовуються уніграми й біграми, мінімальна частота документа $\text{min_df}=2$, обмеження $\text{max_df}=0,95$ і сублінійна частота терміна. Така конфігурація вилучає одиничний шум, зменшує вплив надто загальних слів і зберігає сталі словосполучення.

Загальну схему навчання та використання моделей наведено на рисунку 2.3.

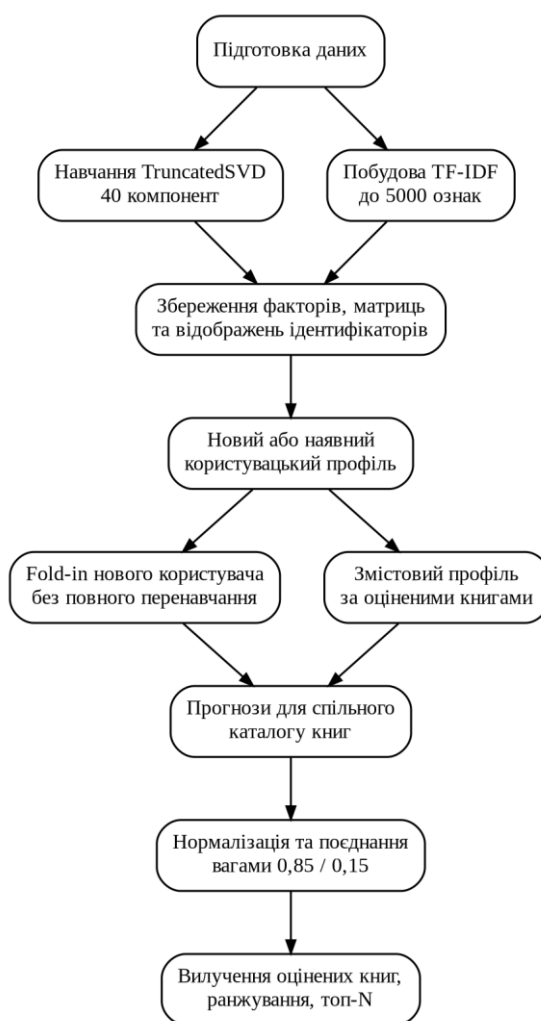


Рисунок 2.3 – Схема навчання та використання рекомендаційних моделей

Для побудови SVD-моделі використано 40 латентних компонент. Таке значення забезпечує компроміс між здатністю моделі відображати приховані закономірності в оцінках і складністю обчислень. Змістове представлення книг

сформовано методом TF-IDF з обмеженням до 5000 ознак і використанням уніграм та біграм. Це дає змогу враховувати як окремі терміни, так і стійкі словосполучення без надмірного збільшення розмірності.

Підсумковий прогноз гібридної моделі формується з вагами 0,85 для SVD-складової та 0,15 для змістової. Значення визначено за результатами експериментального оцінювання на незалежній вибірці користувачів. За наявності оцінки книги книжковим двійником до підсумкового результату додається сигнал із вагою 0,15, що підсилює пояснюваність рекомендації, але не дозволяє оцінці одного користувача домінувати у фінальному рейтингу.

Для нового локального профілю немає готового рядка у матриці користувацьких факторів. Повне перенавчання SVD після кожної зміни оцінки потребувало б зайвих обчислень і порушувало б відокремлення офлайн- та онлайн-етапів. Тому застосовано метод fold-in: латентний вектор нового користувача визначається за вже навченими факторами книг як розв'язок регуляризованої задачі найменших квадратів [22]. Перед цим середня оцінка профілю згладжується до глобального середнього, що зменшує нестійкість для профілів із малою кількістю оцінок.

Згладжене середнє профілю обчислюється за формулою:

$$\bar{r}_p = (\sum r_i + \lambda \bar{r}) / (n + \lambda), \quad (2.1)$$

де \bar{r}_p – згладжене середнє профілю; r_i – введені користувачем оцінки; \bar{r} – глобальне середнє; n – кількість оцінок профілю; λ – коефіцієнт апріорного згладжування. У програмній реалізації значення λ дорівнює 5.

Після центрування локальних оцінок латентний фактор нового користувача визначається з регуляризованої системи нормальних рівнянь. Параметр регуляризації встановлено на рівні 8,0, щоб обмежити надмірні значення факторів за недостатньої кількості взаємодій.

Під час обчислення враховуються лише ті книги, які користувач уже оцінив і для яких у навченій моделі наявні латентні представлення. Відповідні

вектори книг утворюють локальну матрицю факторів, а центровані оцінки формують вектор спостережуваних значень. На їх основі визначається положення нового профілю в тому самому латентному просторі, у якому представлені користувачі навчального набору. Завдяки цьому можна отримувати прогнози для спільного каталогу книг без повторного розкладання всієї матриці оцінок.

Латентний вектор нового користувача визначається шляхом розв'язання регуляризованої задачі найменших квадратів за формулою:

$$q_u = (V^T V + \lambda I)^{-1} V^T (r_u - \bar{r}_u), \quad (2.2)$$

де q_u – латентний вектор нового користувача; V – матриця факторів оцінених книг; I – одинична матриця; λ – коефіцієнт регуляризації; r_u – вектор введених оцінок; \bar{r}_u – згладжене середнє профілю. За неможливості прямого розв'язання через числову нестійкість використовується псевдообернена матриця, а прогнозовані оцінки обмежуються діапазоном від 1 до 5.

Ключовий фрагмент обчислення латентного представлення нового користувача наведено в лістингу 2.2.

Лістинг 2.2 – Fold-in нового користувача до латентного SVD-простору

```
prior_strength = 5.0
profile_mean = (
    float(available["rating"].sum())
    + prior_strength * resources["global_mean"]
) / (len(available) + prior_strength)

item_matrix = resources["svd_item_factors"][:, rated_indices].T
centered_values = available["rating"].to_numpy() - profile_mean

regularization = 8.0
system_matrix = (
    item_matrix.T @ item_matrix
    + regularization * np.eye(item_matrix.shape[1])
)
target_vector = item_matrix.T @ centered_values
```

```

try:
    user_factor = np.linalg.solve(system_matrix, target_vector)
except np.linalg.LinAlgError:
    user_factor = np.linalg.pinv(system_matrix) @ target_vector

predictions = profile_mean + user_factor @ item_factors
predictions = np.clip(predictions, 1.0, 5.0)

```

У наведеному кодї обчислюється згладжене середнє профілю, формується система рівнянь і визначається латентний фактор користувача. Регуляризація стабілізує розв'язок, а псевдообернена матриця є резервним варіантом для виродженої системи.

Змістовий профіль нового користувача будується як зважена сума TF-IDF-векторів оцінених книг. Вага кожного твору дорівнює відхиленню його оцінки від середнього профілю: позитивні відхилення підсилюють відповідні ознаки, а негативні зменшують їхній внесок. Після L2-нормалізації профілю його подібність до всіх книг обчислюється як добуток на TF-IDF-матрицю [23], [24].

Ваги гібридної моделі визначено не довільно, а за результатами перебору на окремій групі користувачів. Для ваги SVD перевірено значення від 0,50 до 0,90 з кроком 0,05, а вага змістової складової визначалася як доповнення до одиниці. Найкраще середнє значення NDCG@10 отримано для поєднання 0,85 SVD і 0,15 змістової складової. Уже оцінені книги вилучаються з кандидатів; верхня частина впорядкованого списку формує рекомендації, а нижня – антирекомендації.

2.5 Проектування алгоритму пошуку книжкового двійника

Процес машинного навчання у системі поділено на два режими: офлайн-підготовку моделей і онлайн-формування результату для

Книжковий двійник визначається як користувач набору даних, чий профіль оцінювання найбільш узгоджений із поточним локальним профілем. Порівняння лише середніх оцінок є недостатнім, оскільки читачі можуть однаково часто

ставити високі бали, але оцінювати різні книги. Тому алгоритм працює на перетині оцінених творів і поєднує кілька показників схожості.

Спочатку оцінки активного профілю центруються відносно його середнього значення. Якщо всі введені оцінки однакові, вони відхиляються від нейтральної точки 3, щоб уникнути нульового вектора. Далі створюється розріджений вектор у просторі книг і виконується L2-нормалізація. Після нормування матричний добуток із підготовленою матрицею користувачів відповідає косинусній подібності [24], [26].

Щоб не виконувати детальне порівняння з усіма користувачами, для подальшого аналізу відбираються 250 кандидатів із найбільшою початковою косинусною подібністю. Для кожного кандидата визначаються спільно оцінені книги та середня абсолютна різниця оцінок. Отримана різниця перетворюється на показник узгодженості від 0 до 1. За наявності щонайменше трьох спільних книг цей показник додатково усереднюється з нормованим коефіцієнтом кореляції Пірсона.

Надійність збігу залежить від кількості спільно оцінених книг. Тому показник узгодженості коригується вагою перекриття, яка зростає зі збільшенням кількості спільних оцінок і зменшує вплив випадково високої схожості, отриманої лише за одним або двома збігами.

Підсумковий показник формується шляхом поєднання невід'ємної косинусної подібності та узгодженості оцінок з однаковими вагами. Після цього результат коригується з урахуванням кількості спільно оцінених книг. Негативні значення косинусної подібності не враховуються, оскільки вони свідчать про протилежний напрям оцінювання. Кандидати сортуються за підсумковим показником, а за однакового значення перевага надається користувачеві з більшою кількістю спільних книг.

Додатковою умовою вибору книжкового двійника є достатня кількість спільно оцінених творів. Подібність, отримана лише за однією або двома книгами, не може вважатися надійною, навіть якщо оцінки повністю збігаються. Тому підсумковий показник коригується з урахуванням розміру перекриття між

профілями. Зі збільшенням кількості спільних книг вплив цього коригування послаблюється, оскільки результат уже підтверджується більшою кількістю спостережень.

Алгоритм також зберігає проміжні характеристики найкращого кандидата: початкову косинусну подібність, кількість спільно оцінених книг, середню різницю між оцінками та підсумковий показник схожості. Ці дані використовуються не лише для ранжування кандидатів, а й для формування зрозумілого пояснення користувачеві. У результаті система показує не просто ідентифікатор схожого читача, а й кількісні ознаки, на підставі яких його було визначено книжковим двійником.

Послідовність визначення книжкового двійника наведено на рисунку 2.4.

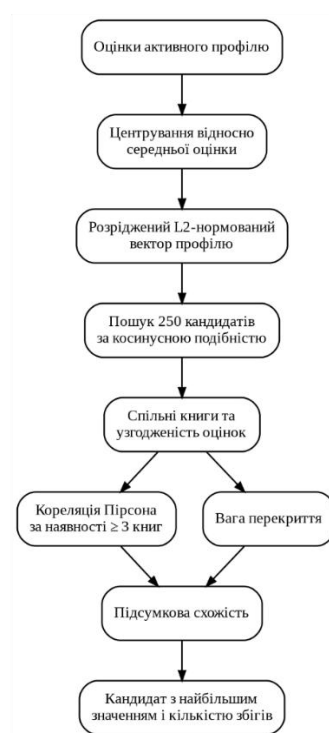


Рисунок 2.4 – Алгоритм визначення книжкового двійника

Схема відображає двоетапний пошук: швидке визначення кандидатів у розрідженому просторі та точніше повторне оцінювання на множині спільних

книг. Такий підхід скорочує обчислення й водночас не дозволяє одному випадковому збігу визначити результат.

2.6 Проектування бази даних і локальних профілів

Процес Локальні профілі та поставлені ними оцінки зберігаються у файлі SQLite book_twin.sqlite3. Вбудована база даних не потребує окремого сервера, переноситься разом із застосунком і підтримує транзакційне збереження змін [20]. Каталог книг і навчені артефакти моделей залишаються зовнішніми файлами, тому до локальної бази записуються лише дані, створені користувачем.

Структура сховища містить дві основні сутності. У таблиці profiles зберігаються внутрішній ідентифікатор, унікальна назва профілю та час його створення. Таблиця profile_ratings містить посилання на профіль, ідентифікатор книги, оцінку від 1 до 5 і час останнього оновлення. Поєднання ідентифікаторів профілю та книги використовується як складений ключ, тому одна книга не може мати кілька одночасних оцінок у межах одного профілю.

Логічну структуру локальної бази даних наведено на рисунку 2.5.

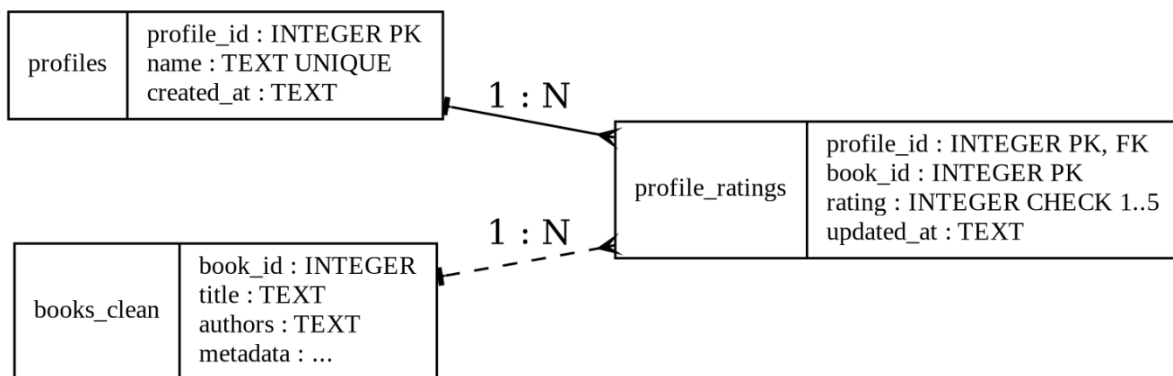


Рисунок 2.5 – Логічна схема бази даних локальних профілів

Сутність books_clean на схемі позначає очищений каталог книг, який не дублюється у SQLite. Зв'язок із ним є логічним: перед збереженням оцінки

прикладний модуль перевіряє, що вказаний ідентифікатор належить завантаженому каталогу. Такий підхід зменшує розмір локального файлу й унеможливорює появу оцінок для неіснуючих книг.

Під час додавання нової оцінки створюється відповідний запис, а за повторного оцінювання тієї самої книги значення і час оновлення замінюються. Видалення профілю супроводжується автоматичним очищенням залежних оцінок. Параметризовані запити стандартного модуля `sqlite3` відокремлюють значення від тексту SQL-команд, а транзакції забезпечують цілісність даних [21].

Відокремлення локальних профілів від навчального набору дає змогу багаторазово запускати застосунок, продовжувати роботу з уже створеним профілем і поступово уточнювати його читацькі вподобання. Після кожної зміни оцінок дані зчитуються з бази та передаються рекомендаційним модулям для оновлення результату.

2.7 Методика експериментального оцінювання

Оцінювання рекомендаційної системи має враховувати не лише точність окремого числового прогнозу, а й якість упорядкованого списку. Користувач бачить обмежену кількість верхніх позицій, тому розташування релевантної книги на початку добірки має більшу практичну цінність, ніж її поява наприкінці великого переліку. З цієї причини експеримент проводиться для десяти найвищих позицій [34], [35].

Для перевірки використовується вибірка з 80 незалежних користувачів. Частина їхніх оцінок вважається відомою та формує профіль, а решта використовується як тестовий набір релевантних книг. Окрема група користувачів застосовується для підбору ваг гібридної моделі, щоб параметри не визначалися за результатами кінцевого тестування.

Порівнюються три підходи: базова рекомендація популярних книг, SVD-модель і гібридна модель SVD + Content. Базова стратегія показує результат без

персонального профілю, SVD відображає внесок колаборативної складової, а гібридний підхід дає змогу оцінити ефект додавання змістового сигналу.

Послідовність експериментального оцінювання наведено на рисунку 2.6.

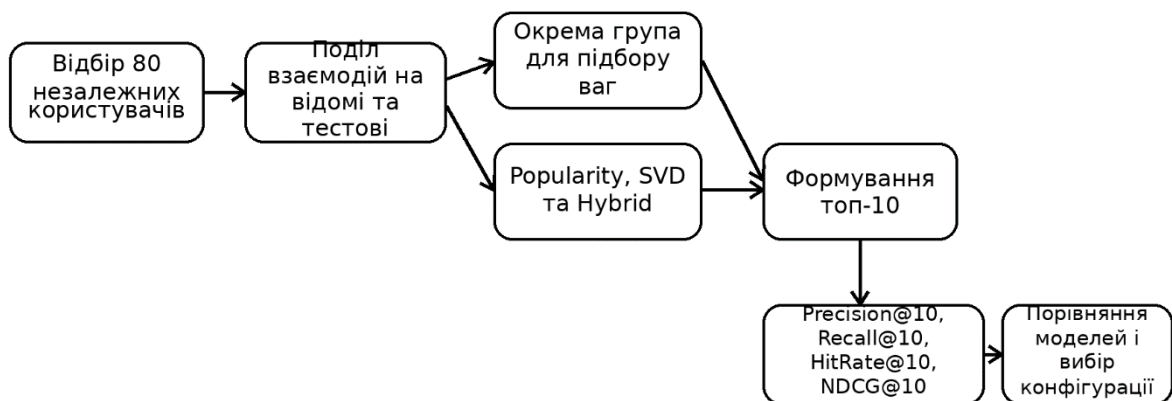


Рисунок 2.6 – Схема експериментального оцінювання рекомендаційних моделей

Для кожного користувача вже оцінені книги вилучаються з множини кандидатів, після чого формується топ-10. Precision@10 показує частку релевантних книг серед рекомендованих, Recall@10 – частку відомих релевантних книг, які модель змогла знайти, а HitRate@10 відображає частку профілів, для яких у списку є хоча б один збіг. NDCG@10 додатково враховує позицію релевантної книги та надає більшу вагу збігам на початку списку [25].

Під час підбору ваг основним критерієм є NDCG@10, а Recall@10 і HitRate@10 використовуються як додаткові показники стійкості.

Під час формування навчальної та тестової частин важливо не допустити витоку інформації. Оцінки, відкладені для перевірки, не використовуються під час побудови профілю користувача, навчання моделей або підбору ваг гібридної складової. Крім того, з переліку кандидатів вилучаються книги, які вже були доступні моделі як відомі взаємодії. Це забезпечує перевірку здатності системи знаходити нові релевантні твори, а не повторно відтворювати відомі оцінки.

Значення метрик спочатку обчислюються окремо для кожного користувача, після чого усереднюються за всією тестовою вибіркою. Такий підхід не дозволяє профілям із великою кількістю оцінок непропорційно впливати на загальний результат. Окремо фіксується кількість користувачів, для яких модель змогла сформувати повний список рекомендацій, оскільки стабільність роботи для різних профілів є важливою характеристикою практичної придатності системи.

Для забезпечення відтворюваності експерименту використовуються однаковий поділ даних, фіксовані параметри моделей і спільна процедура формування кандидатів. Базова, колаборативна та гібридна моделі перевіряються на тих самих користувачах і тестових взаємодіях. Завдяки цьому різниця між отриманими показниками пояснюється особливостями алгоритмів, а не відмінностями у вихідних умовах оцінювання.

2.8 Висновок до другого розділу

У другому розділі сформовано функціональні та нефункціональні вимоги до інформаційної системи «Book Twin», визначено основні користувацькі сценарії та спроектовано модульну архітектуру. Обґрунтовано структуру даних, порядок їх попередньої обробки й відокремлення офлайн-навчання моделей від онлайн-формування рекомендацій.

Описано застосування TruncatedSVD, TF-IDF і гібридного ранжування, а також спосіб формування латентного представлення нового користувача без повного перенавчання моделі. Окремо спроектовано алгоритм пошуку книжкового двійника, який враховує подібність профілів, узгодженість оцінок і кількість спільно прочитаних книг.

Визначено структуру локальної бази SQLite та правила збереження профілів і оцінок. Запропоновано методику експериментального оцінювання з використанням Precision@10, Recall@10, HitRate@10 і NDCG@10 та порівнянням гібридної моделі з SVD і рекомендацією популярних книг.

РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ДОСЛІДЖЕННЯ РОЗРОБЛЕНОЇ СИСТЕМИ

3.1 Обґрунтування вибору програмних засобів

Програмну реалізацію виконано мовою Python, оскільки вона має розвинену екосистему для опрацювання даних, машинного навчання та швидкого створення інтерактивних застосунків [27]. Табличні дані завантажуються й перетворюються засобами pandas [28], числові обчислення виконуються за допомогою NumPy [29], а розріджені матриці оцінок зберігаються у форматах SciPy [30]. Це дає змогу працювати з великою кількістю взаємодій без формування щільної матриці «користувач – книга».

Обраний підхід до реалізації системи узгоджується з сучасною практикою побудови прикладних інтелектуальних рішень, де важливими є попереднє опрацювання даних, використання методів машинного навчання, модульна структура програмного забезпечення та можливість подальшого розширення системи [37], [38], [39], [40], [41].

Для навчання й використання моделей застосовано бібліотеку scikit-learn, яка надає узгоджений інтерфейс до алгоритмів машинного навчання та перетворення ознак [31]. Колаборативна складова побудована на основі класу TruncatedSVD [22], а змістове представлення книг – за допомогою TfidfVectorizer [23]. L2-нормалізація використовується під час формування змістового профілю та пошуку користувачів зі схожими оцінками [26]. Збереження навчених об'єктів і параметрів забезпечують формати NumPy, NPZ, CSV та joblib; останній орієнтований на серіалізацію Python-об'єктів, що містять великі масиви [33].

Вебінтерфейс реалізовано засобами Streamlit. Його модель повторного виконання сценарію спрощує зв'язування елементів інтерфейсу з функціями Python [18]. Оскільки повторне завантаження факторів SVD, TF-IDF-матриці та книжкового каталогу потребує часу, спільні ресурси кешуються через

`st.cache_resource`, а результати обчислень для конкретного профілю – через `st.cache_data` [19]. Стан активного профілю та вибраних елементів підтримується за допомогою механізму `Session State`, який зберігає значення між повторними виконаннями сценарію в межах однієї користувацької сесії [32]. Після зміни оцінок кеш результатів очищується, тому наступна добірка формується за актуальним станом профілю.

Для постійного збереження локальних профілів використано SQLite. База розміщується у файлі `book_twin.sqlite3` і не потребує окремого серверного процесу [20]. Доступ до неї реалізовано через стандартний модуль `sqlite3` Python [21]. Такий підхід відповідає масштабу демонстраційної системи та дозволяє переносити застосунок разом із локальними даними.

Програмний проєкт побудовано за модульним принципом: очищені дані, артефакти моделей, експериментальні ноутбуки, вебінтерфейс і локальна база зберігаються окремо. Таке розмежування відповідальності спрощує інтеграцію компонентів, повторне проведення експериментів і заміну моделей без зміни функцій керування профілями [41].

3.2 Реалізація підготовки даних і моделей

Практична частина підготовки даних виконана у послідовності окремих Jupyter-ноутбуків. Спочатку формуються очищені файли `books_clean.csv` і `ratings_clean.csv`, у яких узгоджено ідентифікатори, опрацьовано пропущені значення та вилучено дублікати. Для табличних операцій застосовано `pandas` [28], а числові перетворення й компактні масиви виконано засобами `NumPy` [29]. Ці файли стають спільним джерелом даних для SVD-моделі, змістової складової, алгоритму книжкового двійника й вебінтерфейсу [28], [29], [31].

Для колаборативної моделі оцінки кожного користувача центруються відносно його середнього значення. Після перетворення ідентифікаторів на послідовні індекси створюється CSR-матриця, яка забезпечує швидке множення матриці на вектор і ефективне опрацювання рядків [30]. На цій матриці

навчається TruncatedSVD із 40 компонентами, алгоритмом randomized і п'ятьма ітераціями. Отримані фактори користувачів і книг зберігаються в масивах NumPy, а середні оцінки користувачів – в окремому CSV-файлі.

Змістова модель використовує назви, авторів і найінформативніші теги книг. Після очищення тексту формується поле content_text, яке перетворюється на TF-IDF-матрицю з обмеженням до 5000 ознак. Використання уніграм і біграм дає змогу врахувати як окремі терміни, так і стійкі словосполучення. Матриця зберігається у форматі NPZ, а налаштований векторизатор – у файлі joblib [33].

У гібридному ноутбучі результати SVD і змістової моделі приводяться до спільного масштабу.

Під час запуску застосунк визначає перетин каталогів, доступних SVD і TF-IDF, та формує єдиний масив книг-кандидатів. Це запобігає невідповідності індексів під час об'єднання прогнозів [28], [31].

Перед використанням артефактів перевіряються розміри матриць і відповідність масивів ідентифікаторів їхнім рядкам та стовпцям. Це запобігає зміщенню індексів і помилковому пов'язуванню прогнозу з іншою книгою.

Описаний процес підготовки даних і навчання моделей забезпечує узгодженість усіх компонентів рекомендаційної системи та можливість їх подальшого спільного використання. Збереження проміжних результатів у вигляді окремих артефактів дає змогу уникнути повторного навчання моделей під час кожного запуску застосунку та скоротити час формування рекомендацій. Крім того, такий підхід спрощує супровід системи й оновлення окремих модулів без внесення змін до всієї програмної архітектури.

Такий спосіб організації результатів є зручним для подальшого тестування, оскільки кожен збережений файл можна окремо перевірити та використати у відповідному модулі системи. Це також забезпечує стабільність роботи вебзастосунку, адже під час запуску він звертається вже до підготовлених даних і моделей.

Окреме збереження ідентифікаторів книг і користувачів дає змогу коректно пов'язувати результати обчислень із початковими записами набору

даних. Завдяки цьому система може швидко відновлювати потрібні зв'язки між прогнозами, метаданими книг і локальними профілями користувачів. Це є важливим для коректної роботи рекомендаційного модуля та уникнення помилок під час відображення результатів у вебінтерфейсі.

Результат виконання ноутбуків і збережені артефакти моделей показують на рисунку 3.1.

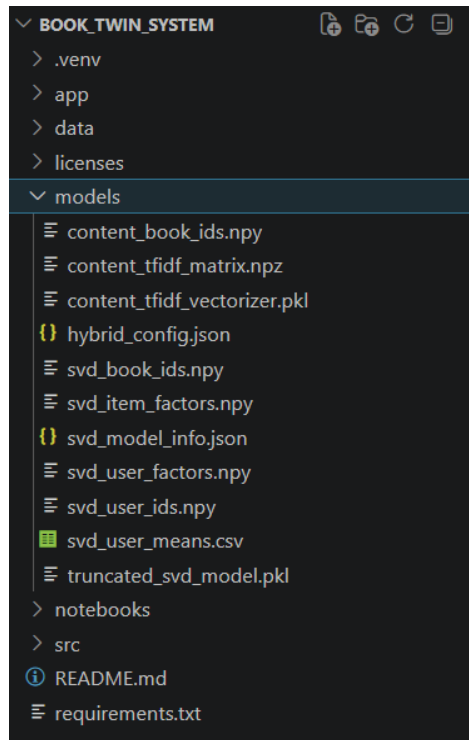


Рисунок 3.1 – Збережені артефакти навчених рекомендаційних моделей

Збереження результатів офлайн-етапу скорочує час запуску: під час роботи користувача система лише формує профіль, прогнозує оцінки, знаходить книжкового двійника та ранжує кандидатів. Разом із моделями завантажуються параметри конфігурації, тому ноутбуки й вебзастосунок використовують однакову логіку розрахунків.

Отже, на цьому етапі було сформовано набір підготовлених файлів і моделей, які забезпечують подальшу роботу рекомендаційного модуля. Після завантаження цих артефактів система може переходити безпосередньо до персоналізації результатів для конкретного користувача.

3.3 Реалізація алгоритмів персоналізації

Для нового локального профілю відсутній готовий рядок у матриці користувачьких факторів. Тому функція `fold_in_new_user_svd` формує його латентне представлення за оціненими книгами та вже навченими факторами об'єктів. Середня оцінка профілю попередньо згладжується до глобального середнього, а регуляризована система рівнянь розв'язується засобами NumPy. Якщо пряме розв'язання нестійке, використовується псевдообернена матриця.

Змістовий профіль будується функцією `build_content_profile`. Для кожної оціненої книги визначається відхилення оцінки від середнього значення профілю. Вектори позитивно оцінених книг підсилюють відповідні тематичні ознаки, а низькі оцінки зменшують їхній внесок. Після нормалізації профілю обчислюється його подібність до всіх книг спільного каталогу. Додатково визначаються найбільш характерні позитивні й негативні терміни, які потім використовуються у вкладці огляду.

Алгоритм книжкового двійника реалізовано функцією `find_dataset_twin_for_profile`. Локальні оцінки центруються та перетворюються на L2-нормований розріджений вектор. Матричний добуток із підготовленою матрицею користувачів дає початкову косинусну подібність. Для 250 найкращих кандидатів додатково аналізуються спільні книги, абсолютна різниця оцінок, кореляція Пірсона та кількість збігів.

Остаточний результат формується функцією `finish_recommendation_result`. Базова оцінка книги є зваженим поєднанням SVD і змістового прогнозу. Якщо книжковий двійник оцінював конкретний твір, його оцінка додається як окремий сигнал із вагою 0,15. Такий сигнал не домінує у фінальному рейтингу, але робить результат більш зрозумілим для користувача.

Перед ранжуванням із множини кандидатів вилучаються всі вже оцінені книги. Верхні позиції за підсумковою оцінкою формують рекомендації, а нижні – антирекомендації. При цьому антирекомендація трактується лише як низька

очікувана відповідність поточному профілю, а не як характеристика об'єктивної якості твору.

Для кожної картки формується коротке пояснення. У позитивній добірці воно може містити значення SVD-прогнозу, відповідність змістовому профілю та оцінку книжкового двійника. Для антирекомендацій пояснюються низький прогноз, слабка змістова відповідність або негативний сигнал від схожого читача. Пояснення генеруються за фактичними складовими підсумкового результату, тому не потребують зовнішньої мовної моделі. Такий підхід відповідає принципам пояснюваних рекомендацій, де користувачеві подають не лише результат, а й зрозумілу причину його формування [36].

Обчислення кешуються за впорядкованою множиною оцінок і оновлюються після будь-якої зміни профілю. За відсутності оцінок персональне ранжування не запускається, а книги, яких немає у спільному каталозі моделей, не беруть участі в прогнозуванні. За малої кількості спільних творів результат пошуку двійника позначається як обмежено надійний.

3.4 Реалізація локальних профілів і збереження даних

Під час першого запуску застосунок автоматично створює таблиці `profiles` і `profile_ratings`, якщо вони ще відсутні. Для кожного з'єднання вмикається підтримка зовнішніх ключів. Назва профілю має бути непорожньою та унікальною без урахування регістру, а оцінка перевіряється обмеженням від 1 до 5.

Користувач може створювати кілька незалежних профілів, перемикатися між ними та видаляти непотрібні. Каскадне видалення забезпечує автоматичне очищення всіх пов'язаних оцінок. Кількість оцінених книг обчислюється під час отримання списку профілів і відображається безпосередньо у полі вибору активного профілю.

Пошук книг здійснюється за об'єднаним полем із назви та автора. Після вибору твору користувач задає оцінку за шкалою від 1 до 5. Операція UPSERT

додає новий запис або оновлює наявний для тієї самої пари профілю та книги. Окремо передбачено видалення вибраної оцінки.

Після збереження взаємодії застосунок очищує кеш персональних результатів і повторно виконує сценарій Streamlit. Завдяки цьому таблиця оцінок, кількість взаємодій і сформовані рекомендації відображають актуальний стан бази. Після перезапуску програми профілі відновлюються з файла SQLite без повторного введення даних.

Операції з базою виконуються короткими транзакціями. Помилки дублювання назви профілю, недопустимі оцінки та відсутні книги перетворюються на зрозумілі повідомлення інтерфейсу.

Роботу з профілями та оцінками показано на рисунку 3.2.

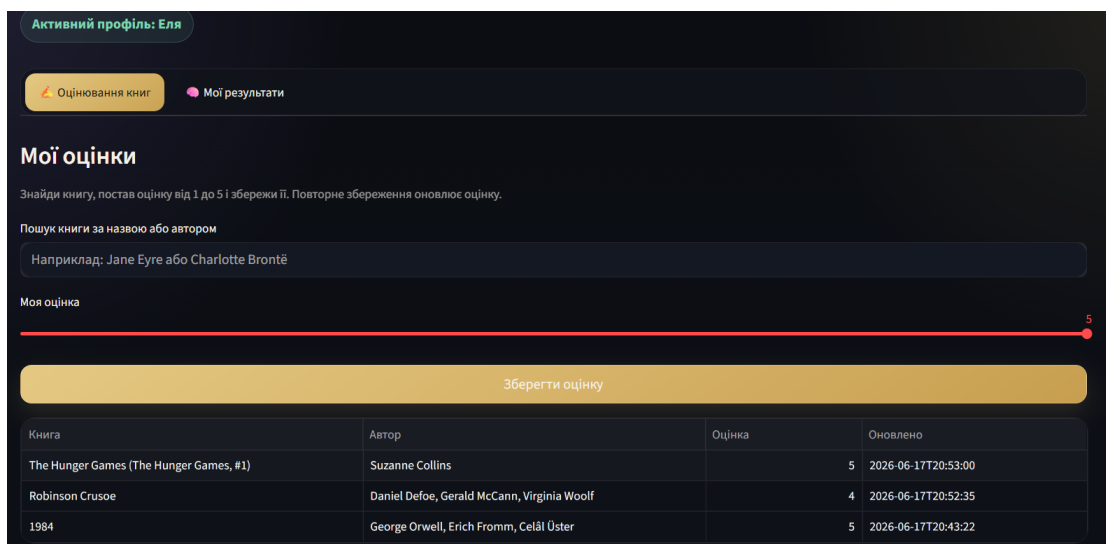


Рисунок 3.2 – Створення локального профілю та керування оцінками книг

Такий інтерфейс дає змогу перевірити весь цикл зміни даних без прямої роботи з SQL. Користувач бачить назву книги, автора, поставлену оцінку та час оновлення. Для отримання стійкішого профілю система рекомендує додати щонайменше три–п’ять різних оцінок.

Підхід до організації профілів користувачів забезпечує накопичення інформації про читацькі вподобання та створює основу для подальшої персоналізації рекомендацій. На відміну від одноразового пошуку книг за

ключовими словами, система враховує історію оцінок користувача та може адаптувати результати відповідно до його інтересів. Накопичення більшої кількості оцінок підвищує достовірність прогнозування та дає змогу точніше визначати схожість між читачами.

3.5 Розроблення вебінтерфейсу інформаційної системи

Інтерфейс реалізовано як односторінковий вебзастосунок із бічною панеллю та центральною областю результатів. Користувач обирає профіль, кількість книг у добірці та переходить між оцінюванням, оглядом, рекомендаціями, антирекомендаціями, книжковим двійником і показниками якості.

Початковий стан вебзастосунку після завантаження ресурсів наведено на рисунку 3.3.

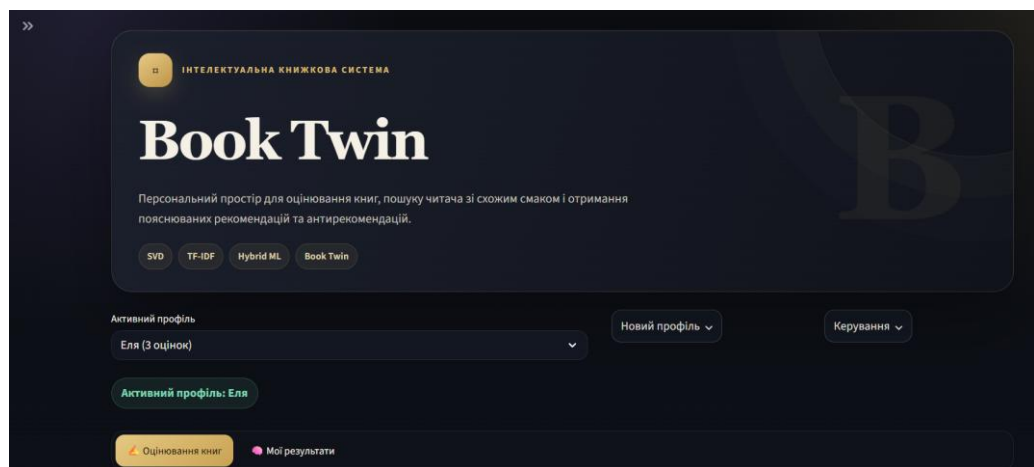


Рисунок 3.3 – Головна сторінка вебзастосунку «Book Twin»

Після вибору профілю користувач переходить між вкладками оцінювання та персональних результатів. У вкладці результатів спочатку відображаються п'ять узагальнених показників: назва профілю, кількість оцінених книг, середня оцінка, ідентифікатор книжкового двійника та підсумкова схожість.

Вкладка «Огляд» містить найбільш характерні терміни читацького профілю, кількість непрочитаних кандидатів, розподіл поставлених оцінок і короткий опис внеску трьох складових: SVD, Content-based і Book Twin. Для нового користувача додатково виводиться повідомлення про використання fold-in без повного перенавчання моделі.

Загальний профіль і внесок моделей продемонстровано на рисунку 3.4.

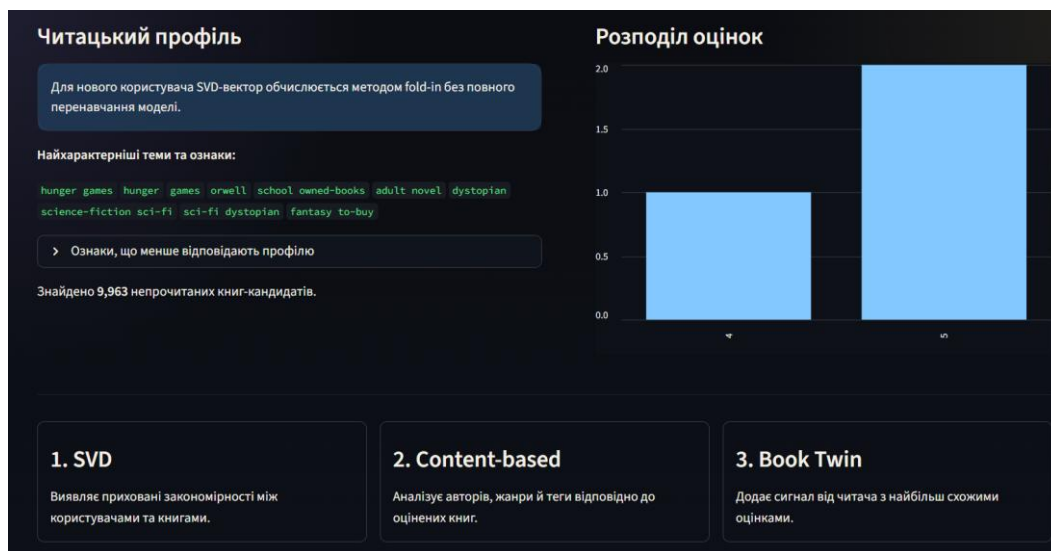


Рисунок 3.4 – Огляд читацького профілю та складових гібридної моделі

Вкладка «Рекомендації» виводить картки найбільш релевантних непрочитаних книг. Для кожної картки показуються обкладинка, назва, автор, прогнозована підсумкова оцінка, окремі сигнали моделей і текстове пояснення. Додатковий розгортний блок дозволяє переглянути добірку у табличному вигляді.

Перед відображенням результатів із переліку кандидатів вилучаються вже оцінені користувачем книги та перевіряється наявність основних метаданих. Навіть за відсутності обкладинки картка містить назву, автора, прогнозовану оцінку й пояснення, тому користувач отримує повну інформацію про рекомендацію.

Рекомендаційний список формується шляхом сортування непрочитаних книг за підсумковою оцінкою гібридної моделі. Найвищі позиції займають

твори, для яких колаборативна та змістова складові прогнозують найбільшу відповідність поточному читацькому профілю.

Приклад сформованих персональних рекомендацій наведено на рисунку 3.5.

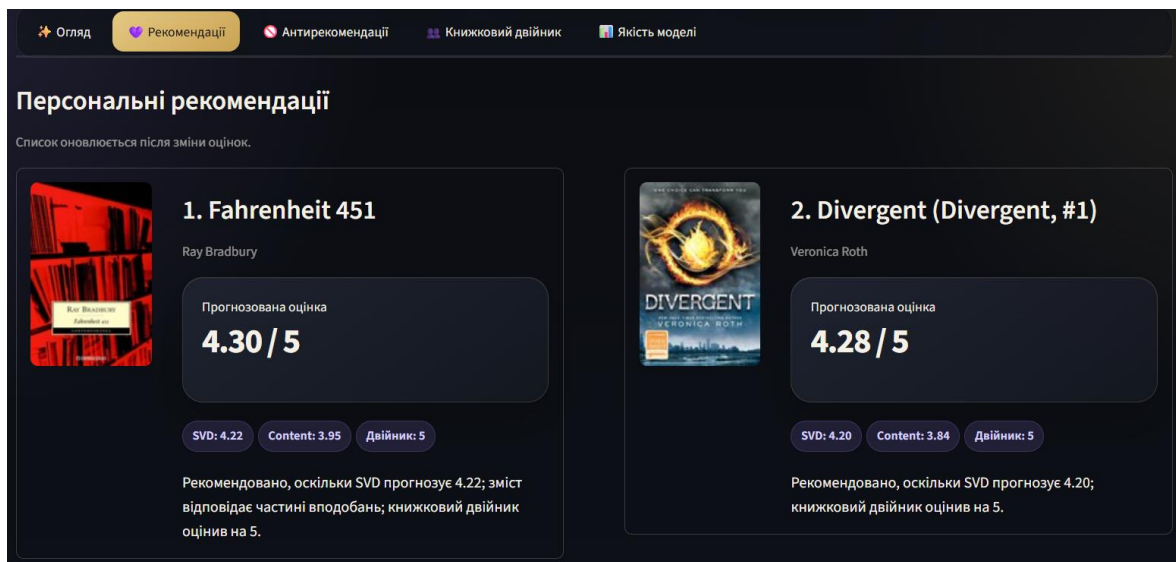


Рисунок 3.5 – Персональні рекомендації, сформовані системою

Під час формування антирекомендацій застосовується та сама послідовність підготовки кандидатів, що й для позитивної добірки. Із загального каталогу вилючаються вже оцінені користувачем книги, після чого для решти творів обчислюється підсумковий гібридний показник. До нижньої частини ранжованого списку потрапляють книги, для яких колаборативна та змістова складові одночасно вказують на низьку очікувану відповідність профілю. Завдяки цьому антирекомендації формуються персонально, а не на основі загальної популярності чи середнього рейтингу книги.

Для кожної позиції система відображає назву, автора, прогнозовану оцінку та коротке пояснення отриманого результату. Пояснення дає змогу відрізнити низьку змістову подібність від негативного прогнозу, сформованого за закономірностями користувацьких оцінок. Якщо книга має слабку відповідність одразу за кількома сигналами, вона розміщується вище у списку антирекомендацій. Водночас користувач може самостійно оцінити такий твір,

після чого нова взаємодія буде врахована під час наступного перерахунку профілю.

Відображення антирекомендацій показано на рисунку 3.6.

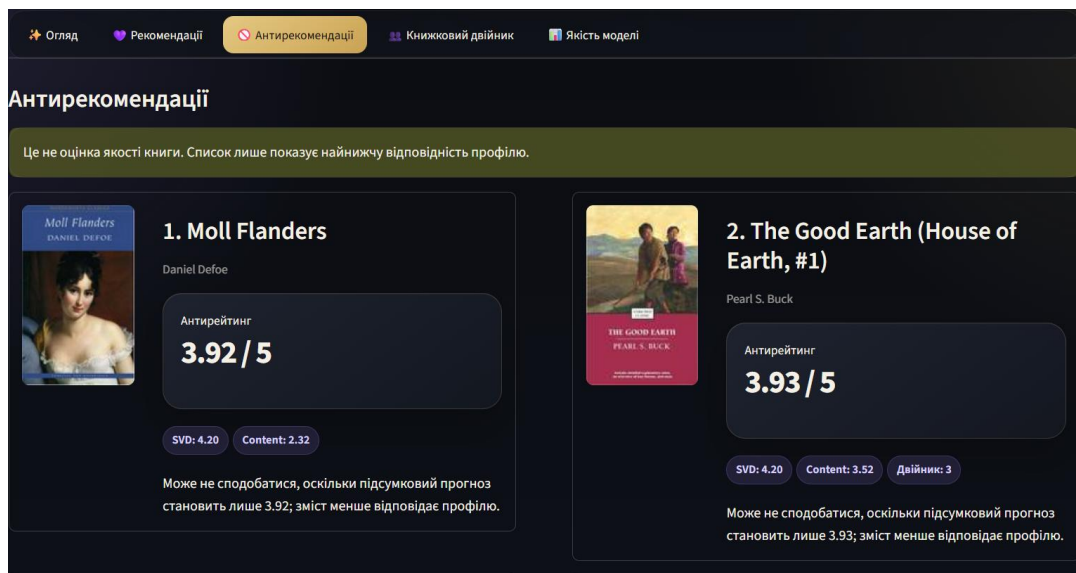


Рисунок 3.6 – Антирекомендації для вибраного читацького профілю

Окрема вкладка присвячена книжковому двійнику. У ній відображаються підсумкова та косинусна схожість, узгодженість оцінок і кількість спільних книг.

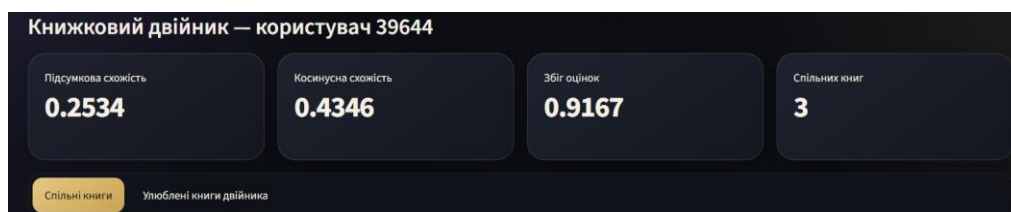


Рисунок 3.7 – Результат пошуку книжкового двійника

Поділ результатів на вкладки не перевантажує головну сторінку, а Session State зберігає активний профіль і вибраний режим між повторними виконаннями сценарію Streamlit [32].

Це дозволяє використовувати ресурси більш розумно.

3.6 Експериментальне оцінювання рекомендаційних моделей

Експериментальне оцінювання проведено на 80 незалежних користувачах, які не використовувалися для глобального підбору ваг. Для кожного профілю частина взаємодій залишалася відомою, а релевантні книги з відкладеної частини використовувалися для перевірки. Уже оцінені твори вилучалися з кандидатів перед формуванням топ-10. Така схема відповідає поширеному підходу до офлайн-порівняння рекомендаційних алгоритмів на однакових користувачах і тестових взаємодіях [34–35].

Порівнювалися базова стратегія популярності, SVD та гібридна модель SVD + Content.

Гібридна модель отримала найкращі значення за всіма чотирма метриками. Порівняно з SVD Precision@10 збільшилася приблизно на 42,3 %, Recall@10 – на 49,5 %, HitRate@10 – на 37,8 %, а NDCG@10 – на 52,7 %. Абсолютне значення HitRate@10 дорівнює 0,6375, тобто принаймні одна релевантна книга була знайдена для 63,75 % тестових профілів.

Низькі показники базової стратегії підтверджують, що загальна популярність не відображає індивідуальні вподобання. SVD суттєво покращує результат завдяки латентним закономірностям, однак додавання змістового сигналу забезпечує кращу повноту та розташування релевантних книг у верхній частині списку.

Перевага NDCG@10 є особливо важливою для практичного використання, оскільки ця метрика враховує позицію релевантної книги [25]. Поєднання декількох показників необхідне через те, що одна метрика не відображає одночасно точність, повноту та корисність ранжованого списку [34], [35]. Отже, гібридна модель не лише знаходить більше корисних кандидатів, а й частіше розміщує їх ближче до початку добірки [34], [35].

Збережені метрики завантажуються разом з іншими ресурсами та використовуються у вкладці «Якість моделі». Збіг цих значень із результатами

ноутбука підтверджує коректність інтеграції експериментальних даних у вебзастосунок.

3.7 Функціональне тестування та аналіз результатів

Функціональне тестування охопило створення й видалення профілів, перевірку унікальності їхніх назв, пошук книг, додавання, редагування та видалення оцінок, формування рекомендацій і пошук книжкового двійника. Також перевірено відновлення даних після перезапуску та відсутність уже оцінених книг у персональних добірках. Усі основні сценарії виконано успішно.

За порожнього профілю застосунок пропонує спочатку додати оцінки. Після зміни взаємодії кеш очищується, профіль і список рекомендацій перераховуються, а локальні дані зберігаються незалежно для кожного користувача.

До обмежень належать локальна база даних, відсутність мережевої автентифікації та залежність від каталогу Goodbooks-10k. Подальший розвиток може передбачати книжковий API, серверне збереження профілів, автоматичне оновлення моделей і користувацьке тестування.

Модульна структура також дає змогу надалі винести рекомендаційні обчислення у серверний сервіс і замінити SQLite на мережеву СУБД без зміни основних користувацьких сценаріїв [40].

3.8 Висновок до третього розділу

У третьому розділі виконано програмну реалізацію та експериментальне дослідження інтелектуальної інформаційної системи «Book Twin». Для розроблення використано мову Python і бібліотеки pandas, NumPy, SciPy та scikit-learn, а користувацький вебінтерфейс створено засобами Streamlit. Постійне збереження локальних профілів та оцінок реалізовано у базі даних SQLite. Обраний набір програмних засобів забезпечив опрацювання розріджених даних,

навчання моделей машинного навчання, збереження підготовлених артефактів і їх подальше використання без повторного виконання ресурсоемних операцій.

Практичну підготовку моделей організовано у вигляді послідовності окремих Jupyter Notebook. У них виконано очищення та узгодження книжкових даних, формування розрідженої матриці користувачьких оцінок, навчання SVD-моделі, побудову TF-IDF-представлення книг, підбір ваг гібридної моделі та оцінювання якості рекомендацій.

Для роботи з користувачами, відсутніми у початковому наборі даних, реалізовано формування нового латентного представлення методом fold-in. На основі введених оцінок створюється персональний SVD-профіль без повного повторного навчання моделі.

У вебзастосунку реалізовано створення, вибір і видалення локальних профілів, пошук книг, додавання, редагування та видалення оцінок. Для кожного профілю система формує персональні рекомендації, антирекомендації, інформацію про книжкового двійника та короткі пояснення результатів. Уже оцінені книги вилучаються з множини кандидатів, тому користувач отримує лише нові твори.

Експериментальне оцінювання проведено на 80 незалежних користувачах із використанням метрик Precision@10, Recall@10, HitRate@10 та NDCG@10. Гібридна модель отримала значення 0,1050, 0,1338, 0,6375 та 0,1670 відповідно.

Функціональне тестування підтвердило коректність створення профілів, збереження та оновлення оцінок, пошуку книг, формування рекомендацій і визначення книжкового двійника.

Отже, розроблена інформаційна система реалізує повний цикл персоналізації: від введення користувачьких оцінок і побудови читацького профілю до прогнозування ставлення до непрочитаних книг, формування позитивних і попереджувальних добірок та пояснення отриманих результатів. Програмна реалізація відповідає вимогам, сформульованим у попередніх розділах, а результати експериментального оцінювання підтверджують доцільність використання гібридної рекомендаційної моделі.

РОЗДІЛ 4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

Розроблення та використання інформаційної системи «Book Twin» пов'язане з тривалою взаємодією людини з комп'ютерною технікою, програмним інтерфейсом і робочим середовищем. Тому питання безпеки доцільно розглядати одночасно з позицій безпеки життєдіяльності та охорони праці. Перша зосереджується на закономірностях виникнення небезпек у системі «людина – машина – середовище існування», друга – на організаційних, правових та інформаційних механізмах збереження життя, здоров'я і працездатності працівника [42], [43].

4.1 Характеристика життєдіяльності людини у системі «людина – машина – середовище існування»

У сучасних умовах значна частина професійної й повсякденної діяльності здійснюється у складних людино-машинних системах. У таких системах людина сприймає інформацію, приймає рішення і виконує керувальні дії, машина забезпечує оброблення та подання даних, а середовище визначає фізичні, психофізіологічні та соціальні умови взаємодії. Безпечне функціонування можливе лише за узгодженості всіх трьох складових, оскільки порушення в одній із них здатне знизити працездатність людини або спричинити помилкові дії [42].

У межах інформаційної системи «Book Twin» людиною є користувач або розробник, машиною – персональний комп'ютер разом із програмним забезпеченням, а середовищем – робоче місце, умови освітлення і мікроклімату, інформаційне навантаження та організація праці. Користувач вводить оцінки книг, переглядає рекомендації та інтерпретує пояснення, тоді як система зберігає дані, виконує обчислення і формує відповідь. Якість зворотного зв'язку між людиною та програмою безпосередньо впливає на правильність дій.

Найважливішими характеристиками людини-оператора є здатність тривалий час підтримувати увагу, розрізняти візуальні сигнали, запам'ятовувати

послідовність дій і приймати рішення за неповної інформації. Під час роботи з рекомендаційною системою навантаження є переважно зоровим та нервово-емоційним. Воно зростає за надмірної кількості елементів інтерфейсу, нечітких повідомлень, складної навігації або суперечливих результатів. Тому інтерфейс має надавати користувачеві зрозумілий стан системи, передбачувані елементи керування і можливість виправити помилково введені дані.

Під час проєктування «Book Twin» передбачено окремі вкладки для оцінювання книг, рекомендацій, антирекомендацій, книжкового двійника та показників якості. Такий поділ зменшує одночасне інформаційне навантаження і допомагає користувачеві послідовно переходити від введення даних до аналізу результатів. Текстові пояснення доповнюють числові прогнози, а попередження біля антирекомендацій уточнює, що йдеться про індивідуальну відповідність, а не про об'єктивну якість книги.

Небезпеки в системі «людина – машина – середовище» можуть мати фізичний, ергономічний та психофізіологічний характер. До них належать тривала статична поза, перевантаження зорового аналізатора, незручне розміщення екрана і пристроїв введення, монотонність, дефіцит перерв і підвищена напруженість під час пошуку помилок у програмному коді. Вимоги до роботи з екранними пристроями передбачають достатній простір для зміни робочого положення, ергономічне розташування елементів робочого місця, належне освітлення та регламентовані перерви [45].

Зниження ризику досягається не одним заходом, а їх поєднанням. Організаційні заходи охоплюють чергування періодів роботи і відпочинку, планування складних завдань, контроль тривалості безперервної роботи та навчання безпечним прийомам. Технічні заходи включають використання справного обладнання, стабільного зображення, регульованого екрана, клавіатури й крісла. Програмні заходи полягають у зменшенні кількості зайвих дій, перевірки введених значень, повідомленні про помилки та збереженні результатів без необхідності повторного введення. Для «Book Twin» це означає

не лише технічну працездатність алгоритмів, а й створення зрозумілої, передбачуваної та зручної взаємодії.

4.2 Інформація в системі управління охороною праці

Система управління охороною праці є сукупністю взаємопов'язаних організаційних рішень, процедур, відповідальних осіб і ресурсів, спрямованих на запобігання травмам, професійним захворюванням та аваріям. Її ефективність значною мірою залежить від інформації, на основі якої виявляють небезпеки, оцінюють ризики, планують профілактичні заходи і контролюють їх виконання [43].

Інформаційне забезпечення охорони праці охоплює нормативно-правові акти, відомості про умови праці, результати перевірок робочих місць, дані про інструктажі та навчання, повідомлення про небезпечні події, плани заходів і документи щодо їх виконання. Закон України «Про охорону праці» покладає на роботодавця обов'язок створити безпечні умови, організувати управління охороною праці та інформувати працівників про небезпечні й шкідливі фактори [44].

Для прийняття обґрунтованих рішень інформація повинна бути достовірною, повною, своєчасною, зрозумілою та придатною до перевірки. Неповні або застарілі записи можуть призвести до неправильного оцінювання ризику, пропуску строків навчання чи невиконання профілактичних заходів. Тому важливо фіксувати джерело відомостей, дату їх отримання, відповідальну особу та історію внесених змін.

Інформаційний цикл у системі управління охороною праці включає збирання даних, перевірку, систематизацію, зберігання, аналіз, прийняття рішення та подальший контроль. Наприклад, повідомлення про несправність обладнання спочатку реєструється, потім оцінюється рівень небезпеки, визначаються відповідальні особи та строк усунення, а після виконання заходу

робиться відмітка про результат. Такий цикл забезпечує простежуваність управлінського рішення від виявлення проблеми до її усунення.

Цифрові інформаційні системи дають змогу вести електронні журнали, формувати нагадування, розмежовувати права доступу, створювати звіти та швидко знаходити потрібні записи. Автоматизація зменшує ризик втрати документів і полегшує контроль строків, але не скасовує відповідальність працівників за правильність введених даних. Перед використанням інформації необхідно перевіряти її актуальність і відповідність нормативним вимогам.

Для працівників, які постійно використовують комп'ютер, окремим інформаційним напрямом є облік умов роботи з екранними пристроями. Роботодавець має повідомляти про наявні небезпечні та шкідливі фактори, організувати навчання, забезпечувати відповідність робочого місця встановленим вимогам і вживати заходів для усунення виявлених ризиків [45]. Тому в системі управління доцільно зберігати дані про інструктажі, оцінювання робочих місць, регламентовані перерви та виконання коригувальних заходів.

Інформація з охорони праці може містити персональні дані працівників, результати медичних оглядів, відомості про інциденти та внутрішні перевірки. Доступ до таких записів повинен надаватися лише уповноваженим особам. Доцільно використовувати автентифікацію, розмежування прав, резервне копіювання і журналювання змін. Захист інформації є складовою надійності системи управління, оскільки втрата або несанкціонована зміна даних може вплинути на безпеку працівників.

4.3 Висновок до четвертого розділу

У четвертому розділі розглянуто взаємодію людини з комп'ютерною технікою та програмним середовищем у межах системи «людина – машина – середовище існування». Встановлено, що безпечність такої взаємодії залежить від відповідності інтерфейсу можливостям людини, ергономічної організації

робочого місця, належного зворотного зв'язку та дотримання раціонального режиму праці й відпочинку.

Для інформаційної системи «Book Twin» важливими є зрозуміла навігація, перевірка введених оцінок, пояснення результатів і попередження про призначення антирекомедацій. Такі рішення знижують імовірність помилкових дій користувача та зменшують інформаційне навантаження. Під час розроблення системи необхідно також враховувати вимоги безпеки й захисту здоров'я працівників, які постійно працюють з екранними пристроями.

Проаналізовано роль інформації у системі управління охороною праці. Визначено, що ефективне управління потребує достовірних і своєчасних даних про небезпеки, умови праці, навчання, інциденти та виконання профілактичних заходів. Цифрові засоби спрощують реєстрацію, аналіз і контроль, однак потребують перевірки даних, розмежування доступу, резервного копіювання та захисту від несанкціонованих змін.

Отже, поєднання ергономічного проектування людино-машинної взаємодії та якісного інформаційного забезпечення охорони праці створює умови для безпечної й стабільної роботи. Використання сучасних навчальних видань з безпеки життєдіяльності та професійної безпеки дозволило врахувати актуальні підходи до аналізу небезпек, організації праці та управління ризиками.

Дотримання розглянутих вимог дає змогу знизити вплив професійних ризиків, підтримувати працездатність розробника та своєчасно реагувати на виявлені порушення. Отже, безпека праці залежить не лише від технічного оснащення робочого місця, а й від систематичного контролю умов праці, актуальності інформації та відповідального виконання профілактичних заходів.

ВИСНОВКИ

У кваліфікаційній роботі розроблено інтелектуальну інформаційну систему «Book Twin», призначену для аналізу читацьких уподобань, прогнозування оцінок книг і формування персоналізованих рекомендацій. Запропоноване рішення поєднує колаборативну та змістову фільтрацію, підтримує роботу з новими локальними користувачами, визначає книжкового двійника, формує рекомендації й антирекомендації та надає пояснення отриманих результатів.

У першому розділі кваліфікаційної роботи освітнього рівня «Бакалавр»:

- Подано характеристику предметної області книжкових рекомендаційних систем і визначено основні проблеми персоналізованого вибору літератури.

- Розглянуто функціональні можливості сервісів Goodreads, The StoryGraph, LibraryThing і BookBub.

- Проаналізовано колаборативну та змістову фільтрацію, матричну факторизацію SVD, метод TF-IDF і гібридні рекомендаційні моделі.

- Встановлено, що застосування лише одного рекомендаційного підходу не забезпечує повного врахування читацьких уподобань.

- Обґрунтовано доцільність поєднання прогнозів на основі користувацьких оцінок із текстовими характеристиками книг.

- Сформульовано завдання розроблення інформаційної системи, яка повинна підтримувати локальні профілі, прогнозування оцінок, рекомендації, антирекомендації, пошук книжкового двійника та пояснення результатів.

У другому розділі кваліфікаційної роботи:

- Сформовано функціональні та нефункціональні вимоги до інформаційної системи «Book Twin».

- Спроектовано модульну архітектуру системи, що охоплює вебінтерфейс, прикладну логіку, рекомендаційні моделі, локальну базу даних і збережені програмні артефакти.

- Охарактеризовано набір даних Goodbooks-10k та виконано підготовку книг, користувацьких оцінок і метаданих.
 - Обґрунтовано використання розрідженого представлення матриці «користувач – книга» та центрування оцінок відносно середнього значення користувача.
 - Спроектовано SVD-модель із 40 латентними компонентами та змістову модель на основі TF-IDF із обмеженням до 5000 ознак.
 - Визначено ваги гібридного прогнозу: 0,85 для SVD-складової та 0,15 для змістової складової.
 - Запропоновано метод fold-in для формування латентного представлення нового користувача без повного перенавчання SVD-моделі.
 - Спроектовано алгоритм пошуку книжкового двійника з урахуванням косинусної подібності, узгодженості оцінок, кореляції Пірсона та кількості спільно оцінених книг.
 - Розроблено структуру локальної бази SQLite для збереження профілів та оцінок.
 - Сформовано методику експериментального оцінювання з використанням метрик Precision@10, Recall@10, HitRate@10 і NDCG@10.
- У третьому розділі кваліфікаційної роботи:
- Обґрунтовано вибір мови Python і бібліотек pandas, NumPy, SciPy, scikit-learn, Streamlit та SQLite.
 - Реалізовано підготовку очищених даних, навчання SVD-моделі, побудову TF-IDF-матриці та збереження програмних артефактів.
 - Розроблено механізм формування латентного та змістового профілів нового користувача.
 - Реалізовано гібридне ранжування книг, пошук книжкового двійника, формування рекомендацій і антирекомендацій.
 - Запропоновано формування коротких текстових пояснень на основі фактичного внеску окремих рекомендаційних складових.

- Створено вебінтерфейс із локальними профілями, пошуком книг, додаванням, редагуванням і видаленням оцінок.

- Реалізовано постійне збереження користувацьких даних у базі SQLite та відновлення профілів після перезапуску програми.

- Проведено експериментальне оцінювання на 80 незалежних користувачах.

- Встановлено, що гібридна модель отримала $\text{Precision@10} = 0,1050$, $\text{Recall@10} = 0,1338$, $\text{HitRate@10} = 0,6375$ та $\text{NDCG@10} = 0,1670$.

- Підтверджено, що гібридний підхід перевершує окрему SVD-модель і базову стратегію рекомендації популярних книг за всіма використаними метриками.

- Виконано функціональне тестування створення профілів, пошуку книг, керування оцінками, формування рекомендацій, пошуку книжкового двійника та відновлення даних після перезапуску.

У розділі «Безпека життєдіяльності, основи охорони праці» розглянуто взаємодію людини, комп'ютерної техніки та робочого середовища під час розроблення і використання інформаційної системи. Висвітлено основні фактори, що впливають на працездатність користувача персонального комп'ютера, зокрема зорове й нервово-емоційне навантаження, тривале статичне положення та інформаційне перевантаження. Розглянуто роль своєчасної, достовірної та захищеної інформації в системі управління охороною праці, а також значення електронного обліку, контролю профілактичних заходів і розмежування доступу до даних.

Отже, поставлену мету кваліфікаційної роботи досягнуто. Розроблена система забезпечує повний цикл персоналізації: від створення профілю й введення користувацьких оцінок до прогнозування ставлення до непрочитаних книг, формування рекомендацій та антирекомендацій і пояснення отриманого результату.

ПЕРЕЛІК ДЖЕРЕЛ

- 1 Ricci F., Rokach L., Shapira B. Introduction to Recommender Systems Handbook. Recommender Systems Handbook. Boston : Springer, 2011. P. 1–35. DOI: 10.1007/978-0-387-85820-3_1.
- 2 Burke R. Hybrid Recommender Systems: Survey and Experiments. User Modeling and User-Adapted Interaction. 2002. Vol. 12, No. 4. P. 331–370. DOI: 10.1023/A:1021240730564.
- 3 Çano E., Morisio M. Hybrid Recommender Systems: A Systematic Literature Review. 2019. arXiv:1901.03888.
- 4 Vaz P. C., Matos D. M., Martins B., Calado P. Improving an Hybrid Literary Book Recommendation System through Author Ranking. 2012. arXiv:1203.5324.
- 5 Goodreads. Meet your next favorite book. URL: <https://www.goodreads.com/> (дата звернення: 21.06.2026).
- 6 The StoryGraph. Because life's too short for a book you're not in the mood for. URL: <https://www.thestorygraph.com/> (дата звернення: 21.06.2026).
- 7 LibraryThing. Quick Start. URL: <https://www.librarything.com/quickstart.php> (дата звернення: 21.06.2026).
- 8 BookBub. What is BookBub. URL: <https://www.bookbub.com/home/overview.php> (дата звернення: 21.06.2026).
- 9 Resnick P., Iacovou N., Suchak M., Bergstrom P., Riedl J. GroupLens: An Open Architecture for Collaborative Filtering of Netnews. Proceedings of CSCW'94. New York : ACM, 1994. P. 175–186. DOI: 10.1145/192844.192905.
- 10 Sarwar B., Karypis G., Konstan J., Riedl J. Item-Based Collaborative Filtering Recommendation Algorithms. Proceedings of the 10th International Conference on World Wide Web. New York : ACM, 2001. P. 285–295. DOI: 10.1145/371920.372071.
- 11 Koren Y., Bell R., Volinsky C. Matrix Factorization Techniques for

Recommender Systems. *Computer*. 2009. Vol. 42, No. 8. P. 30–37. DOI: 10.1109/MC.2009.263.

12 Pazzani M. J., Billsus D. *Content-Based Recommendation Systems. The Adaptive Web. Lecture Notes in Computer Science*. Berlin : Springer, 2007. Vol. 4321. P. 325–341. DOI: 10.1007/978-3-540-72079-9_10.

13 Salton G., Buckley C. *Term-Weighting Approaches in Automatic Text Retrieval. Information Processing & Management*. 1988. Vol. 24, No. 5. P. 513–523. DOI: 10.1016/0306-4573(88)90021-0.

14 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, No. 6. P. 734–749. DOI: 10.1109/TKDE.2005.99.

15 Zygmuntz M. *goodbooks-10k: Ten thousand books, six million ratings*. URL: <https://github.com/zygmuntz/goodbooks-10k> (дата звернення: 21.06.2026).

16 ISO/IEC/IEEE 42010:2022. *Software, systems and enterprise — Architecture description*. Geneva : International Organization for Standardization, 2022.

17 Object Management Group. *Unified Modeling Language (UML), Version 2.5.1*. 2017. URL: <https://www.omg.org/spec/UML/2.5.1> (дата звернення: 21.06.2026).

18 Streamlit. *Understanding Streamlit’s client-server architecture*. URL: <https://docs.streamlit.io/develop/concepts/architecture/architecture> (дата звернення: 21.06.2026).

19 Streamlit. *Caching overview*. URL: <https://docs.streamlit.io/develop/concepts/architecture/caching> (дата звернення: 21.06.2026).

20 SQLite. *About SQLite*. URL: <https://sqlite.org/about.html> (дата звернення: 21.06.2026).

- 21 Python Software Foundation. `sqlite3` — DB-API 2.0 interface for SQLite databases. URL: <https://docs.python.org/3/library/sqlite3.html> (дата звернення: 21.06.2026).
- 22 Scikit-learn developers. `TruncatedSVD`. URL: <https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.TruncatedSVD.html> (дата звернення: 21.06.2026).
- 23 Scikit-learn developers. `TfidfVectorizer`. URL: https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html (дата звернення: 21.06.2026).
- 24 Scikit-learn developers. `cosine_similarity`. URL: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.pairwise.cosine_similarity.html (дата звернення: 21.06.2026).
- 25 Järvelin K., Kekäläinen J. Cumulated Gain-Based Evaluation of IR Techniques. *ACM Transactions on Information Systems*. 2002. Vol. 20, No. 4. P. 422–446. DOI: 10.1145/582415.582418.
- 26 Scikit-learn developers. `normalize`. URL: <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.normalize.html> (дата звернення: 21.06.2026).
- 27 Python Software Foundation. Python 3.14.6 Documentation. URL: <https://docs.python.org/> (дата звернення: 21.06.2026).
- 28 pandas development team. pandas 3.0.3 documentation. URL: <https://pandas.pydata.org/docs/> (дата звернення: 21.06.2026).
- 29 Array programming with NumPy / C. R. Harris, K. J. Millman, S. J. van der Walt et al. *Nature*. 2020. Vol. 585. P. 357–362. DOI: 10.1038/s41586-020-2649-2.
- 30 SciPy 1.0: fundamental algorithms for scientific computing in Python / P. Virtanen, R. Gommers, T. E. Oliphant et al. *Nature Methods*. 2020. Vol. 17. P. 261–272. DOI: 10.1038/s41592-019-0686-2.
- 31 Scikit-learn: Machine Learning in Python / F. Pedregosa, G. Varoquaux, A. Gramfort et al. *Journal of Machine Learning Research*. 2011. Vol. 12. P. 2825–2830.

32 Streamlit. Session State API reference. URL: https://docs.streamlit.io/develop/api-reference/caching-and-state/st.session_state (дата звернення: 21.06.2026).

33 Joblib. Persistence documentation. URL: <https://joblib.readthedocs.io/en/stable/persistence.html> (дата звернення: 21.06.2026).

34 Evaluating Collaborative Filtering Recommender Systems / J. L. Herlocker, J. A. Konstan, L. G. Terveen, J. T. Riedl. ACM Transactions on Information Systems. 2004. Vol. 22, No. 1. P. 5–53. DOI: 10.1145/963770.963772.

35 Gunawardana A., Shani G. Evaluating Recommender Systems. Recommender Systems Handbook. 3rd ed. New York : Springer, 2022. P. 547–601. DOI: 10.1007/978-1-0716-2197-4_15.

36 Zhang Y., Chen X. Explainable Recommendation: A Survey and New Perspectives. Foundations and Trends in Information Retrieval. 2020. Vol. 14, No. 1. P. 1–101. DOI: 10.1561/15000000066.

37 Machine Learning system for detecting malicious traffic generated by IoT devices / Y. Klots, N. Petliak, S. Martsenko, V. Tymoshchuk, I. Bondarenko. CEUR Workshop Proceedings. 2024. Vol. 3742 : 2nd International Workshop on Computer Information Technologies in Industry 4.0 (CITI 2024), Ternopil, June 12–14, 2024. P. 97–110.

38 Information technology platform for the selection and analytical processing of information on COVID-19 / O. Duda, N. Kunanets, S. V. Martsenko, V. Nykytyuk, V. Pasichnyk. 2021 IEEE 16th International Conference on Computer Sciences and Information Technologies (CSIT), Lviv, September 22–25, 2021. Vol. 2. P. 231–238. DOI: 10.1109/CSIT52700.2021.9648839.

39 COVID-19 data collections and analytical processing / O. Duda, N. Kunanets, S. V. Martsenko, V. Nykytyuk, V. Pasichnyk. 2021 IEEE 16th International Conference on Computer Sciences and Information Technologies (CSIT), Lviv, September 22–25, 2021. Vol. 2. P. 252–257. DOI: 10.1109/CSIT52700.2021.9648658.

40 Cyber-physical systems at “Digital University” / O. Duda, O. Karnaukhov, S. Martsenko, V. Yatsyshyn. CEUR Workshop Proceedings. 2024. Vol.

3628 : 3rd International Workshop on Information Technologies: Theoretical and Applied Problems (ITTAP 2023), Ternopil–Opole, November 22–24, 2023. P. 306–314.

41 Koval V., Martsenko S., Zin M. Designing and Implementing Intelligent Lighting Control System. CEUR Workshop Proceedings. 2023. Vol. 3468 : 1st International Workshop on Computer Information Technologies in Industry 4.0 (CITI 2023), Ternopil, June 14–16, 2023. P. 241–249.

42 Грибан В. Г., Фоменко А. Є., Казначеев Д. Г. Безпека життєдіяльності та охорона праці : підручник. Дніпро : Дніпропетровський державний університет внутрішніх справ, 2022. 388 с. ISBN 978-617-8032-91-3. URL: https://fpk.in.ua/images/biblioteka/2FMB_Pravo/Hryban.-2022-Pidruchnyk-BZHD-ta-OP.pdf (дата звернення: 22.06.2026).

43 Основи професійної безпеки та здоров'я : підручник / Ю. Г. Масікевич, В. Ф. Райко, О. В. Шестопапов, А. Ю. Масікевич, О. Г. Янчик, Є. О. Семенов ; Нац. техн. ун-т «Харків. політехн. ін-т». Чернівці : Місто, 2023. 288 с.

44 Про охорону праці : Закон України від 14.10.1992 № 2694-XII. База даних «Законодавство України» / Верховна Рада України. URL: <https://zakon.rada.gov.ua/go/2694-12> (дата звернення: 22.06.2026).

45 Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями : наказ Міністерства соціальної політики України від 14.02.2018 № 207. База даних «Законодавство України» / Верховна Рада України. URL: <https://zakon.rada.gov.ua/go/z0508-18> (дата звернення: 22.06.2026).