

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління

КАПАЦІЛА Марія Ігорівна

**Інтелектуальна система обліку книжкового фонду в
бібліотеці / Intelligent Book Inventory Management System in
a Library**

спеціальність: 122 - Комп'ютерні науки
освітньо-професійна програма - Комп'ютерні науки

Кваліфікаційна робота

Виконав студент групи КНм-21
М. І. Капаціла

Науковий керівник:
к.т.н., доцент, Ліп'яніна-
Гончаренко Х.В.

Кваліфікаційну роботу
допущено до захисту:
«___» _____ 20___ р.
Завідувач кафедри
_____ М.П. Комар

ТЕРНОПІЛЬ - 2023

Факультет комп'ютерних інформаційних технологій
 Кафедра інформаційно-обчислювальних систем і управління
 Освітній ступінь «магістр»
 спеціальність: 122 – Комп'ютерні науки
 освітньо-професійна програма – Комп'ютерні науки

ЗАТВЕРДЖУЮ
 Завідувач кафедри
 _____ М.П. Комар
 « ____ » _____ 20__ р.

**ЗАВДАННЯ
 НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ
 Капацілі Марії Ігорівні**

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи

Інтелектуальна система обліку книжкового фонду в бібліотеці / Intelligent Book
 Inventory Management System in a Library

керівник роботи к.т.н., доцент, Ліп'яніна-Гончаренко Х.В.

затверджені наказом по університету від 8 грудня 2022 року № 491.

2. Строк подання студентом закінченої кваліфікаційної роботи 1 грудня 2023 р.

3. Вихідні дані до кваліфікаційної роботи: завдання на кваліфікаційну роботу студента, наукові статті, технічна література.

4. Основні питання, які потрібно розробити

1. Розглянути суть та особливостей інтелектуальних систем обліку книжкового фонду
2. Визначення принципів та етапів відбору факторів для створення рекомендацій
3. Аналіз існуючих наукових підходів у цьому напрямку
4. Огляд та аналіз існуючих інструментів системи обліку книжкового фонду
5. Розробити концепції рекомендаційної системи
6. Провести опис методів кластеризації документів для реалізації системи
7. Розробити методу виводу рекомендацій до книжкового фонду
8. Спроекувати структуру особливостей програмної реалізації
9. Створення тестової бази даних для валідації системи
10. Реалізувати розроблений метод для генерації рекомендованих книг
11. Провести тестування та оцінка ефективності системи.

5. Перелік графічного матеріалу у роботі

Схематичне відображення колаборативної фільтрації

Загальна Use-case діаграма

Структурне представлення розробленого методу

Схема компонентів frontend частини

Тестова сторінка даних книг

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 8 грудня 2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів кваліфікаційної роботи	Примітка
1	Сучасний стан і перспективи розвитку інтелектуальних систем обліку книжкового фонду	12.2022 р. – 03.2023 р.	
2	Методи та засоби для побудови інтелектуальної системи обліку книжкового фонду	03.2023 р. – 05.2023 р.	
3	Реалізація інтелектуальної системи обліку книжкового фонду	05.2023 р. – 11.2023 р.	
4	Повне завершення та представлення кваліфікаційної роботи на кафедрі	01.12.2023 р.	

Студент _____ М. І. Капаціла
підпис

Керівник роботи _____ к.т.н., доцент, Ліп'яніна-Гончаренко Х.В.
підпис

РЕЗЮМЕ

Дипломна робота на тему "Інтелектуальна система обліку книжкового фонду в бібліотеці" у рамках навчального процесу для отримання ступеня "Магістр" за спеціальністю 122 "Комп'ютерні науки" включає 87 сторінок тексту, 12 рисунків, 3 таблиці та список з 32 джерел.

Метою дослідження було створення інноваційної системи обліку книжкового фонду, яка використовує методи штучного інтелекту та машинного навчання для ефективної організації бібліотечних ресурсів та персоналізації рекомендацій для користувачів.

У роботі були застосовані методи нечіткого кластерного аналізу, алгоритми машинного навчання та сучасні підходи до обробки великих обсягів текстових даних. Також було розроблено інтерфейс користувача з використанням сучасних технологій веб-розробки.

Отримані результати демонструють значне покращення у процесах каталогізації, пошуку та рекомендації книжок, забезпечуючи більшу задоволеність користувачів та підвищення ефективності бібліотечних послуг.

Розроблена система має велике практичне значення для бібліотек різних рівнів, дозволяючи їм впроваджувати передові технології для модернізації обліку книжкових фондів та покращення взаємодії з читачами.

Ключові слова: ІНТЕЛЕКТУАЛЬНА СИСТЕМА, ОБЛІК КНИЖКОВОГО ФОНДУ, БІБЛІОТЕКА, ШТУЧНИЙ ІНТЕЛЕКТ, МАШИННЕ НАВЧАННЯ.

RESUME

The thesis titled "Intelligent Book Inventory Management System in a Library" for obtaining a Master's degree in the field of Computer Science 122 includes 87 pages of text, 12 figures, 3 tables, and a reference list of 32 sources.

The aim of the study was to develop an innovative book inventory system that utilizes artificial intelligence and machine learning techniques for efficient organization of library resources and personalization of recommendations for users.

The work employed fuzzy cluster analysis methods, machine learning algorithms, and modern approaches to processing large volumes of textual data. Additionally, a user interface was developed using contemporary web development technologies.

The obtained results demonstrate significant improvements in cataloging, searching, and book recommendation processes, enhancing user satisfaction and the efficiency of library services.

The developed system has great practical value for libraries of various levels, allowing them to implement advanced technologies for modernizing book inventory management and improving interactions with readers.

Keywords: INTELLIGENT SYSTEM, BOOK INVENTORY, LIBRARY, ARTIFICIAL INTELLIGENCE, MACHINE LEARNING.

ЗМІСТ

ВСТУП	8
1 СУЧАСНИЙ СТАН І ПЕРСПЕКТИВИ РОЗВИТКУ ІНТЕЛЕКТУАЛЬНИХ СИСТЕМ ОБЛІКУ КНИЖКОВОГО ФОНДУ	12
1.1 Суть та особливості інтелектуальних систем обліку книжкового фонду.....	12
1.2 Принципи та етапи відбору факторів для створення рекомендацій.....	18
1.3 Аналіз існуючих наукових підходів	23
1.4 Огляд і аналіз існуючих інструментів системи обліку книжкового фонду.....	27
1.5 Постановка задачі.....	32
Висновки до розділу 1.....	35
2 МЕТОДИ ТА ЗАСОБИ ДЛЯ ПОБУДОВИ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ ОБЛІКУ КНИЖКОВОГО ФОНДУ	37
2.1 Проектування рекомендаційної системи	37
2.2 Опис методів кластеризації документів	46
2.3 Метод виводу рекомендацій до книжкового фонду	52
Висновки до розділу 2.....	55
3 РЕАЛІЗАЦІЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ ОБЛІКУ КНИЖКОВОГО ФОНДУ	57
3.1 Структурні особливості програмної реалізації	57
3.2 Створення структури тестової бази даних.....	61
3.3 Реалізація розробленого методу для генерації рекомендованих книг	65
3.4 Тестування системи.....	69
Висновки до розділу 3.....	75
ВИСНОВКИ	78
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	83

ДОДАТОК А Апробація отриманих результатів	87
---	----

ВСТУП

Актуальність дослідження інтелектуальних систем обліку книжкового фонду та рекомендаційних систем в бібліотеках обумовлена швидким розвитком інформаційних технологій та зростаючою потребою в ефективному управлінні великими обсягами даних. У сучасному цифровому світі, де об'єми інформації неперервно зростають, бібліотеки стикаються з викликами, пов'язаними з організацією, зберіганням і доступом до інформації. Впровадження інтелектуальних систем обліку дозволяє не тільки оптимізувати процес каталогізації, але й розширити можливості для аналізу та категоризації книжкових колекцій, сприяючи покращенню бібліотечних послуг та задоволення потреб користувачів.

З розвитком штучного інтелекту та машинного навчання виникає унікальна можливість створити рекомендаційні системи, які здатні надавати персоналізовані пропозиції читачам на основі їхніх попередніх інтересів та поведінки. Це стає особливо важливим в контексті сучасної бібліотечної практики, де увага зосереджується на індивідуальному підході до кожного користувача. Персоналізовані рекомендації можуть значно покращити досвід користувачів, сприяючи підвищенню їх зацікавленості та задоволення від бібліотечних послуг.

У сучасному інформаційному просторі значну роль відіграє доступ до актуальної та надійної інформації. Бібліотеки, як традиційні хранителі знань, стикаються з викликом адаптації до змінних умов та потреб користувачів. Дослідження та розробка інтелектуальних систем, які можуть автоматизувати процеси сортування, каталогізації та пошуку інформації, є критично важливими для підтримки ефективності та релевантності бібліотечних послуг.

Крім того, інтелектуальні системи обліку та рекомендацій допомагають бібліотекам виявляти та аналізувати тенденції в читацьких

перевагах, що є ключовим для стратегічного планування та розвитку колекцій. Вони надають можливість відслідковувати популярні теми та жанри, визначати затребуваність різних видань, що дозволяє бібліотекам ефективно управляти своїми ресурсами та пропонувати користувачам актуальний контент.

Нарешті, дослідження в області інтелектуальних систем обліку та рекомендацій відкривають нові можливості для інтеграції технологічних інновацій в бібліотечний простір. Впровадження цих систем сприяє не лише поліпшенню внутрішніх операційних процесів бібліотек, але й забезпечує більш глибоке залучення користувачів, підвищуючи їхній інтерес до читання та дослідження, а також підвищуючи загальну інформаційну грамотність. Все це сприяє реалізації основної місії бібліотек як центрів знань та освіти.

Мета даного дослідження полягає в розробці та удосконаленні інтелектуальних систем обліку книжкового фонду та рекомендаційних систем у бібліотеках.

Досягнення цієї мети зумовило потребу теоретичних розробок, визначення та послідовного вирішення таких завдань:

1. Розглянути суть та особливостей інтелектуальних систем обліку книжкового фонду
2. Визначення принципів та етапів відбору факторів для створення рекомендацій
3. Аналіз існуючих наукових підходів у цьому напрямку
4. Огляд та аналіз існуючих інструментів системи обліку книжкового фонду
5. Розробити концепції рекомендаційної системи
6. Провести опис методів кластеризації документів для реалізації системи
7. Розробити методу виводу рекомендацій до книжкового фонду

8. Спроекувати структуру особливостей програмної реалізації
9. Створення тестової бази даних для валідації системи
10. Реалізувати розроблений метод для генерації рекомендованих книг
11. Провести тестування та оцінка ефективності системи.

Об'єктом дослідження є процес обліку книжкового фонду та рекомендацій в бібліотечних системах. Це включає в себе усі аспекти, пов'язані з управлінням книжковими колекціями, від каталогізації та інвентаризації до розробки та впровадження систем рекомендацій. Особлива увага приділяється новітнім технологіям та підходам, що забезпечують ефективність та актуальність бібліотечних послуг в умовах сучасного інформаційного суспільства.

Предметом дослідження є алгоритми та технології, які лежать в основі інтелектуальних систем обліку та рекомендаційних систем у бібліотеках. Це включає в себе методи штучного інтелекту, машинного навчання, аналітики даних, а також різноманітні стратегії обробки та аналізу великих обсягів інформації, спрямованих на підвищення ефективності та персоналізації бібліотечних послуг.

Методи дослідження включають в себе ряд підходів та технік, спрямованих на аналіз, розробку та вдосконалення систем обліку та рекомендацій. Основні з них - це емпіричні методи (аналіз даних, спостереження), теоретичні (моделювання, аналітичний огляд наявних досліджень) та експериментальні (тестування розроблених систем та алгоритмів). Також використовуються кількісні та якісні методи дослідження для оцінки ефективності та задоволеності користувачів системами рекомендацій.

Наукова новизна отриманих результатів полягає у розробці методу для систем обліку книжкового фонду та рекомендацій, який інтегрує передові технології штучного інтелекту та машинного навчання. Цей метод

включає в себе використання нечіткої кластеризації з косинусною відстанню та алгоритмів колаборативної фільтрації, що дозволяє створювати більш точні та персоналізовані рекомендації для користувачів. Завдяки глибокому аналізу великих даних і використанню комплексних метрик для оцінки схожості між книгами, новий метод значно покращує якість бібліотечного сервісу, забезпечуючи користувачам більш відповідні та задовільні варіанти вибору книг, що відкриває нові перспективи для розвитку сучасних бібліотечних інформаційних систем.

Практичне значення отриманих результатів полягає у значному поліпшенні процесу управління книжковим фондом у бібліотеках та оптимізації системи рекомендацій. Розроблений метод забезпечує більш точну та ефективну класифікацію книг, підвищує швидкість та якість обслуговування користувачів, знижуючи при цьому витрати часу та ресурсів. Інтеграція передових технологій штучного інтелекту дозволяє бібліотекам створювати персоналізовані рекомендації, що відповідають індивідуальним перевагам користувачів, збільшуючи їх задоволеність та залученість. Впровадження цих інноваційних рішень відкриває нові можливості для розвитку бібліотечних послуг, забезпечуючи більш гнучке та адаптивне реагування на сучасні інформаційні потреби суспільства.

Структура та обсяг роботи. Дипломна робота складається з вступу, трьох основних розділів, висновків, списку використаних джерел і додатків. В цілому, обсяг роботи становить 87 сторінок, на яких розміщено 12 ілюстрацій і 3 таблиці. Використаний список літератури включає 32 наукових джерела.

Апробація результатів дослідження. Основні теоретичні положення роботи й практичні результати дослідження доповідалися й обговорювалися: XI науково-технічна конференція «Інформаційні моделі, системи та технології», яка відбулася 13-14 грудня 2023 року у місті Тернопіль, Україна;

1 СУЧАСНИЙ СТАН І ПЕРСПЕКТИВИ РОЗВИТКУ ІНТЕЛЕКТУАЛЬНИХ СИСТЕМ ОБЛІКУ КНИЖКОВОГО ФОНДУ

1.1 Суть та особливості інтелектуальних систем обліку книжкового фонду

Для початку слід ознайомитись із поняттям автоматизованих бібліотечних інформаційних систем. Вони є надважливим інструментом, який використовують бібліотеки будь-якого типу для керування даними та надання послуг своїм читачам. Із постійним розвитком інформаційних технологій АІБС також невпинно розвиваються. Створюються нові тенденції щодо галузі бібліотекознавства та роботи бібліотек, як громадських так і персональних. З урахуванням активного розвитку онлайн формату та появою все більшої кількості книг у електронному форматі дуже гостро постає питання покращення систем управління бібліотеками.

З кожним роком все більше застосовуються інтелектуальні системи для обліку книжкових фондів. Отож встановимо, що ці системи використовують різні алгоритми машинного навчання і штучного інтелекту. Це дозволяє значно покращувати автоматизацію та оптимізацію процесів обліку.

Концепція інтелектуальних систем обліку книжкових фондів полягає у використанні штучного інтелекту та машинного навчання для автоматизації процесів пов'язаних із обліком книг. Вони дозволяють вирішувати ряд завдань, таких як:

- Автоматизація процесів пов'язаних із веденням обліку, а саме автоматизують ідентифікацію книг, їх каталогізацію. Дозволяють покращити контроль над рухом книг;
- Автоматичний аналіз інформації книжкового фонду, який дозволяє бібліотекам виявляти помилки, тенденції та інші

важливі показники. Для прикладу, ШІ може бути використано для виявлення помилок в даті авторі або кількості сторінок. Окрім цього можна виявляти тенденції у читанні, популярність жанрів чи авторів. Цю інформацію згодом можна застосовувати для покращення послуг і задоволення вимог користувачів;

- Автоматичне прогнозування попиту на книги. Цей аспект дозволяє бібліотекам або книгарням краще розподіляти ресурси та книги між своїми точками. До прикладу можна спрогнозувати очікувану кількість певних книг для закупівлі на основі попиту.

Створення рекомендацій для користувачів. Персоналізовані рекомендації для користувачів із книгами, які імовірно їм сподобаються дозволяють підвищити задоволеність клієнта та підняти рейтинг бібліотеки або книгарні.. Наприклад, можливе створення переліку книг, які відповідають індивідуальним інтересам користувачів. Вони можуть формуватись на основі різноманітних факторів, зокрема прочитані книги, жанр, автор.

ІСОКФ мають певний ряд переваг над звичайними системами обліку, таких як:

- Вища ефективність за рахунок використання ШІ;
- Менша трудомісткість;

Значне розширення можливостей зокрема через можливість надання рекомендаційних списків книг.

Ці системи можна розподілити на умовні дві частини. До першої можна віднести усі стандартні процеси, такі як ідентифікація, каталогізація, система видачі/повернення та відслідковування статусів книг. До другої ж частини віднесемо компоненти інтелектуалізації за допомогою штучного інтелекту.

У інтелектуальних системах для обліку книг часто застосовуються такі технології:

- Штрих-коди , QR коди та RFID-мітки для швидшої ідентифікації книг;
- Електронні системи для каталогізації, що забезпечують автоматизацію даних процесів;
- електронні системи для видачі або повернення книг. Забезпечують пришвидшення цих процесів та дозволяють швидше відслідковувати наявність та статус конкретних книг;
- функції на основі машинного навчання, які дозволяють створювати прогнози щодо попиту на книги та безліч інших можливостей для покращення системи та просування серед користувачів.

Для удосконалення систем обліку можна застосовувати декілька різних підходів, які далі розглянемо і розпочнемо із традиційного. Він передбачає застосування ручних методів і є доволі простим та не дороговартісним, порівняно з іншими, проте він також є дуже трудомістким та не надто ефективним.

Наступним є автоматизований підхід, що передбачає використання раніше досліджених АБІС. Вони забезпечують автоматизацію всіх процесів обліку, зокрема і рекомендації книг. А отже цей метод уже має переваги над традиційним, оскільки матиме значно вищу ефективність та зможе розширити функції бібліотеки.

Останнім є інтегрований підхід передбачає використання комбінації ручних і автоматизованих методів обліку. Цей підхід дозволяє поєднати переваги обох підходів. У таблиці 1.1 наведено порівняльний аналіз всіх трьох підходів.

Таблиця 1.1 - Порівняльний аналіз підходів до удосконалення

Характеристика	Традиційний підхід	Автоматизований підхід	Інтегрований підхід
Складність реалізації	Легкий	Складний	Середній
Вартість реалізації	Невисока	Висока	Середня
Ефективність	Низька	Висока	Середня
Трудомісткість	Висока	Низька	Середня
Гнучкість	Низька	Висока	Середня
Можливості розширення	Обмежена	Велика	Середня

Питання вибору підходу для удосконалення системи насправді залежить від конкретних умов бібліотеки, оскільки як можна побачити із таблиці у кожного із них різна вартість, ефективність та складність. У випадку, коли бібліотека має не значні фінансові спроможності – найкращим буде традиційний підхід. Звісно за умов достатнього забезпечення та наявності великого книжкового фонду рекомендовано застосовувати автоматизований підхід, адже він допоможе значно підвищити ефективність обліку.

На даний час найбільш перспективним напрямком все ж є розвиток систем обліку за рахунок впровадження більшої кількості ІТ-технологій. Тобто автоматизовані системи обліку для книжкового фонду на даний момент можна вважати найкращим рішенням для роботи над удосконаленням.

Давайте виділимо ряд найважливіших у даному випадку переваг цього підходу, до них можна віднести:

- можливість автоматизувати всі процеси обліку;
- значне підвищення ефективності та швидкості обліку, зменшення помилок при цьому;
- можливість розширення системи за допомогою модулів прогнозування, звітування та створення рекомендацій.

Проте, як і будь який і будь-який інший цей підхід має недоліки:

- вартість реалізації буде вищою ніж у традиційного;
- необхідність у залучені висококваліфікованих спеціалістів для впровадження системи.

Системи обліку книжкового фонду в бібліотеках є важливим інструментом для ефективного управління бібліотечними ресурсами. Вони допомагають бібліотекарям відстежувати наявність книг, їх стан, використання та інші важливі дані.

Окрім цього слід зазначити, що для покращення роботи будь яких систем такого типу можуть застосовуватись і наступні підходи:

- впровадження єдиних стандартів для обліку, що дозволяє уніфікувати ці процеси в різних бібліотеках та зробити системи більш гнучкими та масштабованими;
- використання сучасних технологій типу RFID-мітки чи штрикоду для кожної книги. Цей підхід дозволяє полегшити ідентифікацію кожного екземпляру бібліотеки.

Отож для вибору конкретного підходу удосконалення системи слід враховувати цілий спектр факторів. У нашому випадку ми будемо займатись удосконаленням систем шляхом автоматизації, а саме впровадження системи рекомендацій, яка дозволить на основі різних факторів, які детальніше розглянемо у наступній частині роботи, створювати перелік рекомендованої літератури. Для подальшого розгляду обрано саме цю частину, оскільки вона має широкий спектр для розвитку та має низку недоліків у існуючих системах. Насправді рекомендації книг є одним із дуже важливих аспектів у сучасному світі, оскільки дуже часто люди шукають щось схоже, на уже прочитану книгу. Цей функціонал дозволяє бібліотекам притягувати нових клієнтів та піднімати рейтинг задоволеності користувача.

На даний момент, після розгляду у попередній частині роботи існуючих рішень можна виділити ряд конкретних прикладів, на яких побудовані рекомендації певних сервісів, а саме:

- рекомендації на основі історії використання книг користувачем. Бібліотека може використовувати дані про те, які книги користувач читав у минулому, щоб створювати рекомендації на основі його інтересів. Наприклад, якщо користувач читав кілька книг про історію, йому можна рекомендувати інші книги про історію.
- рекомендації на основі рейтингів та відгуків інших користувачів. Бібліотека може використовувати дані про рейтинги та відгуки інших користувачів, щоб створювати рекомендації на основі популярності книг. Наприклад, якщо книга має високий рейтинг у бібліотеки, то вона, ймовірно, буде цікава іншим користувачам.
- рекомендації на основі тематики та жанру. Бібліотека може використовувати дані про тематику та жанр книг, щоб створювати рекомендації на основі інтересів користувачів. Наприклад, якщо користувач цікавиться науковою фантастикою, йому можна рекомендувати інші книги в цьому жанрі.

У більшості існуючих систем рекомендації формуються саме на жанрах та рейтингу, інколи авторі, проте не враховуючи кількості сторінок та основну тематику книги. Використання таких додаткових параметрів могло б покращити результативність системи. А саме дозволила б знаходити більший спектр книг, які є схожими за цими параметрами.

У підсумку, інтелектуальні системи обліку книжкових фондів відіграють ключову роль у сучасному управлінні бібліотечними ресурсами. Вони не лише значно підвищують ефективність та

автоматизацію процесів обліку книг, але й відкривають нові можливості для бібліотек, такі як персоналізовані рекомендації книг, прогнозування попиту та автоматизований аналіз даних. Ці технології, які включають застосування штучного інтелекту, машинного навчання та інших сучасних ІТ-інструментів, дозволяють не лише оптимізувати роботу бібліотек, але й підвищити задоволеність користувачів завдяки кращому обслуговуванню та підбору матеріалів. Інтеграція таких систем обліку є важливим кроком у розвитку бібліотечної сфери, оскільки вона забезпечує гнучкість, ефективність та сучасний підхід до обслуговування користувачів.

1.2 Принципи та етапи відбору факторів для створення рекомендацій

Принципи відбору факторів для створення рекомендацій

При виборі факторів для створення рекомендацій слід враховувати такі принципи:

- релевантність – відповідність підібраних книг із інтересами користувача.
- адекватність. Рекомендації обов'язково мають бути достатньо точними, і відповідати запиту, щоб користувач міг їх використовувати.
- актуальність – відповідність до поточних потреб користувача.
- доступність для користувача.

Відбір факторів для створення рекомендацій можна розділити на такі етапи:

- визначення цілей рекомендацій: На першому етапі необхідно визначити, які цілі необхідно досягти за допомогою рекомендацій. Наприклад, рекомендації можуть бути призначені для того, щоб:
 - задовольнити інтереси користувача.

- полегшити користувачеві пошук потрібної інформації.
- збільшити використання бібліотечних ресурсів.

Збір даних: На другому етапі необхідно зібрати дані, які будуть використовуватися для створення рекомендацій. Ці дані можуть включати:

- історію використання бібліотечних ресурсів користувачем.
- демографічні дані про користувача.
- оцінки користувачів.
- рейтинги користувачів.

Аналіз даних: На третьому етапі необхідно проаналізувати зібрані дані, щоб визначити, які фактори є найбільш релевантними для створення рекомендацій.

Відбір факторів: На четвертому етапі необхідно відібрати фактори, які будуть використовуватися для створення рекомендацій.

Оцінка рекомендацій: На п'ятому етапі необхідно оцінити рекомендації, щоб переконатися, що вони відповідають цілям, які були визначені на першому етапі.

До факторів, які можуть використовуватися для створення рекомендацій, відносяться:

- історія використання бібліотечних ресурсів користувачем: Цей фактор є одним з найважливіших для створення рекомендацій. Він дозволяє рекомендувати користувачеві ресурси, які він, ймовірно, буде зацікавлений.
- демографічні дані про користувача: Цей фактор може використовуватися для створення рекомендацій, які відповідають інтересам користувача відповідно до його віку, статі, освіти та інших демографічних характеристик.
- оцінки користувачів: Ці оцінки можуть використовуватися для створення рекомендацій, які мають високу оцінку інших користувачів.

- рейтинги користувачів: Ці рейтинги можуть використовуватися для створення рекомендацій, які високо оцінюються користувачем.

Вибір факторів для створення рекомендацій залежить від конкретних цілей, які необхідно досягти, а також від доступних даних. До прикладу Автор книги може бути важливим фактором для рекомендацій, оскільки він може дати уявлення про жанр, стиль і тему книги. Наприклад, якщо користувачеві подобаються книги Дж.К. Роулінг, то йому можна рекомендувати інші книги цієї авторки.

Кількість сторінок книги може бути важливим фактором для рекомендацій, оскільки вона може дати уявлення про складність і тривалість книги. Наприклад, якщо користувачеві подобаються короткі книги, то йому можна рекомендувати книги з меншою кількістю сторінок.

Місце розгортання подій книги може бути важливим фактором для рекомендацій, оскільки воно може дати уявлення про атмосферу і настрій книги.

Інші фактори:

- рік видання книги. Цей фактор може використовуватися для створення рекомендацій, які відповідають інтересам користувача відповідно до його віку або інтересів до певного періоду часу.
- тип книги. Цей фактор може використовуватися для створення рекомендацій, які відповідають інтересам користувача до певного типу книги.

Окрім цього слід приділити увагу основній тематиці книг, яку можна використовувати як теги для кожної книги, для прикладу «Любов і втрата».

Згідно із дослідженнями зараз люди часто справді зацікавлені в пошуку книг з певною кількістю сторінок, а точніше з межами кількості сторінок. Оскільки у такому випадку кожен користувач може створювати

власні судження щодо тривалості прочитання книги. Інколи людям важливо прочитати книгу за вихідні, що у більшості випадків означатиме, що не варта обирати книги більш ніж на 400 сторінок, оскільки залишається високий шанс не встигнути.

Задля покращення результатів пошуків проведено аналіз існуючих жанрів, та створено їхнє групування відповідно до певних параметрів. Таким чином виділено три групи жанрів:

- люди та відносини;
- природа та світ;
- культура та історія.

Кожен із них налічує низку жанрів, які найбільш схожі між собою.

Для прикладу до групи люди та відносини жанрами є:

- любовний роман;
- сімейний роман;
- психологічний роман;
- трилер;
- детектив;
- комедія.

Такий розподіл пов'язаний із тим, що кожен із жанрів групи має стосунок до людей, їхніх почуттів, відносин, дій. Звісно цей розподіл є відносним і в даному випадку використовується для покращення рекомендацій, оскільки книги матимуть певну схожість в будь-якому випадку. Однак ця тема є суперечливою та може модифікуватись в залежності від потреб користувачів. Тобто розподіл може зазнавати модифікацій після тестового періоду впровадження на основі відгуків користувачів.

Важливим для користувачів все ж прийнято вважати місце розгортання подій або ж тематику, яка висвітлюється у книзі. Тобто якщо людина хоче прочитати щось романтичне із хорошим фіналом вона обере

книгу, що матиме тематику кохання, добро і тому подібне. Також багато систем зараз пропонують серед інформації про книги надавати певні теги, типу: «добро і зло», «від ворогів до друзів» і тому подібне. За цими даними можна проводити фільтрацію інформації та підбір книг. Оскільки більш імовірно що читач вподобавши певну книгу із певними тегами захоче прочитати щось максимально схоже за визначенням тегів.

Звісно не слід пропускати і важливість авторства твору, оскільки зараз є досить частими випадки, коли читачі надають перевагу одному автору та його творам. Оскільки читач часто прив'язується саме до манери викладу інформації, швидкості розвитку подій, деталізації світу і тому подібних речах.

Ну і звісно при розробці системи не можна пройти повз загальні рейтинги книг. Звісно для багатьох читачів не такий важливий цей момент, проте він дозволяє судити про актуальність та якість книги. Багато користувачів не звертають зовсім уваги на рейтинги та цікавляться вмістом, манерою письма автора, тематикою та жанром. Проте у будь-якій системі генерації рекомендацій слід враховувати цей аспект також. При цьому не обов'язково звертати увагу лише на книги із найвищими показниками, цей параметр теж можна використовувати для побудови векторів та знаходження схожості. Оскільки доволі часто користувач може хотіти прочитати книги, які не настільки відомі, як інші та дослідити для себе нову частинку літературного ринку.

У висновку, принципи та етапи відбору факторів для створення рекомендацій в бібліотечних системах є ключовими для задоволення індивідуальних потреб користувачів. Релевантність, адекватність, актуальність та доступність книжкових рекомендацій грають важливу роль у забезпеченні цінності та корисності для користувачів. Важливо враховувати такі фактори, як історія використання, демографічні дані, оцінки та рейтинги користувачів для створення точних і релевантних

рекомендацій. Подальше дослідження і аналіз цих факторів можуть вести до створення більш ефективних і персоналізованих бібліотечних рекомендаційних систем, що в свою чергу сприяє підвищенню задоволеності та залученості користувачів.

1.3 Аналіз існуючих наукових підходів

У роботі [1] представлено інтелектуальну систему бібліотеки, яка базується на матричному розкладі та алгоритмі SSA. Система покращує ефективність пошуку інформації на 29.81%, використовуючи матричний розклад для оптимізації пошукових запитів. Алгоритм SSA використовується для прогнозування потреб у книгах, забезпечуючи точність прогнозів з відхиленням у межах 25%. Цей підхід дозволяє бібліотекам ефективніше управляти своїми ресурсами та краще задовольняти запити користувачів.

У дослідженні [2] розроблено AuRoSS, автономну роботизовану систему сканування полиць, яка включає навігаційну систему з можливістю відстеження поверхонь. Система використовує фільтрований перетворення Хафа та макро-міні маніпулятор для забезпечення високої точності відстеження. Цей підхід дозволяє автоматизувати процес інвентаризації книжкових фондів, значно підвищуючи ефективність та точність роботи бібліотек.

У статті [3] досліджується майбутнє бібліотек, підштовхнуте застосуванням штучного інтелекту в Китаї. Автори прогнозують, що майбутні бібліотеки будуть мати інтелектуальний простір, систему ресурсів та обслуговування клієнтів. Дослідження акцентує на потенціалі ШІ у покращенні бібліотечних послуг та управлінні ресурсами, пропонуючи нові можливості для розвитку бібліотечної сфери.

У роботі [4] представлено інтелектуальну систему сортування книг, яка демонструє високу точність, стабільність та масштабованість. Система розроблена для задоволення потреб щоденного управління бібліотекою, забезпечуючи ефективність та надійність процесу сортування книг. Це дозволяє бібліотекам оптимізувати робочі процеси та підвищити загальну продуктивність.

У статті [5] описано нову систему RFID та комп'ютерного зору для інвентаризації книг, використовуючи мобільного робота (RC-BI). Система досягає середньої точності 98.4% для фільтрації рівнів та високої точності для впорядкування книг. Цей підхід значно покращує стандарти інвентаризації книжкових фондів, забезпечуючи високу точність та ефективність.

У дослідженні [6] розроблено автоматичну систему інвентаризації книг на базі алгоритму глибокого навчання з EAST та CRNN. Система ефективно вирішує проблему ручної інвентаризації книг у бібліотеках, використовуючи EAST для виявлення тексту та CRNN для його розпізнавання. Цей підхід дозволяє автоматизувати та оптимізувати процес інвентаризації.

У роботі [7] представлено проект інтелектуальної системи сортування книг на базі PLC та сенсорної технології. Система інтегрує різні технології автоматичного управління, такі як RFID, сенсорні технології, технології передачі, для ефективного сортування книг. Це дозволяє бібліотекам підвищити точність та швидкість процесу сортування.

У статті [8] досліджується проектування персоналізованого алгоритму рекомендацій для мобільної інтелектуальної системи управління книгами на базі хмарних технологій. Система, розроблена на базі Java, спрощує процес управління бібліотекою та надає користувачам

зручність у пошуку книг. Цей підхід дозволяє бібліотекам надавати більш персоналізовані та ефективні послуги.

У дослідженні [9] представлено інтелектуальну мобільну систему рекомендацій книг, яка підтримує навчання на основі проблем у бібліотеках. Система використовує мобільне місцезнаходження та картографічне керівництво для підтримки співпраці під час навчання, покращуючи продуктивність пошуку книг порівняно з традиційними методами.

У роботі [10] розглядається дизайн інтелектуального робота для повернення книг у бібліотеці. Дослідження фокусується на загальній структурі дизайну робота та схемі управління системою захоплення, вивчаючи принципи роботи та характеристики робота для повернення книг.

У роботі [11] проведено огляд різних методів, які застосовуються у системах рекомендацій книг. Автори аналізують переваги використання різних технік, додаткові методи, які можуть покращити системи рекомендацій книг, та їх застосування. Основна увага приділяється розгляду того, як ці системи можуть бути адаптовані до потреб користувачів, забезпечуючи більш точні та персоналізовані рекомендації. Оцінка ефективності цих систем не надана, але наголошується на потенціалі для покращення користувацького досвіду в бібліотеках.

У дослідженні [12] розроблена хмарна система рекомендацій книг, яка використовує алгоритм колаборативного фільтрування. Система призначена для рекомендації книг різним віковим категоріям користувачів, зменшуючи людські зусилля при пошуку книг у публічних бібліотеках. Особливість цієї системи полягає у використанні хмарних технологій, що дозволяє забезпечити високу масштабованість та доступність. Автори наголошують на зручності використання системи, але не вказують на

можливі виклики, пов'язані з обробкою та захистом персональних даних користувачів.

У роботі [13] представлено гібридну систему рекомендацій книг для університетської бібліотеки. Система використовує комбінацію двох алгоритмів схожості для покращення точності рекомендацій та вирішення проблеми "холодного старту". Автори демонструють, як злиття різних методів може допомогти у виявленні більш релевантних книг для користувачів, особливо нових, які ще не мають історії взаємодії з системою. Однак, дослідження не зосереджується на можливих технічних складнощах, пов'язаних з інтеграцією та масштабуванням такої системи.

У статті [14] описано інтелектуальну систему рекомендацій книг на базі штучного інтелекту. Система використовує алгоритми ШІ для забезпечення високої точності рекомендацій, мінімізуючи помилки та оптимізуючи контрольні параметри. Автори підкреслюють, що система показує високу ефективність у відборі книг, які відповідають інтересам користувачів. Проте, дослідження не висвітлює потенційні виклики, пов'язані з обробкою великих обсягів даних та необхідністю високих обчислювальних ресурсів для функціонування системи.

Порівняльна таблиця 1.2 підсумовує ключові аспекти найближчих аналогів досліджень інтелектуальних систем обліку книжкового фонду в бібліотеках, з акцентом на рекомендаційні системи.

Таблиця 1.2 – Найближчі аналоги

Назва	Рік	Основні технології	Досягнення/Особливості
Інтелектуальна мобільна система рекомендацій книг з урахуванням місцезнаходження [9]	2013	Мобільна система з урахуванням місцезнаходження	Покращення навчальних результатів при пошуку книг
Хмарна система колаборативного фільтрування для	2020	Колаборативне фільтрування, Хмарні технології	Зменшення людських зусиль при пошуку книг

рекомендацій книг у бібліотеці [12]			
Гібридна система рекомендацій книг для університетської бібліотеки [13]	2022	Гібридна техніка рекомендацій	Покращення точності та вирішення проблеми "холодного старту"
Інтелектуальна система рекомендацій книг у бібліотеці на базі штучного інтелекту [14]	2021	Штучний інтелект	Висока точність з мінімальною помилкою та оптимізованими параметрами управління

Отже, аналізуючи існуючі наукові підходи до розробки інтелектуальних систем обліку книжкового фонду, можна відзначити значний прогрес у цій галузі. Різні дослідження представляють інноваційні рішення, що включають матричний розклад для оптимізації пошукових запитів, автоматизовані системи для інвентаризації книжкових фондів, використання штучного інтелекту для покращення бібліотечних послуг, інтелектуальні системи сортування книг, та інноваційні рекомендаційні системи на базі хмарних технологій. Ці підходи спрямовані на підвищення ефективності управління ресурсами бібліотек, оптимізацію робочих процесів, забезпечення високої точності та задоволення потреб користувачів. Порівняльний аналіз цих досліджень вказує на широкий спектр можливостей для розвитку бібліотечних систем і підкреслює важливість інтеграції новітніх технологій для створення більш ефективних та інноваційних бібліотечних послуг.

1.4 Огляд і аналіз існуючих інструментів системи обліку книжкового фонду

На сьогоднішній день звісно існують певні готові рішення створені різними компаніями. Це можуть бути як готові системи, так і набори

інструментів чи методів, які дозволяють створювати власні інтелектуальні системи обліку книжкового фонду із їх використанням.

Під час дослідження існуючих рішень було розглянуто три системи. Усі вони мають дещо схожий функціонал, як базовий так і додатковий – заснований на штучному інтелекті.

Отож першим рішенням є:

1. SirsiDynix;
2. Voyager;
3. Koha

Далі детальніше розглянемо ці системи, їх переваги та недоліки.

Розпочнемо із SirsiDynix Symphony, яка є інтегрованим бібліотечним середовищем. Задо допомогою нього можна управляти бібліотеками різного типу та розміру, в залежності від чого і формується вартість.

The screenshot displays the SirsiDynix web interface. At the top, there is a navigation menu with options like 'File', 'Preference', 'Wizards', 'Helpers', 'Modules', 'Tools', and 'Help'. Below this is a sub-menu with 'Acquisitions', 'Booking', 'Cataloging', 'Circulation', 'Configuration', 'Outreach', 'Reports', 'Requests', 'Reserves', 'Selection', 'Serial Control', and 'Utility'. The main area is titled 'Item Search and Display' and shows search results for 'Geography' in the 'MAIN' library. A table lists 84 records with columns for Title, Author, Publication info, Call Number, and Pub. Year. Below the table, a detailed record is shown for 'The economic geography of air transportation : space, time, and the freedom of the sky / John Bowen.' with fields for Leader, key, Data source, Date/time stamp, Fixed field data, LCCN, ISBN, Cataloging Source, LC Call Number, Dewey Decimal Classification Number, Personal Author, Title, Publication info, Projected pub date, Physical description, Bibliography note, Subject term, Subject term, Held by, and Bulletin heading.

Title	Author	Publication info	Call Number	Pub. Year
The economic geography of air transporta...	Bowen, John, 1966-	Abingdon, Oxon : Routledge, 2010.	HE9774 .B69 2010	2010
Understanding cultural geography : place...	Anderson, Jon, 1973-	Abingdon : Routledge, 2010.	304.2	2010
Physical geography : the key concepts	Huggett, Richard J.	London ; New York : Routledge, 2010.	GB54.5 .H84 2009	2010
Studying PGCE geography at M-level : refl...	Brooks, Clare.	New York : Routledge, 2009.	G76.5.G7 S78 2009	2009
Learning to teach geography in the secon...	Balderstone, David.	Milton Park, Abingdon, Oxon ; New York : ...	G73 .L345 2009	2009
Tourism geography : a new synthesis	Williams, Stephen, 1951 May 30-	London ; New York : Routledge, 2009.	G155.A1 W49 2009	2009
The geography of transport systems	Rodriquez, Jean-Paul, 1967-	London ; New York : Routledge, 2009.	HE323 .R63 2009	2009
Global perspectives in the geography curr...	Standish, Alex.	London ; New York : Routledge, 2009.	G76.5.US S73 2009	2009
Urban geography : a global perspective	Pacione, Michael.	Milton Park, Abingdon, Oxon ; New York : ...	GF125 .P33 2008	2009
Branding cities : cosmopolitanism, parochi...	Donald, Stephanie.	New York ; London : Routledge, c2009.	HT151 .B645 2009	2009

Рисунок 1.1 – SirsiDynix

Звісно як і більшість таких систем вона підтримує каталогізацію з стандартами MARC 21 і BIBFRAME. Також система дозволяє спрощувати

управління замовленнями, рахунками та періодичними виданнями. Окрім цього система дозволяє краще контролювати кругообіг бібліотечних матеріалів, зокрема управління штрафами та резервами. Обмін ресурсами також використовується для даної системи, тобто бібліотеки можуть обмінюватись матеріалами одна з одною.

Звісно як і всі подальші системи у ній реалізовано пошук книг, який включає в себе сортування за релевантністю. Система також надає повний набір звітів, що може використовуватись для відстеження діяльності бібліотеки.

SirsiDynix Symphony є платною системою. Вартість ліцензії залежить від розміру бібліотеки, кількості користувачів та обсягу колекцій. Проте загалом, SirsiDynix Symphony - це потужне та універсальне ILS, яке може допомогти бібліотекам будь-якого типу та розміру ефективно управляти своїми ресурсами та послугами.

Наступним є Voyager – це також інтегроване бібліотечне середовище (ILS), яке надає широкий спектр функцій для управління бібліотечними ресурсами та послугами. Воно використовується бібліотеками всіх типів і розмірів, включаючи публічні, академічні, спеціальні та шкільні бібліотеки. Voyager відоме своєю масштабованістю, гнучкістю та надійністю.

Слід зазначити, що Voyager є платною системою. Вартість ліцензії залежить від розміру бібліотеки, кількості користувачів та обсягу колекцій.

Innovative Interfaces пропонує різні варіанти ліцензування Voyager, щоб відповідати потребам бібліотек будь-якого розміру. Наприклад, бібліотеки з невеликою кількістю користувачів та колекцій можуть вибрати ліцензію на основі кількості користувачів. Бібліотеки з більшою кількістю користувачів та колекцій можуть вибрати ліцензію на основі обсягу колекцій.

Звісно система підтримує всі функції, що раніше зазначались для SirsiDynix. Зокрема Voyager підтримує систему рекомендацій, яка використовує алгоритми машинного навчання для аналізу даних про те, які книги та інші ресурси беруть користувачі бібліотеки. На основі цього аналізу система рекомендує користувачам інші ресурси, які вони можуть бути зацікавлені.

Система рекомендацій Voyager доступна в кількох варіантах:

- рекомендації на основі історії використання: Цей варіант використовує дані про те, які книги та інші ресурси користувачі бібліотеки взяли в прокат або повернули в минулому. На основі цих даних система рекомендує користувачам інші ресурси, які вони можуть бути зацікавлені.
- рекомендації на основі жанрів і тем: Цей варіант використовує дані про те, які жанри та теми користувачі бібліотеки віддають перевагу. На основі цих даних система рекомендує користувачам інші ресурси в цих жанрах і темах.
- рекомендації на основі геолокації: Цей варіант використовує дані про місцезнаходження користувача, щоб рекомендувати ресурси, які доступні в його бібліотеці.

Останнім розглянемо систему Koha - це безкоштовна і відкрита система управління бібліотеками, яка розробляється спільнотою бібліотекарів та розробників. Koha пропонує широкий спектр функцій, включаючи: каталогізацію, обмін ресурсами, управління фондами, пошук та аналіз використання бібліотечних ресурсів.

Система рекомендацій Koha доступна в двох варіантах:

Рекомендації на основі історії використання та рекомендації на основі жанрів і тем, що по суті є менш варіативним ніж у конкурентів.

Однак усі ці системи мають часткову підтримку української мови, тобто певні модулі перекладено на українську, проте далеко не всі, що є

ще одним фактором, який відзначає необхідність у дослідженні та розвитку інтелектуальних систем обліку бібліотеки.

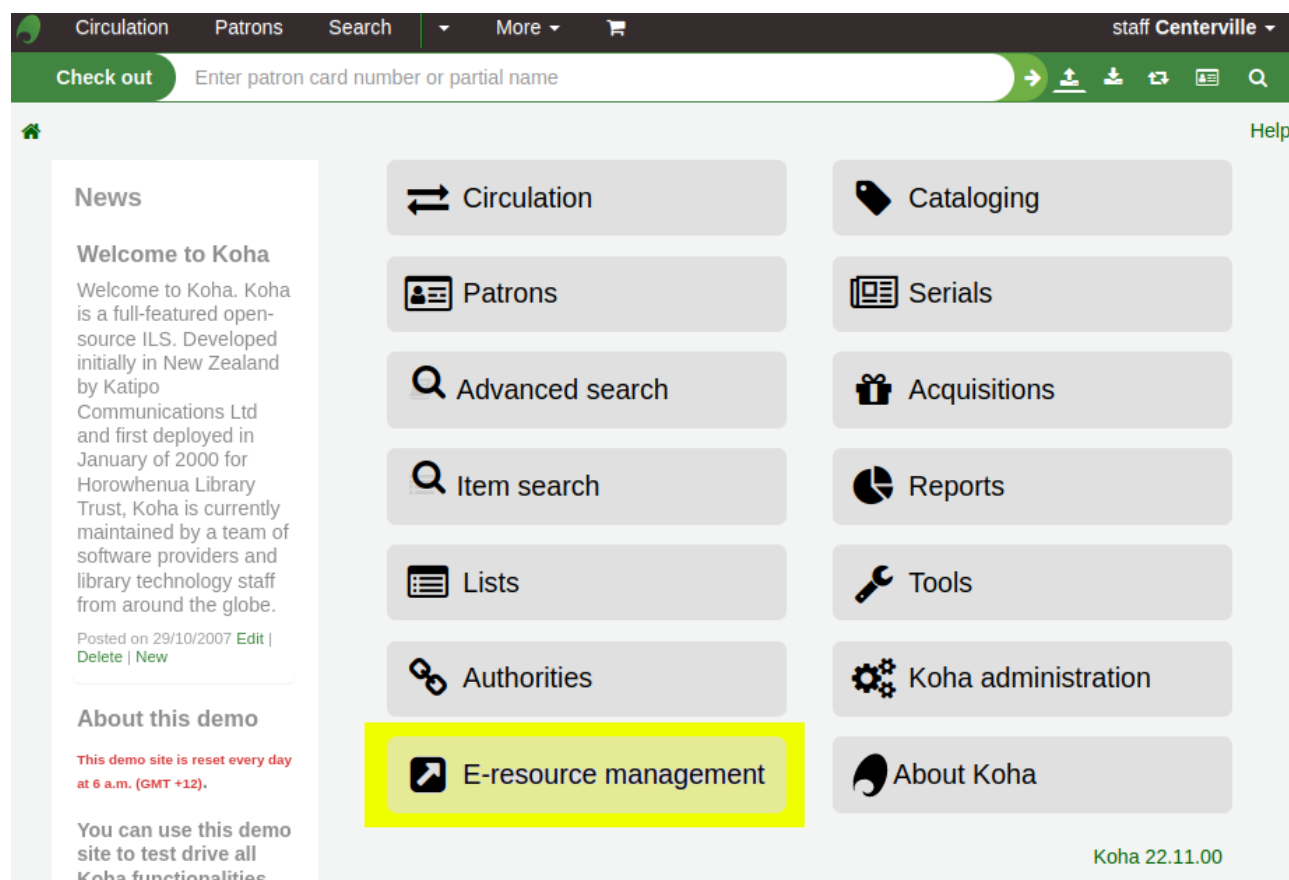


Рисунок 1.2 – Koha

Звісно підсумувавши все наведене вище, слід також зазначити, що при виборі системи ІСОКФ бібліотекам слід враховувати такі фактори, як:

- розмір бібліотеки;
- тип бібліотеки;
- фінансові можливості;
- потреби користувачів;
- підтримка мов.

Бібліотекам також слід розглянути можливість використання наборів інструментів для розробки власних ІСОКФ. Ці набори інструментів

дозволяють бібліотекам адаптувати систему ІСОКФ до своїх конкретних потреб.

Отже, аналізуючи існуючі інструменти систем обліку книжкового фонду, можна побачити, що вони пропонують різноманітний функціонал, який варіюється від простих до складних рішень, заснованих на штучному інтелекті. Системи як SirsiDynix, Voyager, та Koha, кожна має свої унікальні особливості, що роблять їх придатними для бібліотек різного типу та розміру. Важливо відзначити, що кожна з цих систем надає можливості для ефективного управління бібліотечними ресурсами, включаючи каталогізацію, управління фондами, обмін ресурсами та системи рекомендацій. Особливу увагу слід приділити підтримці мови та фінансовій доступності цих рішень. Вибір системи обліку книжкового фонду повинен враховувати специфічні потреби бібліотеки, її розмір, тип, фінансові можливості та потреби користувачів. Також варто розглянути можливість розробки власних систем за допомогою доступних інструментів, щоб забезпечити максимальну відповідність системи вимогам бібліотеки.

1.5 Постановка задачі

Актуальність дослідження інтелектуальних систем обліку книжкового фонду та рекомендаційних систем у бібліотеках пояснюється рядом ключових чинників. Перш за все, це пов'язано зі стрімким розвитком цифрових технологій та зростаючими обсягами інформації. Сучасні бібліотеки стикаються з викликами ефективного управління великими масивами даних, які включають не тільки традиційні паперові видання, але й електронні ресурси. Інтелектуальні системи дозволяють оптимізувати процеси зберігання, каталогізації та пошуку книг, роблячи бібліотечні послуги більш доступними та ефективними.

Другий важливий аспект актуальності пов'язаний з потребами сучасних користувачів. З розвитком інтернету та цифрових технологій змінилися і запити читачів. Люди очікують від бібліотек не лише доступу до широкого спектру інформації, але й індивідуалізованого підходу. Рекомендаційні системи, що базуються на алгоритмах штучного інтелекту, дозволяють надавати персоналізовані пропозиції, враховуючи інтереси та попередні запити користувачів. Це значно підвищує задоволеність користувачів та сприяє більш активному використанню бібліотечних ресурсів.

Третім аспектом є важливість інтеграції сучасних ІТ-рішень у бібліотечні системи з метою підвищення їх конкурентоспроможності. В умовах сучасного інформаційного суспільства бібліотеки конкурують не тільки між собою, але й з іншими джерелами інформації, такими як інтернет-пошукові системи, електронні бібліотеки та онлайн-ресурси. Впровадження інноваційних технологій є ключовим фактором для підтримки актуальності та значущості бібліотечних послуг.

Четверте, дослідження в цій області відкриває нові можливості для наукових та практичних розробок. Розвиток алгоритмів машинного навчання та штучного інтелекту в контексті бібліотечних систем може привести до створення нових інструментів та методів для аналізу даних, що, у свою чергу, сприятиме розвитку галузі інформаційних наук.

Нарешті, п'ятим важливим аспектом є питання забезпечення безпеки та конфіденційності даних у бібліотечних системах. Зі збільшенням обсягу цифрових даних та розвитком онлайн-послуг зростає ризик витоку персональної інформації користувачів. Дослідження в області інтелектуальних систем обліку допоможуть розробити надійні механізми захисту інформації, гарантуючи конфіденційність та безпеку користувачів бібліотек.

Мета даного дослідження полягає в розробці та удосконаленні інтелектуальних систем обліку книжкового фонду та рекомендаційних систем у бібліотеках.

Досягнення цієї мети зумовило потребу теоретичних розробок, визначення та послідовного вирішення таких завдань:

1. Розглянути суть та особливостей інтелектуальних систем обліку книжкового фонду
2. Визначення принципів та етапів відбору факторів для створення рекомендацій
3. Аналіз існуючих наукових підходів у цьому напрямку
4. Огляд та аналіз існуючих інструментів системи обліку книжкового фонду
5. Розробити концепцію рекомендаційної системи
6. Провести опис методів кластеризації документів для реалізації системи
7. Розробити методу виводу рекомендацій до книжкового фонду
8. Спроекувати структуру особливостей програмної реалізації
9. Створення тестової бази даних для валідації системи
10. Реалізувати розроблений метод для генерації рекомендованих книг
11. Провести тестування та оцінка ефективності системи.

Завдання 1-4 розглянуто в параграфах 1.1-1.4 відповідно, а виконання завдань 5-11 представлені у параграфах 2.1-3.4.

Висновки до розділу 1

Інтелектуальні системи обліку книжкового фонду відіграють ключову роль у сучасному бібліотекознавстві, надаючи значні переваги порівняно з традиційними методами. Завдяки використанню штучного інтелекту та машинного навчання, ці системи не лише автоматизують процеси ідентифікації та каталогізації книг, але й забезпечують глибокий аналіз даних книжкового фонду. Це дозволяє виявляти помилки, аналізувати тенденції читання, прогнозувати попит та створювати персоналізовані рекомендації для користувачів. Переваги таких систем включають високу ефективність, зниження трудомісткості та здатність до розширення функціоналу. Проте, вони вимагають більших витрат на реалізацію та залучення кваліфікованих спеціалістів. Вибір підходу до удосконалення системи обліку залежить від специфічних потреб та можливостей кожної бібліотеки, але в цілому, автоматизовані системи представляють собою майбутнє бібліотечного обліку.

Ефективне створення систем рекомендацій у бібліотечних інформаційних системах вимагає ретельного відбору факторів, які відповідають важливим принципам, таким як релевантність, адекватність, актуальність, та доступність. Через аналіз даних, включаючи історію використання, демографічні характеристики, оцінки та рейтинги користувачів, можна визначити ключові елементи для формування персоналізованих рекомендацій. Додаткові фактори, як-от тематика, авторство, кількість сторінок, рік видання, та тип книги, забезпечують більш глибоке розуміння інтересів читачів. Остаточне рішення щодо відбору факторів базується на цілях рекомендацій та потребах користувачів, що дозволяє створювати більш ефективні та задовільні рекомендаційні системи в бібліотеках.

Аналіз існуючих наукових підходів до інтелектуальних систем обліку книжкового фонду та рекомендаційних систем у бібліотеках демонструє значний прогрес та інноваційність у цій сфері. Використання технологій, таких як матричний розклад, алгоритми SSA, роботизовані системи сканування, штучний інтелект, та хмарні технології, значно підвищує ефективність управління ресурсами та оптимізує бібліотечні послуги. Ці підходи не лише сприяють автоматизації інвентаризації та сортування книжкових фондів, але й надають змогу розробляти точніші та персоналізовані рекомендації для користувачів, забезпечуючи підвищення задоволення та зручності їхнього досвіду. Однак, важливо враховувати й можливі виклики, пов'язані з обробкою великих обсягів даних, необхідністю високих обчислювальних ресурсів, та захистом персональних даних користувачів.

Аналіз існуючих інструментів системи обліку книжкового фонду показує, що бібліотеки мають доступ до різноманітних технологічних рішень, кожне з яких має свої унікальні особливості, переваги та обмеження. Системи, такі як SirsiDynix, Voyager і Koha, пропонують багатий набір функціональних можливостей, включаючи каталогізацію, управління замовленнями, контроль за кругообігом матеріалів та інтеграцію з системами рекомендацій на основі штучного інтелекту. Важливо відзначити, що вибір певної системи залежить від багатьох факторів, включаючи розмір бібліотеки, її тип, фінансові можливості та специфічні потреби користувачів. Незважаючи на наявність готових рішень, бібліотекам також варто розглянути використання наборів інструментів для розробки індивідуалізованих інтелектуальних систем обліку, що дозволить створити більш гнучкі та адаптивні до місцевих умов системи. Це особливо важливо у контексті підтримки мовних потреб, наприклад, забезпечення повної підтримки української мови.

2 МЕТОДИ ТА ЗАСОБИ ДЛЯ ПОБУДОВИ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ ОБЛІКУ КНИЖКОВОГО ФОНДУ

2.1 Проектування рекомендаційної системи

Системи створення рекомендацій є типом штучного інтелекту, який використовується для генерування рекомендацій для користувачів. Вони застосовуються в широкому спектрі додатків, включаючи електронну комерцію, потокове відео, соціальні мережі та інші.

Системи створення рекомендацій працюють, навчаючись на наборі даних, який містить інформацію про користувачів та об'єкти, які вони оцінили. Цей набір даних може включати інформацію про покупки, перегляди, лайки та інші форми взаємодії з користувачами.

Система може використовувати ці дані для генерування рекомендацій для нових користувачів або користувачів, які ще не оцінили певні об'єкти. Рекомендації можуть бути генеровані різними способами, включаючи:

- класифікація: система може класифікувати користувачів за їхніми інтересами, а потім рекомендувати об'єкти, які зазвичай подобаються користувачам з такими ж інтересами.
- фільтрація: система може фільтрувати об'єкти відповідно до критеріїв, які задаються користувачем. Наприклад, система може рекомендувати книги, які мають рейтинг не нижче 4 з 5.
- аналіз співвідношень: система може аналізувати споживчі дані, щоб знайти об'єкти, які часто купуються або переглядаються разом. Наприклад, система може рекомендувати фільми, які схожі на фільми, які користувач вже переглянув.

Системи створення рекомендацій можна класифікувати за різними критеріями, включаючи:

- залежність від користувачів;
- залежність від об'єктів;
- залежність від часу.

Системи створення рекомендацій можуть бути дуже ефективними для підвищення задоволеності користувачів. Вони можуть допомогти користувачам знайти об'єкти, які вони дійсно люблять, і зменшити кількість часу, який вони витрачають на пошук інформації.

Існує багато різних методів створення рекомендацій, які можна класифікувати за різними критеріями. Один із поширених способів класифікації - це залежність від користувачів. Системи можуть бути залежні від користувачів, якщо вони використовують інформацію про користувачів для генерування рекомендацій. Наприклад, система, яка використовує історію покупок користувача для рекомендування продуктів, є залежною від користувачів. Системи можуть бути незалежними від користувачів, якщо вони використовують інформацію про об'єкти для генерування рекомендацій. Наприклад, система, яка використовує рейтинги продуктів іншими користувачами для рекомендування продуктів, є незалежною від користувачів.

Інший поширений спосіб класифікації - це залежність від об'єктів. Системи можуть бути залежними від об'єктів, якщо вони використовують інформацію про об'єкти для генерування рекомендацій. Наприклад, система, яка використовує рейтинги продуктів іншими користувачами для рекомендування продуктів, є залежною від об'єктів. Системи можуть бути незалежними від об'єктів, якщо вони використовують інформацію про користувачів для генерування рекомендацій. Наприклад, система, яка використовує історію покупок користувача для рекомендування продуктів, є незалежною від об'єктів.

Методи створення рекомендацій можна також класифікувати за способом генерування рекомендацій. Один із поширених способів - це

рекомендації на основі контенту. У цьому випадку система рекомендує об'єкти, які схожі на об'єкти, які користувач уже оцінив. Наприклад, система, яка використовує історію переглядів користувача для рекомендування фільмів, є рекомендацією на основі контенту.

Інший поширений спосіб - це рекомендації на основі поведінки. У цьому випадку система рекомендує об'єкти, які, як очікується, будуть цікаві користувачеві на основі його поведінки в минулому. Наприклад, система, яка використовує історію покупок користувача для рекомендування продуктів, є рекомендацією на основі поведінки.

Рекомендації на основі машинного навчання - це ще один поширений спосіб генерування рекомендацій. У цьому випадку система навчається на наборі даних взаємодій користувачів та об'єктів, щоб прогнозувати, як користувач оцінить об'єкт. Потім система рекомендує об'єкти, які, як очікується, будуть мати високу оцінку користувача.

Вибір методу створення рекомендацій залежить від конкретних цілей, які необхідно досягти, а також від доступних даних.

Ось деякі фактори, які слід враховувати при виборі методу створення рекомендацій:

- тип ресурсів, для яких створюються рекомендації Деякі методи створення рекомендацій краще підходять для одних типів ресурсів, ніж для інших. Наприклад, колаборативна фільтрація добре працює для рекомендацій книг, фільмів та музики, але вона може бути не дуже корисною для рекомендацій продуктів або послуг.
- кількість даних, які доступні. Для деяких методів створення рекомендацій необхідні великі набори даних. Наприклад, колаборативна фільтрація вимагає даних про взаємодії користувачів з ресурсами. Якщо таких даних мало, то колаборативна фільтрація може не працювати добре.

- цілі створення рекомендацій. Деякі методи створення рекомендацій краще підходять для досягнення одних цілей, ніж для інших. Наприклад, колаборативна фільтрація добре працює для рекомендацій ресурсів, які подобаються користувачам, але вона може не дуже корисною для рекомендацій ресурсів, які будуть корисні користувачам.

Після того, як ви визначили, які фактори важливі для вас, ви можете почати розглядати різні методи створення рекомендацій.

Для створення рекомендацій книг можна використовувати різні методи. Вибір методу залежить від конкретних цілей, які необхідно досягти, а також від доступних даних.

Одним із варіантів є колаборативна фільтрація - це метод рекомендаційних систем, який використовує інформацію про взаємодії користувачів з об'єктами для генерування рекомендацій. У контексті генерування рекомендацій книг колаборативна фільтрація використовує інформацію про те, які книги користувачі оцінили, щоб рекомендувати їм інші книги, які, як очікується, будуть їм цікаві (рис. 2.1).



Рисунок 2.1 – Схематичне відображення колаборативної фільтрації

Існує два основних типи колаборативної фільтрації: фільтрація на основі схожості і фільтрація на основі спільнот.

Фільтрація на основі схожості використовує інформацію про те, які книги оцінили користувачі, щоб визначити, наскільки схожі їхні інтереси. Потім система рекомендує користувачам книги, які оцінили інші користувачі, які мають схожі інтереси.

Фільтрація на основі спільнот використовує інформацію про те, які книги оцінили користувачі, щоб визначити, які користувачі мають схожі інтереси. Потім система рекомендує користувачам книги, які оцінили інші користувачі з їхньої спільноти.

Для генерування рекомендацій книг за допомогою колаборативної фільтрації система повинна мати доступ до даних про взаємодії користувачів з книгами. Ці дані можуть включати інформацію про те, які книги користувачі оцінили, які книги вони додали до своєї бібліотеки, які книги вони повернули в бібліотеку, які книги вони прочитали, які книги вони прослухали та інші форми взаємодії.

Система потім використовує ці дані для визначення схожості між інтересами користувачів. Для цього система може використовувати різні методи, такі як:

- матриця схожості: система створює матрицю, в якій кожна комірка представляє схожість між двома користувачами.
- кластерний аналіз: система використовує кластерний аналіз для об'єднання користувачів у групи з схожими інтересами.
- машинне навчання: система використовує машинне навчання для навчання моделі, яка може прогнозувати схожість між двома користувачами.

Після того, як система визначила схожість між інтересами користувачів, вона може рекомендувати книги, які, як очікується, будуть

цікаві користувачеві. Для цього система може використовувати різні методи, такі як:

Рекомендації на основі рейтингу: система рекомендує книги, які отримали високі рейтинги від користувачів, які мають схожі інтереси.

Рекомендації на основі популярності: система рекомендує книги, які є популярними серед користувачів, які мають схожі інтереси.

Рекомендації на основі рекомендацій інших користувачів: система рекомендує книги, які оцінили інші користувачі з спільноти користувача.

Колаборативна фільтрація є ефективним методом генерування рекомендацій книг, оскільки вона враховує індивідуальні інтереси користувачів. Однак цей метод має і деякі недоліки, такі як:

- система може генерувати рекомендації, які не відповідають новим інтересам користувача.
- система може бути чутливою до шуму в даних.

Фільтрація на основі вмісту - це метод рекомендаційних систем, який використовує інформацію про самі об'єкти для генерування рекомендацій. У контексті генерування рекомендацій книг фільтрація на основі вмісту використовує інформацію про самі книги, щоб рекомендувати користувачам книги, які, як очікується, будуть відповідати їхнім інтересам (рис.2.2).



Рисунок 2.2 – Схема роботи фільтрації на основі вмісту

Для генерування рекомендацій книг за допомогою фільтрації на основі вмісту система повинна мати доступ до даних про самі книги. Ці дані можуть включати інформацію про жанр, автора, тему, ключові слова, опис, рейтинги та інші характеристики книг.

Система потім використовує ці дані для визначення схожості між книгами. Для цього система може використовувати різні методи, такі як:

- відповідність ключових слів: система визначає, які книги мають спільні ключові слова.
- відповідність тегів: система визначає, які книги мають спільні теги.
- відповідність жанрів: система визначає, які книги належать до одних і тих самих жанрів.
- відповідність авторів: система визначає, які книги написані одними і тими ж авторами.

Після того, як система визначила схожість між книгами, вона може рекомендувати книги, які, як очікується, будуть відповідати інтересам користувача. Для цього система може використовувати різні методи, такі як:

- рекомендації на основі рейтингу: система рекомендує книги, які мають високі рейтинги.
- рекомендації на основі популярності: система рекомендує книги, які є популярними.
- рекомендації на основі рекомендацій інших користувачів: система рекомендує книги, які оцінили інші користувачі.

Ось деякі фактори, які слід враховувати при виборі методу створення рекомендацій книг:

Цілі створення рекомендацій. Деякі методи створення рекомендацій краще підходять для досягнення одних цілей, ніж для інших. Наприклад, колаборативна фільтрація добре працює для рекомендацій книг, які

подобаються користувачам, але вона може не дуже корисною для рекомендацій книг, які будуть корисні користувачам.

Тип користувачів, для яких створюються рекомендації. Деякі методи створення рекомендацій краще підходять для одних типів користувачів, ніж для інших. Наприклад, колаборативна фільтрація добре працює для рекомендацій книг для користувачів із обмеженим набором даних про взаємодії з книгами, але вона може не дуже корисною для рекомендацій книг для користувачів із великим набором даних про взаємодії з книгами.

Дані, які доступні. Для деяких методів створення рекомендацій необхідні великі набори даних. Наприклад, колаборативна фільтрація вимагає даних про взаємодії користувачів з книгами. Якщо таких даних мало, то колаборативна фільтрація може не працювати добре.

Після того, як ви визначили, які фактори важливі для вас, ви можете почати розглядати різні методи створення рекомендацій книг.

Після того, як ви визначено, які фактори важливі для вас, ви можете почати розглядати різні методи створення рекомендацій книг.

Для початку розглянемо загальну Use-case діаграму інтелектуальної системи обліку книжкового фонду для зареєстрованого користувача, яка наведена на рисунку 2.3.

Цей функціонал також було розроблено під час дослідження та розширено за допомогою модуля, що може бути вмонтований в уже готові доробки в сфері інтелектуальних систем обліку книжкового фонду. Тож нижче наведено Use-case діаграму використання, які пов'язані на пряму із роботою даного модуля (рис. 2.4).

Слід відзначити що головним героєм у нашому випадку є клієнт або ж користувач бібліотеки. При цьому він попередньо має свій профіль, права та ідентифікатори. Отож для початку будемо розглядати варіант вказання користувачем книги, для якої потрібно знайти схожу літературу. У такому випадку користувач заходить на сторінку із формою для введення

назви книги. Після чого він натискає кнопку пошуку. Система обробляє його запит, знаходить книгу, яку обрав читач, проходить по алгоритму створення рекомендацій та повертає перелік із 5 книг.

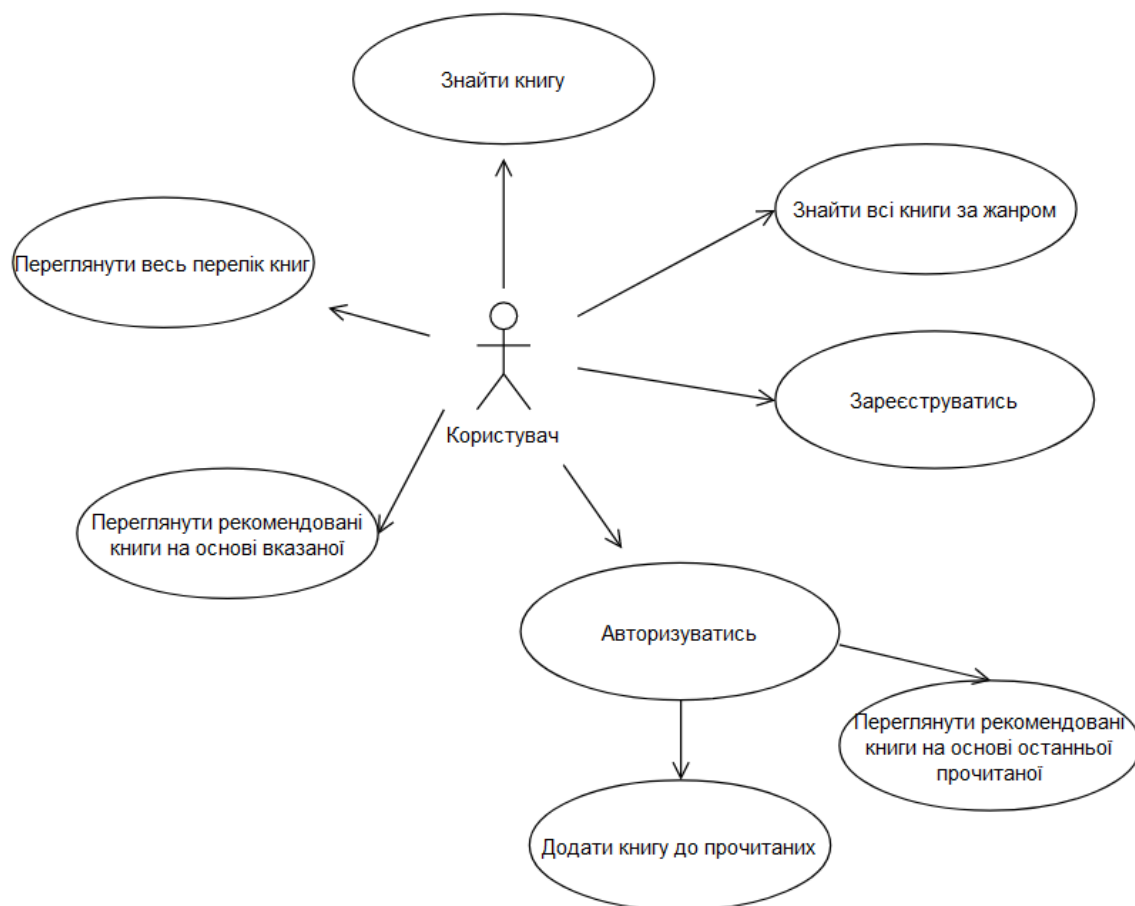


Рисунок 2.3 – Загальна Use-case діаграма

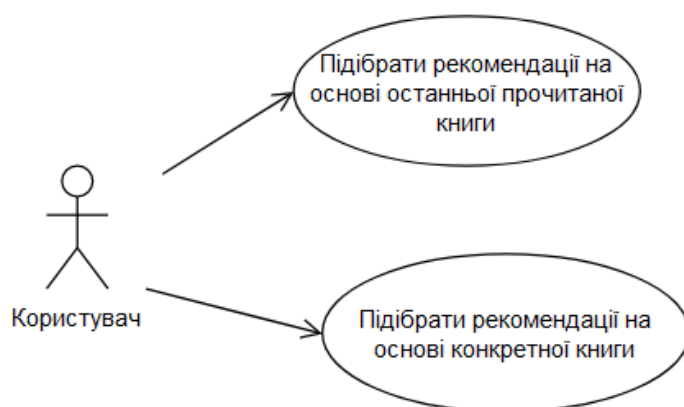


Рисунок 2.4 – Use-case діаграма для модуля

Наступним варіантом є створення рекомендацій за останньою прочитаною книгою. У такому випадку користувач може поруч натиснути кнопку нові рекомендації та отримати у відповідь перелік книг схожих на останню.

Отже, проектування рекомендаційної системи в контексті інтелектуального обліку книжкового фонду підкреслює важливість інтеграції передових методів штучного інтелекту та машинного навчання. Ефективність такої системи залежить від здатності адаптуватися до індивідуальних запитів та інтересів користувачів, використовуючи дані про їхню попередню взаємодію з ресурсами бібліотеки. Ключовими елементами в такому контексті є аналіз поведінкових моделей користувачів, класифікація за інтересами та фільтрація на основі заданих критеріїв. Методи, як колаборативна фільтрація, фільтрація на основі контенту чи поведінкові рекомендації, відіграють ключову роль у формуванні персоналізованих пропозицій. Додатково, використання сучасних технік машинного навчання може покращити точність та релевантність рекомендацій. Вибір конкретного підходу до створення рекомендацій залежить від різних факторів, включаючи наявність даних, типи користувачів, індивідуальні цілі системи, а також можливість її масштабування і адаптації до змінних умов та потреб. Ці аспекти визначають ефективність рекомендаційної системи та її здатність задовольняти запити та потреби користувачів бібліотеки.

2.2 Опис методів кластеризації документів

Як зазначалося в попередньому розділі для вирішення таких задач найкраще підходять методи нечіткого кластерного аналізу. Зокрема аналіз з косинусною відстанню.

Кластерний аналіз із косинусною відстанню — це техніка, яка використовується для групування подібних елементів або книг на основі їх косинусної подібності. Косинусна подібність — це міра, яка зазвичай використовується в аналізі тексту та пошуку інформації для визначення подібності між двома векторами, незалежно від їх величин. Це особливо корисно під час роботи з даними великого розміру, такими як текстові дані.

Розглянемо етапи кластерного аналізу з косинусною відстанню:

Представлення даних у вигляді векторів. Кожен елемент або книга є вектором у багатовимірному просторі, де кожен вимір відповідає функції або терміну.

Обчислення косинусної подібності між парами векторів. Косинус подібності обчислюється як косинус кута між двома векторами та варіюється від -1 (повністю різний) до 1 (повністю подібний). Ми абстрагуємося і для спрощення використаємо від 0 до 1, що буде легше сприйматися кінцевими користувачами.

Створимо матрицю відстані. Перетворіть значення косинусної подібності на косинусну відстань (2.1).

$$CS(A, B) = \frac{A \cdot B}{|A| \cdot |B|} \quad (2.1)$$

Косинусна відстань обчислюється як $1 - CS$, що робить метрику відстані сумісною з традиційними алгоритмами кластеризації, які очікують вимірювання несхожості або відстані.

Застосуємо ієрархічну кластероизацію або кластеризацію K-Means за допомогою косинусної матриці відстані. Ці алгоритми кластеризації групують елементи на основі їх подібності або відстані.

При ієрархічній кластеризації елементи послідовно групуються в кластери, утворюючи деревоподібну структуру, відому як дендрограма. Метод зв'язування (наприклад, повне зв'язування, середнє зв'язування) може впливати на результати кластеризації.

У кластеризації K-Means алгоритм розбиває дані на попередньо визначену кількість кластерів (K) на основі косинусної відстані.

Оскільки основою для реалізації завдання є методи кластеризації, розглянемо їх детальніше. Кластеризація документів (далі – кластеризація) – це процес виявлення природних груп у колекції книг. В нашому випадку мова ітиме про кластеризацію саме книг, бо книги по суті також є документами.

Основні типи кластеризації:

- висхідна(ієрархічна) та низхідна(плоска)
- виключаюча, перекриваюча, нечітка кластеризації
- повна і часткова кластеризації

Розглянемо відразу нечітку кластеризацію.

Нечіткі або ймовірнісні кластеризації є окремими випадками перекриваючої кластеризації. Тоді кожен об'єкт відноситься до кластера з певною вагою чи ймовірністю. Наприклад, вага від 0 до 1, де 0 – абсолютно не належить, 1 – повністю належить.

Спектральна проекція була основою для ряду алгоритмів. Особливий інтерес представляють ті, що створювати нечіткі кластери, оскільки інтуїтивно книги можуть охоплювати кілька тем. Поки ці нечіткі відносини можуть бути охоплені звичайними алгоритмами нечіткої кластеризації, які фіксують ступінь належності книг кожному кластеру нечітка кокластеризація також неявно призначає ступені приналежності до слів.

Різниця між нечіткою та звичайною спільною кластеризацією полягає в тому, що межі між кластерами у першому випадку фазифікуються відповідно до певних функцій належності.

Подібно до алгоритму нечітких середніх, варіант спільної кластеризації оптимізує нечітку цільову функцію. Важливою відмінністю є поняття агрегації, в якому алгоритм оптимізує для кластерів термінів і книг, а не лише групи книг. Агрегацію записана як формула 2.2.

$$\sum_{c=1}^C \sum_{i=1}^T \sum_{j=1}^K u_{ci} u_{cj} d_{ij} \quad (2.2)$$

При цьому кластеризуються відразу терміни і книги. Межі між кластерами не чіткі - термін або книга може входити відразу в декілька кластерів з різною вагою. Наведемо приклад реалізації в Fuzzy Codok алгоритмі за формулами:

$$u_{ci} = \frac{1}{C} + \frac{1}{2T_u} \left(\sum_{j=1}^m v_{cj} d_{ij} - \frac{1}{C} \sum_{j=1}^m v_{cj} d_{ij} \right) \quad (2.3)$$

$$v_{ci} = \frac{1}{K} + \frac{1}{2T_v} \left(\sum_{i=1}^n v_{ci} d_{ij} - \frac{1}{C} \sum_{i=1}^n v_{ci} d_{ij} \right) \quad (2.4)$$

У цих формулах u_{ci} - степінь вхождення книгою i в кластер c , v_{ci} - степінь вхождення терміну j в кластер c , d_{ij} - рівень кореляції між книгою i і терміном; m - число термінів, n - число книгів, C - число кластерів книг, K - число кластерів термінів.

T_u , T_v - параметри, їх потрібно підбирати - це слабе місце даного алгоритму; оптимальні значення параметрів залежать від колекції.

Далі розглянемо метод k середніх

Він використовується оскільки при цьому книги не додаються усі і відразу а добавляються в той кластер k якого він ближче. При цьому центроїд не перераховується як середнє від усіх елементів а коригується за формулою:

$$\mu_{k(x)}^* = \frac{\mu_{k(d)} + \eta d}{|\mu_{k(d)} + \eta d|} \quad (2.5)$$

де: $k(x)$ - кластер, до якого найближче розташована книга, μ - центроїд, d - вектор книги, η - зменшуваний коефіцієнт ітерації:

$$\eta_t = \eta_0 \left(\frac{\eta_f}{\eta_0} \right)^{\frac{t}{NM}} \quad (2.6)$$

де: N – число книг, M – число ітерацій, η_f – бажане кінцеве значення (0,01).

При цьому кластери мають бути розділені лінійно. У випадку нелінійності алгоритм дозволяє перейти в простір більш високої розмірності, де кластери будуть розділені лінійно.

На багатьох книгах вводиться функція, яка відображає ступінь близькості між книгами. Після цього проводиться кластеризація звичайним методом.

Вибір функції дуже важливий і вважається, що для текстових даних найкраще косинусна міра близькості. Взагалі у цьому методі поки що досить багато складнощів та невизначеності; обчислювальний метод дуже важкий, чим і цікава його реалізація. Існує різновид, який породжує нечітку кластеризацію (fuzzy c-means).

Зведемо дані з відкритих джерел в таблицю 2.1 переваг і недоліків різних методів.

Таблиця 2.1 – Переваги та недоліки різних методів

Алгоритм	Знаходження k	Відкритість
Ядро kmeans	ні	ні
RSSC	ні	ні
NMF	ні	ні
EM vMF	ні	так
Fuzzy codok	ні	ні
Eigencluster	так	ні
Spectral coclustering	ні	ні
Lingo	так	так
STC	так	так

Таблиця 2.1 дає можливість зробити наступні висновки:

- Якість кластеризації визначається за стандартними заходами, при цьому підсумкова кластеризація не завжди виглядає «природно»
- Існує проблема ініціалізації (ітеративні алгоритми використовують випадкову ініціалізацію)
- Існує проблема опису кластерів (міток)
- Існує проблема числа кластерів
- Існують інші методи кластеризації, можливо, вони виявляться хорошими для текстових даних

Можливо інші заходи близькості, крім косинусної, виявляться більш застосовними, але в даному дослідженні використано його як найбільш часто рекомендованого. Таким чином кращим методом на сьогоднішній день є метод нечіткої кластеризації з косинусною відстанню.

Отже, аналіз методів кластеризації документів, зокрема книг, показує, що використання нечіткого кластерного аналізу з косинусною відстанню є дуже ефективним підходом. Косинусна відстань використовується для вимірювання подібності між векторами, що представляють книги, тим самим групуючи схожі елементи. Цей підхід є важливим при аналізі текстових даних, особливо великих обсягів. Подальші етапи аналізу включають обчислення косинусної подібності між векторами, створення матриці відстані, та застосування ієрархічної кластеризації чи K-Means кластеризації. Ці методи дозволяють ефективно групувати книги на основі їх схожості чи відстані. Нечітка кластеризація в цьому контексті виокремлюється як особливо ефективна, оскільки вона враховує можливість належності однієї книги до декількох кластерів. Цей метод є ідеальним для текстових даних, де книги часто охоплюють кілька тематик. Таким чином, нечітка кластеризація з косинусною відстанню виокремлюється як найбільш відповідний метод для кластеризації книг у сучасних бібліотеках та інформаційних системах.

2.3 Метод виводу рекомендацій до книжкового фонду

На основі того вище розроблено метод який включає в себе засоби колаборативної фільтрації, які побудовані на нечіткій кластеризації з косинусною відстанню, що базується на K-Means алгоритмі відповідно до формул 2.2 – 2.5.

Ось етапи цього методу (рис.2.5):

- Етап 1. Збір даних: Збір інформації про книги, включаючи жанр, тему, автора, рейтинги та відгуки. Також збираються дані про історію читання користувачів.
- Етап 2. Фільтрація на основі контенту: Рекомендація книг, які схожі на ті, що користувач вже читав або купував, використовуючи інформацію про книги.
- Етап 3. Фільтрація на основі рейтингів: Рекомендація книг з високими загальними рейтингами, використовуючи існуючі рейтинги в базі даних.
- Етап 4. Фільтрація за загальними характеристиками: Врахування додаткових характеристик, таких як кількість сторінок книги.
- Етап 5. Створення рекомендацій: Генерація рекомендацій на основі різних факторів, включаючи жанр, автора, рейтинг, теги та кількість сторінок.
- Етап 6. Використання косинусної відстані: Застосування алгоритмів косинусної відстані для визначення подібності між книгами. Для цього використовуються теги, які відображають ключову тематику книги.
- Етап 7. Видалення прочитаних книг: Виключення з рекомендацій книг, які користувач вже прочитав, для забезпечення актуальності та релевантності рекомендацій.

Етап 8. Оновлення та навчання: Постійне оновлення бази даних та адаптація алгоритмів на основі зворотного зв'язку від користувачів для покращення точності рекомендацій.

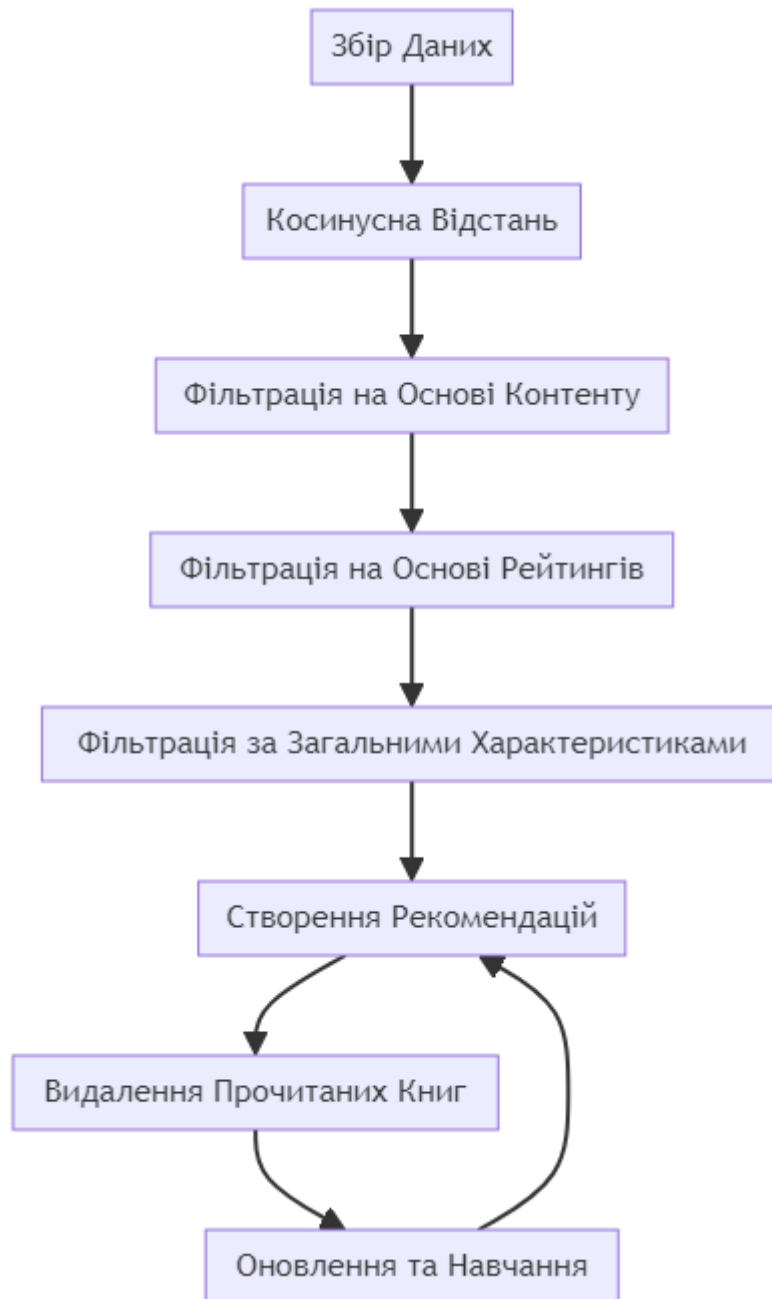


Рисунок 2.3 – Структурне представлення розробленого методу

Цей метод дозволяє створювати більш точні та персоналізовані рекомендації, враховуючи різноманітні інтереси та переваги користувачів, а також забезпечує високу релевантність рекомендованих книг.

Отже, розроблений метод виведення рекомендацій до книжкового фонду, який інтегрує засоби колаборативної фільтрації та нечіткої кластеризації з косинусною відстанню на основі алгоритму K-Means, представляє собою комплексний підхід до персоналізації рекомендацій. Процес починається зі збору даних про книги та історії читання користувачів, далі використовується косинусна відстань для визначення подібності між книгами. На наступних етапах враховуються контент книги, рейтинги, загальні характеристики та виключаються прочитані книги з рекомендацій, щоб гарантувати актуальність та релевантність пропозицій. Постійне оновлення бази даних та адаптація алгоритмів на основі зворотного зв'язку від користувачів сприяє підвищенню точності рекомендацій. Цей метод не тільки враховує різноманітні інтереси та переваги користувачів, але й ефективно забезпечує високу релевантність рекомендованих книг, збагачуючи читацький досвід і задоволення від використання бібліотечних ресурсів.

Висновки до розділу 2

У процесі проектування рекомендаційної системи важливо врахувати широкий спектр методів та підходів для створення точних та релевантних рекомендацій. Системи рекомендацій використовують різноманітні алгоритми, такі як колаборативну фільтрацію, фільтрацію на основі вмісту, аналіз співвідношень та інші техніки машинного навчання, щоб надавати персоналізовані пропозиції користувачам. Важливим є вибір методу, який найкраще відповідає цілям системи та наявним даним. Наприклад, колаборативна фільтрація ефективна для аналізу взаємодій користувачів з об'єктами та створення рекомендацій на основі цих інтеракцій. З іншого боку, фільтрація на основі вмісту зосереджується на аналізі характеристик самих об'єктів. Ефективне проектування рекомендаційної системи вимагає розуміння специфіки аудиторії, доступних даних, та цілей бібліотеки чи іншої установи, для якої система розробляється. Врахування всіх цих аспектів дозволить створити систему, яка надасть користувачам високоякісні та відповідні рекомендації, сприяючи підвищенню їх задоволення та залученості.

Аналізуючи методи кластеризації документів, особливо в контексті бібліотечних систем, можна зазначити, що методи нечіткого кластерного аналізу з косинусною відстанню є надзвичайно ефективними. Вони дозволяють групувати елементи, такі як книги, на основі їхньої взаємної подібності, використовуючи метрику косинусної відстані, яка є особливо корисною для роботи з великими текстовими даними. Процес включає в себе представлення даних у вигляді векторів, обчислення косинусної подібності, створення матриці відстаней та застосування ієрархічної кластеризації або K-Means. Нечітка кластеризація надає додаткову гнучкість у групуванні, дозволяючи елементам належати до кількох кластерів з різною вагою, що відображає реалістичнішу картину

перекривання тем у бібліотечних книжкових колекціях. Вибір методу кластеризації залежить від конкретних цілей проекту, доступних даних та специфіки документів, які потрібно класифікувати.

Метод виводу рекомендацій до книжкового фонду, який включає в себе засоби колаборативної фільтрації та нечіткої кластеризації з косинусною відстанню на основі K-Means алгоритму, представляє собою комплексний та ефективний підхід до створення персоналізованих рекомендацій. Цей метод передбачає збір різноманітних даних про книги та історію читання користувачів, обчислення подібності між книгами на основі косинусної відстані, а також використання додаткових фільтрів за контентом, рейтингами та іншими характеристиками. Генерація рекомендацій з урахуванням цих факторів сприяє підвищенню їх точності, актуальності та релевантності для користувачів. Оновлення бази даних та адаптація алгоритмів на основі зворотного зв'язку від користувачів забезпечує постійне вдосконалення системи рекомендацій, збільшуючи її ефективність та корисність у контексті бібліотечних послуг.

3 РЕАЛІЗАЦІЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ ОБЛІКУ КНИЖКОВОГО ФОНДУ

3.1 Структурні особливості програмної реалізації

Вибір мови програмування для створення системи рекомендацій книжкового фонду в інтелектуальній системі обліку зводиться до двох основних кандидатів: Python та Ruby. Python, як інтерпретована, об'єктно-орієнтована мова високого рівня, відома своєю простотою у вивченні та універсальністю у застосуванні, включаючи веб-розробку, науку про дані та машинне навчання. З іншого боку, Ruby, також об'єктно-орієнтована мова, акцентується на продуктивності та задоволенні від програмування, з особливим фокусом на веб-розробку. Обидві мови мають активні співтовариства та широкий спектр застосування.

При виборі між Python та Ruby для розробки системи рекомендацій, важливо враховувати специфіку проекту та індивідуальні навички розробників. Враховуючи потребу в швидкій розробці та легкості управління кодом, Ruby, особливо з фреймворком Ruby on Rails, видається оптимальним вибором. Ruby on Rails відомий своєю простотою, безпекою, масштабованістю та великою кількістю готових рішень, що робить його ідеальним для швидкої та ефективної розробки веб-додатків. Для фронтенду обрано React.js та Tailwind CSS, що забезпечують ефективність та стильний дизайн інтерфейсу.

Щодо середовища розробки, вибір стоїть між RubyMine від JetBrains та Visual Studio Code. RubyMine, будучи спеціалізованим IDE для Ruby, пропонує глибоку інтеграцію з Ruby on Rails, розумне кодування та високу якість розробки, але є платним. Натомість, Visual Studio Code, хоча і менш спеціалізований на Ruby, пропонує велику універсальність, багато розширень та безкоштовне використання. Враховуючи специфіку проекту

та потребу в глибокій інтеграції з Ruby on Rails, RubyMine є більш підходящим варіантом для цієї розробки.

На рисунку 3.1 наведено схему компонентів фронтенд частини та зв'язків цих компонентів між собою.

Розглянемо додатково складові деяких компонент, зокрема:

- BookList - це компонент, який відображає список книг. Він отримує дані від Recommendations або від SearchBar.

- BookDetails - це компонент, який відображає деталі книги. Він отримує дані від Recommendations або від SearchBar.

- SearchBar - це компонент, який дозволяє користувачам шукати книги. Він передає результати пошуку BookList.

- FilterBar - це компонент, який дозволяє користувачам фільтрувати список книг. Він передає фільтровані результати BookList.

- Recommendations - це компонент, який генерує рекомендації книг для користувача. Він отримує дані від API.

Далі розглянемо структуру системи з точки зору класів backend частини, а саме будову класів та їх зв'язки. На рисунку 3.2 представлена діаграма залежностей моделей, які відображають класи в Rails застосунку, що розробляється.

Згідно із нею можемо бачити зв'язки, які по суті також є зв'язками між таблицями в базі даних, адже як раніше зазначалось модель в MVC відповідає за зав'язок із базою даних. У нашому випадку маємо п'ять моделей, усі вони мають зв'язки із моделлю книг, проте не мають зв'язків між собою.

Структурні особливості програмної реалізації системи рекомендацій книжкового фонду в інтелектуальній системі обліку демонструють глибоке обмірковування вибору технологій та засобів реалізації. Використання Python або Ruby залежить від потреб проекту та вподобань розробників. Оптимальний вибір Ruby, особливо з фреймворком Ruby on Rails,

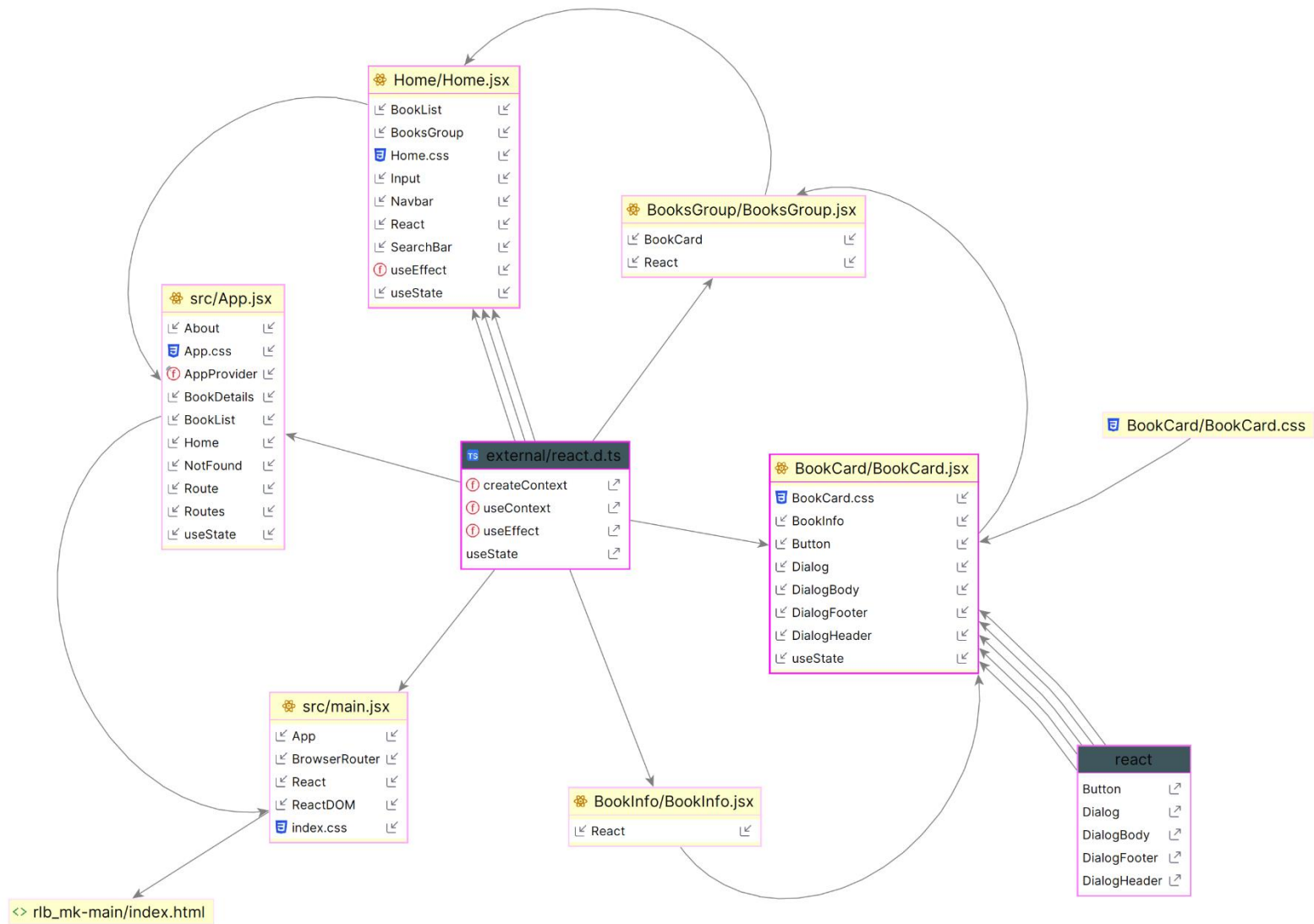


Рисунок 3.1 – Схема компонентів frontend частини

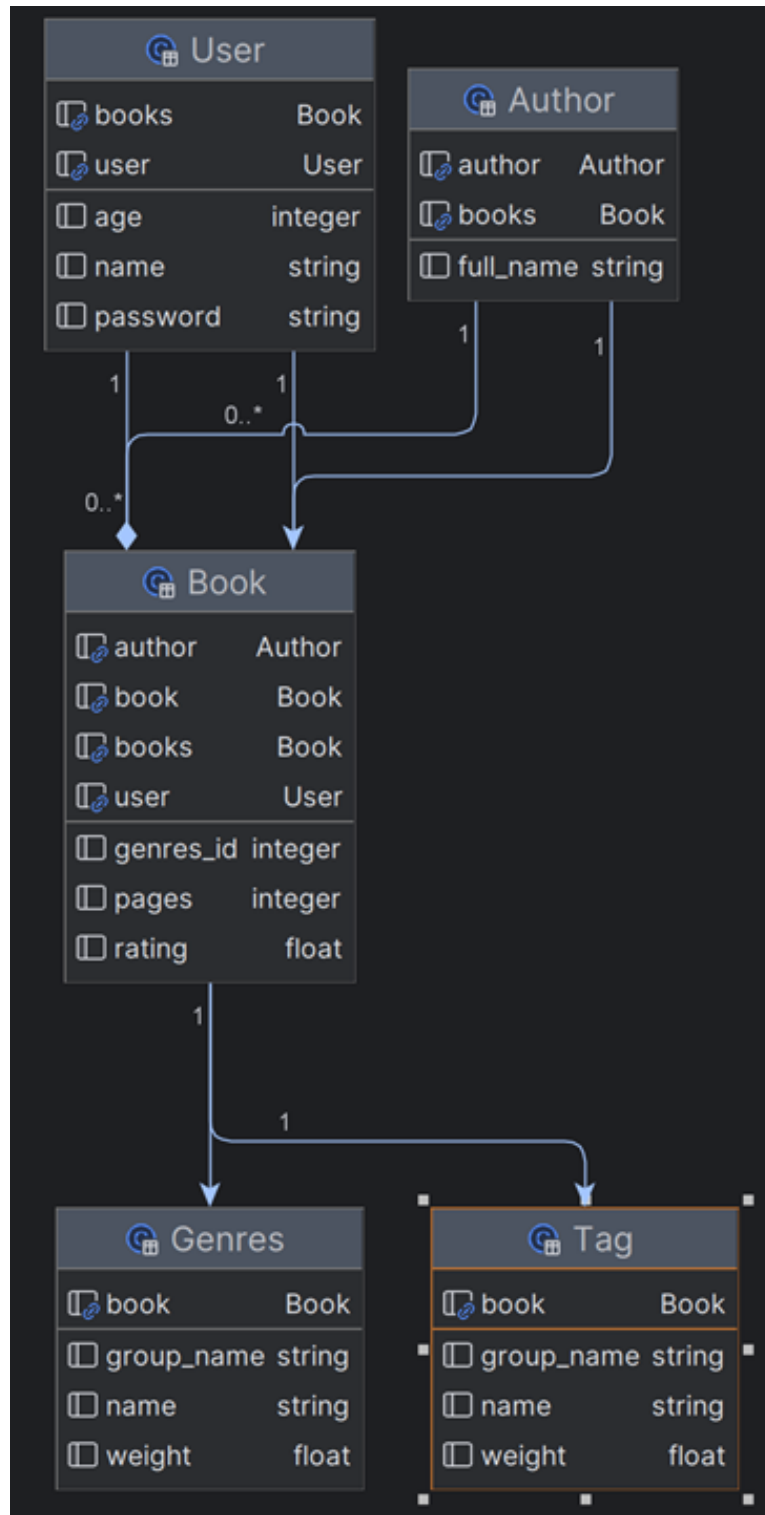


Рисунок 3.2 – Діаграма класів

забезпечує швидку та ефективну розробку веб-додатків. Для фронтенду використання React.js та Tailwind CSS забезпечує гнучкість та сучасний дизайн інтерфейсу. Вибір середовища розробки між RubyMine та Visual Studio Code залежить від потреб глибокої інтеграції з Ruby on Rails та

бюджетних обмежень, з RubyMine як більш підходящим варіантом для цього проекту.

Структура фронтенду та бекенду добре продумана, з чітким визначенням компонентів та їхніх взаємодій. На фронтенді компоненти, такі як BookList, BookDetails, SearchBar, FilterBar, та Recommendations, забезпечують ефективну взаємодію з користувачами та легкість навігації. На бекенді діаграма класів демонструє зв'язки між різними моделями, що забезпечують структуроване та ефективне управління даними.

Загалом, ця розробка підкреслює важливість вибору відповідних технологій та інструментів для створення ефективною та надійною системи рекомендацій, яка відповідає сучасним вимогам веб-розробки та користувацького досвіду.

3.2 Створення структури тестової бази даних

Структура таблиці книг для тестової бази даних, представлена у вигляді схеми бази даних (рис. 3.3). Ось детальний опис кожного поля:

1. Назва книги: Це поле містить назву книги. Це, ймовірно, текстове поле, яке може містити символи, цифри та спеціальні символи.
2. ID жанру: Це поле містить унікальний ідентифікатор жанру книги. Це може бути числове поле, яке зазвичай використовується для зв'язку з таблицею жанрів, де кожен жанр має свій унікальний ID.
3. Автор: Це поле містить ім'я автора книги. Це текстове поле, яке може включати повне ім'я автора.
4. Дата публікації: Це поле вказує на дату, коли книга була опублікована. Це, ймовірно, поле дати, яке може бути представлене у форматі року, місяця та дня.

books	
title	varchar
author	varchar
published	integer
image	varchar
genre_id	integer
created_at	datetime(6)
updated_at	datetime(6)
isbn	varchar
tags	text
pages	integer
rating	float
id	integer

Рисунок 3.3 – Структура таблиці книг

5. ISBN код: Це унікальний ідентифікатор книги, який може використовуватися для розширення системи. Це текстове поле, яке містить стандартний міжнародний номер книги.
6. Кількість сторінок: Це поле вказує на загальну кількість сторінок у книзі. Це числове поле.
7. Рейтинг: Це поле містить рейтинг книги, який може бути виражений у вигляді числа або зірок. Це може бути числове поле або поле з плаваючою комою.
8. Теги: Це поле містить теги, які описують основну тему літературного твору. Це може бути текстове поле, яке містить ключові слова або фрази, пов'язані з книгою.

Ця структура таблиці дозволяє зберігати детальну інформацію про кожну книгу, що є важливим для системи рекомендацій книг, оскільки вона дозволяє аналізувати та порівнювати книги за різними параметрами, такими як жанр, автор, рейтинг та теги.

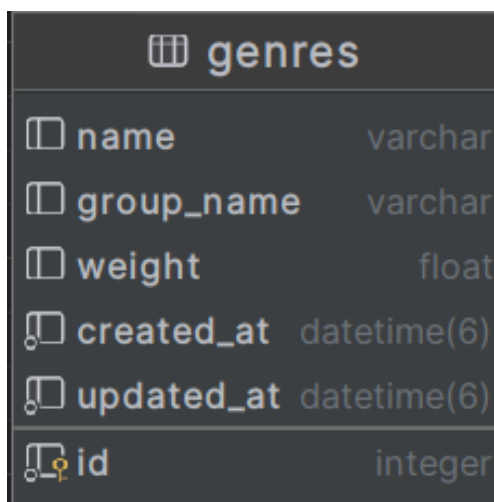
Окрім цього у даному випадку є таблиця жанрів та користувачів. Для початку розглянемо таблицю користувачів, яка містить пароль, ім'я користувача, вік та поле з ід книг, прочитаних раніше в порядку від найдавнішої до найновішої. Ось її поля:

1. **Пароль:** Це поле містить пароль користувача. З міркувань безпеки, пароль зазвичай зберігається у зашифрованому вигляді.
2. **Ім'я користувача:** Це унікальне ім'я, яке використовується користувачем для входу в систему. Це текстове поле.
3. **Вік:** Це поле містить вік користувача. Це може бути числове поле.
4. **ID прочитаних книг:** Це поле містить ідентифікатори книг, які користувач прочитав, відсортовані від найдавнішої до найновішої. Це може бути текстове поле, яке містить список ID, або може бути реалізовано як окрема таблиця зв'язків, яка з'єднає користувачів і книги.

Окрему увагу хочу приділити таблиці жанрів що відображена на рисунку 3.4. Детальніше розглянемо її поля:

1. **name** - текстове поле (`varchar`), імовірно зберігає назви жанрів.
2. **group_name** - також текстове поле (`varchar`), можливо використовується для групування жанрів за певними категоріями або підгрупами.
3. **weight** - числове поле з плаваючою крапкою (`float`), може використовуватися для вказання вагомості чи пріоритету жанру, наприклад, при сортуванні або рекомендаціях.
4. **created_at** - поле дати і часу (`datetime(6)`), фіксує час створення запису про жанр.

5. **updated_at** - поле дати і часу (datetime(6)), фіксує останнє оновлення запису про жанр.
6. **id** - ціле числове поле (integer), яке є унікальним ідентифікатором кожного запису в таблиці, тобто первинним ключем.



The image shows a screenshot of a database table structure for a table named 'genres'. The table has six columns: 'name' (varchar), 'group_name' (varchar), 'weight' (float), 'created_at' (datetime(6)), 'updated_at' (datetime(6)), and 'id' (integer). The 'id' column is highlighted with a key icon, indicating it is the primary key.

Column Name	Data Type
name	varchar
group_name	varchar
weight	float
created_at	datetime(6)
updated_at	datetime(6)
id	integer

Рисунок 3.4 – Структура таблиці жанрів

Базово таблиця (див.рис.3.4) містила лише назви жанрів. Однак задля кращого застосування та детальнішого аналізу книг на схожість ці жанри розподілено на групи. Цей розподіл відповідно дозволяє нам отримати три різних групи жанрів, серед яких: людина і відносити, природа і світ, історія та культура. Кожному із них виділено низку жанрів, які є найближчими до цієї теми. У випадку теми людина та відносини ми маємо наступні жанри: любовні, психологічні романи, комедії, детективи, трилери. Для кожного із елементів групи створено власний коефіцієнт. Коефіцієнти розраховувались шляхом розподілу одиниці на кількість жанрів категорії. У такому випадку кожен із жанрів матиме коефіцієнт, що не надто сильно відрізняється від інших та не буде спотворювати результати пошуку, при цьому не виключаючи можливості рекомендації ширшого спектру книг.

Отже, створення структури тестової бази даних для системи рекомендацій книжкового фонду передбачає ретельне планування та

визначення ключових полів і таблиць. Основною таблицею є таблиця книг, що містить важливу інформацію про кожну книгу, таку як назву, ID жанру, автора, дату публікації, ISBN код, кількість сторінок, рейтинг та теги. Ця деталізація дозволяє системі рекомендацій аналізувати та порівнювати книги за різними параметрами, забезпечуючи точність та релевантність рекомендацій. Додаткові таблиці, такі як таблиця жанрів і користувачів, збагачують базу даних і дозволяють системі краще зрозуміти переваги користувачів та надавати більш персоналізовані рекомендації. Таблиця жанрів, зокрема, розділяє жанри на групи для покращення аналітики і забезпечує додаткову гнучкість у рекомендаціях. Така багатоаспектна та добре структурована база даних є фундаментом для створення ефективної та адаптивної системи рекомендацій книжкового фонду.

3.3 Реалізація розробленого методу для генерації рекомендованих книг

Для створення конкретних рекомендацій буде проводитись пошук найбільш схожих книг до заданої тим чи іншим методом книги. При цьому першим аспектом буде приналежність жанру конкретної книги та інших книг до категорії наприклад людина і відносини. Для цього створимо окремий метод, який буде називатись `define_genre_group` і наведений у лістингу 3.1.

Лістинг 3.1 – Код методу для визначення книг зі спільної групи категорії

```
def define_genre_group
  genre_group_name = Genres.find {|genre| genre.id ==
@selected_book[:genres_id] }[:group_name]
  group_genres = Genres.where(group_name:
genre_group_name)
  group_genres_ids = group_genres.pluck(:id)
  @similar_books = []
```

```

    books = Book.where(genres_id: group_genres_ids)
  end

```

Цей метод дозволяє нам ідентифікувати для початку групу жанру до якого належить книга на основі якої створюватиметься подальший відбір рекомендацій. Далі ця інформація застосовується для кожного із елементів у базі даних, тобто для кожної книги. Окрім того, що це дозволяє зменшити вибірку даних на основі якої буде проводитись подальший відбір рекомендацій, що однозначно пришвидшить систему та забезпечить точніше формування переліку схожих книг. Це також дозволить зменшити можливість грубої похибки при підборі. Отож після знаходження усіх книг, вони передаються у конкретну змінну із якою у подальшому проводитиметься робота. Після цього можемо робити перевірку на схожість книг між собою за тегами, тобто дана функція застосовується до кожної книги із раніше отриманої вибірки. Порядок роботи алгоритму саме такий, оскільки в більшості випадків люди орієнтуються в першу чергу на жанр при виборі книги. Далі ж іде спільна тематика, яка визначається за тегами, кількість сторінок, автор, рейтинг.

Кожна книга додатково перевіряється на збіг так званих тегів, що відображають основну тему книги. Оскільки ми маємо теги у форматі стрічки схожої на наступну: “любов і смерть” ми застосуємо методи для розбиття цього речення на слова та видалення сполучних слів, таких як і, та і тому подібні для цього ми створимо певний код для видалення непотрібних слів, який наведено в лістингу 3.2.

Лістинг 3.2 – фрагмент методу для видалення сполучних слів

```

stop_words = ["і", "та"]
book_themes = book[:tags].split(" ").map { |theme|
  theme.split(" ").reject { |word| stop_words.include?(word)
}.join(" ") }

```

У даному випадку ми задаємо слова, які хочемо прибрати, як елементи масиву та далі створюємо новий масив тегів, виключаючи ці слова. При цьому до інформації застосовуємо ряд методів:

`Split` – цей метод дозволяє розподілити інформацію, що зберігається у вигляді стрічки на масив із стрічок. Для цього у нашому випадку задаємо розділювачем пробіл. За рахунок чого ми отримуємо масив типу: [“добро”, “і”, “зло”]

`Map` – використовується для проходження по кожному елементу масиву та застосування до нього певного функціоналу, у нашому випадку методу `reject`;

`Reject` – у даному випадку використовується для видалення із масивів слів, які нам не потрібні, а саме тих, що внесені до стоп слів. Це необхідно для покращення показників точності.

Оскільки у різних книг в тегах можуть зустрічатись різні сполучні слова, вони спотворюватимуть роботу програми, адже тоді практично будь-яка книга матиме щонайменше одну схожість у тегу.

Далі ми проводимо перевірку на схожість для кожної пари книг. У цьому випадку застосовуємо оператор `&`, який видає кількість збігів між книгами. Наступним кроком є перевірка книги на повторюваність, оскільки ми маємо виключити із алгоритму можливість порівняння книги самої із собою. До цього ще додаємо обов’язкову умову, що кількість схожостей в тегах повинна бути більшою або ж рівною одиниці для ширшого спектру книг. Лише за умови виконання вказаних вимог книгу буде додано до масиву літератури, яка на наступному етапі порівнюватиметься за допомогою векторів. Цей код наведено у лістингу 3.3.

Лістинг 3.3 – Код для визначення схожості тегів

```
common_themes = book_themes & selected_book_themes
```

```

        if common_themes.size >= 1 && book[:id] !=
selected_book[:id]
            @books_with_similar_themes << book
        End

```

Наступний етап полягає у створенні векторів для кожної книги на основі вагового коефіцієнту жанру, та коефіцієнту схожості книг за кількістю сторінок. У лістингу 3.4 наведено фрагмент коду, що відповідає за отримання параметрів та для створення вектора.

Лістинг 3.4 – Фрагмент коду створення векторів

```

selected_book_genre = Genres.find { |genre| genre.id ==
selected_book[:genres_id] }
features1 = [
    selected_book_genre[:weight].to_f,
    selected_book[:rating]/10,
]

book_genre = Genres.find { |genre| genre.id ==
book[:genres_id] }

features2 = [
    book_genre[:weight].to_f,
    selected_pages_weight(selected_book).to_f,
    book[:rating]/10
]
dot_product = features1.zip(features2).map { |a, b| a *
b }.reduce(:+)

```

Отож для початку ми вибираємо інформацію про книгу, згідно із якою формуємо рекомендації. Далі ж повторюємо процедуру для інших книг.

У останній стрічці лістингу 3.4 ми створюємо самі вектори схожості, при цьому використовуємо стандартні рішення, властиві для мови програмування Ruby.

Після проходження усіх цих етапів залишається зробити сортування. Далі відсортовуємо книги спочатку для відображення тих, що мають

найвищий коефіцієнт схожості, далі вводимо сортування згідно із рейтингом. Для відсутності дубляжів книг у переліку із тими, що уже були прочитані робимо попередню звірку на наявність їх у базі прочитаних користувачем. Після чого відбираємо перелік із 10 перших книг, які рекомендуємо користувачу для ознайомлення.

Отже, у процесі реалізації розробленого методу для генерації рекомендованих книг, використовується серія витончених кроків, що базуються на аналізі жанру, тегів, кількості сторінок, автора та рейтингу книги. Спочатку метод `define_genre_group` визначає групу жанру книги, що дозволяє звузити вибірку книг для аналізу. Далі, використовуючи оброблені теги, відбувається вибір книг, що мають спільну тематику з вибраною книгою. Цей підхід забезпечує точність та широту відбору рекомендацій, оскільки враховує основні інтереси читача. Після цього, створюються вектори на основі вагових коефіцієнтів жанру та порівняльної схожості книг за кількістю сторінок, що дозволяє кількісно оцінити подібність між книгами. Нарешті, використовуючи ці вектори, книги сортуються за ступенем схожості, а також за рейтингом, вилучаються дублікати книг, вже прочитаних користувачем, і формується кінцевий список рекомендованих книг. Цей складний, але ефективний підхід забезпечує персоналізовані та відповідні рекомендації для кожного користувача, враховуючи їх унікальні літературні уподобання та попередній досвід читання.

3.4 Тестування системи

Тестування системи рекомендацій - це процес оцінки того, наскільки добре система генерує рекомендації, які відповідають інтересам користувачів. Тестування може проводитися як під час розробки системи,

так і після її впровадження. У цьому випадку зосереджуємось на тестуванні під час розробки системи, адже це дослідницька розробка.

У будь-якому випадку метою тестування цієї системи є визначення того, наскільки якісно вона створює підбірки рекомендацій за параметрами. Воно обов'язково повинно проводитись, адже дозволяє виявити такі проблеми із системою:

- система рекомендує книги, які не цікаві користувачам.
- система не враховує індивідуальні інтереси користувачів.
- система рекомендує книги, які користувачі вже читали.
- система не коректно знаходить схожість між книгами;
- система порівнює книгу із заданою.

Коротко розглянемо типи тестування та звернемо увагу на ті, які застосовувались при розробці.

Найпоширенішими типами тестування систем рекомендацій є:

- тестування користувачів. У даному випадку користувачі тестують систему і оцінюють рекомендації, які вони отримують.
- аналіз даних - дані про взаємодії користувачів з системою використовуються для оцінки ефективності системи.
- порівняльне тестування. Тут система порівнюється з іншими системами рекомендацій, щоб визначити, яка система працює краще.

У даному випадку було використано тестування користувачів, та аналіз даних.

У першу чергу було проведено тестування на основі аналізу даних, а саме функціональне тестування. Обрано книгу та для неї створено вибірку схожих книг за допомогою системи рекомендацій на основі вибірки даних.

Перевірено:

- коефіцієнти схожості;

- дублювання; наявність книг, які користувач уже прочитав;
- похибки у визначенні коефіцієнтів схожості.;
- відповідність підібраних книг із вибірки із справді схожими книгами;
- поряд рекомендаційного списку.

При цьому застосовувалась окрема тестова сторінка, на якій можна було переглянути усю інформацію про книги, вона наведена на рисунку 3.5.

Отже, на рисунок 3.5 представлено фрагмент веб-інтерфейсу або додатка, де рекомендовані книги відображені в таблиці з наступними колонками:

1. **Title:** Назви книг. Перша книга називається "Історія кохання Тристана та Ізольди".
2. **Genre:** Жанр книг. Жанри варіюються від "любовний роман" до "детектив".
3. **Author:** Автори книг. Наприклад, автором першої книги є "Кретьєн де Труа".
4. **Tags:** Теги, які, ймовірно, використовуються для опису змісту або тем книг. У першій книзі тег - "Любов і трагедія".
5. **Pages:** Кількість сторінок у книзі.
6. **Rating:** Рейтинг книги. Перша книга має рейтинг 4.0.
7. **Similarity:** Стовпець, який показує схожість рекомендованої книги з якоюсь іншою книгою, можливо, книгою, яка вже прочитана користувачем або вибрана як базова для рекомендацій. Ці значення подані у вигляді чисел з плаваючою крапкою.

Вгорі таблиці є заголовок "Recommended Books for", але ім'я користувача або книги, для якої рекомендуються ці книги, не вказано. Також є розділ "About this book:", але конкретні деталі про цю книгу не надані.

Recommended Books for

About this book:

Title		Genre	Author	Tags	Rating	
Історія кохання Тристана та Ізольди		любовний роман	Кретьєн де Труа	Любов і трагедія	4.0	
Title	Genre	Author	Tags	Pages	Rating	Similarity
Лоліта	психологічний роман	Дж. Д. Селінджер	Педофілія та трагедія	267	2.0	0.9982743731749956
Ромео і Джульєтта	любовний роман	Вільям Шекспір	Любов і ненависть	236	5.0	0.9965457582448793
Доктор Хаус	трилер	Роберт Лессер	Медицина і таємниці	562	2.0	0.9191450300180576
П'ять негрят	детектив	Агата Крісті	Злочин і розслідування	288	3.0	0.8240419241993675

Рисунок 3.5 – Тестова сторінка даних книг

Таблиця (рис. 3.5) містить чотири рекомендовані книги, кожна з яких має унікальний набір значень у всіх колонках. Наприклад, друга книга в списку має назву "Лоліта", належить до жанру "психологічний роман", написана "Дж. Селінджер" (що, ймовірно, є помилкою, оскільки "Лоліта" написана Володимиром Набоковим), містить 267 сторінок, має рейтинг 2.0 та коефіцієнт схожості приблизно 0.99827.

Тестування користувачів - це найважливіший тип тестування системи рекомендацій. Цей тип тестування дозволяє отримати реальну оцінку того, як система працює для реальних користувачів.

Для тестування користувачів було проведено дослідження, в рамках якого було залучено декількох осіб до взаємодії з системою рекомендацій книг. Кожен з цих користувачів отримав можливість протестувати функціонал системи та перевірити, наскільки точно вона може підібрати книги відповідно до їхніх власних вподобань і потреб.

Процес тестування включав у себе наступні кроки:

1. Створення облікових записів користувачів: Для кожного учасника дослідження було створено окремий обліковий запис у системі. Цей обліковий запис містив інформацію про користувача, включаючи ім'я, вік та, можливо, інші характеристики.
2. Налаштування вподобань: Кожен користувач мав можливість визначити свої вподобання в обранні книг. Наприклад, вони могли вказати певні жанри, авторів або теми, які їх цікавили найбільше.
3. Додавання книг до списку прочитаних: Кожен користувач міг додавати книги до свого списку прочитаних. Це дозволяло системі збирати інформацію про те, які книги були вже прочитані кожним користувачем.
4. Аналіз та рекомендації: Система аналізувала вподобання та список прочитаних книг кожного користувача і намагалася рекомендувати нові книги, які відповідали б їхнім смакам.
5. Оцінка якості рекомендацій: Після того як користувачі отримали рекомендації, вони мали можливість оцінити, наскільки ці

рекомендації відповідали їхнім вподобанням і чи вони задоволені обраними книгами.

У результаті дослідження було відзначено, що система надала користувачам книги, які відповідали їхнім вподобанням з високою точністю та мінімальними похибками. Це свідчить про хорошу якість підбору книг та ефективність рекомендаційної системи.

Отже, тестування системи рекомендацій книг виявилось важливим кроком у процесі розробки, дозволяючи оцінити якість і точність рекомендацій, які система надає користувачам. Під час тестування було використано методи аналізу даних та прямого тестування користувачами, що дозволило отримати цінну зворотню інформацію та виявити потенційні недоліки системи. Серед ключових аспектів, що були протестовані, - точність визначення схожості книг, відсутність дублювання та адекватність рекомендацій з урахуванням вже прочитаних книг. Результати тестування показали, що система ефективно визначає схожість книг та надає релевантні рекомендації, що відповідають індивідуальним інтересам користувачів. Тестування користувачами додатково підтвердило, що система здатна з високою точністю підбирати книги відповідно до їхніх переваг, забезпечуючи задоволеність користувачів обраними рекомендаціями.

Висновки до розділу 3

Розробка програмної реалізації системи рекомендацій книжкового фонду в інтелектуальній системі обліку передбачає вибір між мовами програмування Python та Ruby, кожна з яких має свої переваги та особливості. З урахуванням потреби в швидкій розробці, Ruby на фреймворку Ruby on Rails є оптимальним вибором, завдяки його простоті, безпеці та масштабованості. Для фронтенду обрані React.js та Tailwind CSS, що забезпечують гнучкість та стильний дизайн. У якості середовища розробки вибрано RubyMine, що пропонує глибоку інтеграцію з Ruby on Rails. Структура фронтенду складається з компонентів як BookList, BookDetails, SearchBar, FilterBar, та Recommendations, що ефективно взаємодіють між собою. Бекенд структурується на основі моделей, зв'язаних із моделлю книг, демонструючи чітку організацію та взаємодію класів у контексті Rails застосунку. Такий підхід до розробки гарантує не лише технічну ефективність, але й високу якість користувацького інтерфейсу, сприяючи кращому задоволенню потреб користувачів.

Розробка структури тестової бази даних для системи рекомендацій книжкового фонду передбачає детальне планування та організацію даних у кількох ключових таблицях: книг, жанрів та користувачів. Таблиця книг містить інформацію про назву книги, ID жанру, автора, дату публікації, ISBN код, кількість сторінок, рейтинг та теги. Така структура дозволяє зберігати важливу інформацію про книги, яка використовується для аналізу та рекомендацій. Таблиця користувачів включає поля для пароля, імені користувача, віку та ідентифікаторів прочитаних книг, що забезпечує персоналізацію рекомендацій. Таблиця жанрів, яка розділяє жанри на групи (людина і відносини, природа і світ, історія та культура) та включає коефіцієнти для кожного жанру, сприяє більш точному відбору книг за інтересами користувачів. Структура тестової бази даних є ключовою для

забезпечення ефективності та точності системи рекомендацій, оскільки вона враховує різноманіття літературних жанрів і персоналізує підхід до кожного користувача.

У процесі реалізації методу для генерації рекомендованих книг застосовується послідовний підхід, починаючи з визначення жанрової групи обраної книги, що дозволяє звужити вибірку для більш точного відбору рекомендацій. Метод використовує алгоритм визначення схожості за тегами книг, вилучаючи зайві сполучні слова для чистоти аналізу. Подальша перевірка на схожість виконується з урахуванням уникнення повторень та врахуванням мінімальної кількості спільних тегів. Для кожної книги формуються вектори, включаючи ваговий коефіцієнт жанру та рейтингу книги, що дозволяє обчислити точність схожості. На завершення, відбувається сортування книг за ступенем схожості та рейтингом, а також здійснюється перевірка на наявність книги у списку прочитаних користувачем. В результаті формується список з десяти найбільш рекомендованих книг, які відповідають інтересам та перевагам користувача, що сприяє підвищенню ефективності та точності рекомендацій системи.

Процес тестування системи рекомендацій книг передбачав використання комбінованого підходу, що включав аналіз даних та тестування реальними користувачами. Це дозволило оцінити функціональність системи та її здатність адекватно реагувати на індивідуальні запити та вподобання користувачів. Тестування на основі аналізу даних зосереджувалось на точності алгоритмів визначення схожості між книгами, виявлення дублів, перевірки наявності вже прочитаних книг серед рекомендацій та відповідності підібраних книг із реальними схожими творами. Додатково, було здійснено тестування користувачів, під час якого учасники дослідження взаємодіяли з системою, оцінюючи точність та релевантність запропонованих книг. Вони мали

можливість налаштувати свої вподобання, додати книги до списку прочитаних та отримати персоналізовані рекомендації. В результаті тестування користувачами було підтверджено високу точність системи у підборі відповідних книг, свідчачи про її ефективність та корисність у виборі літератури, що відповідає індивідуальним запитам користувачів.

ВИСНОВКИ

Сучасне бібліотекознавство активно впроваджує інноваційні інтелектуальні системи обліку книжкових фондів, які перевершують традиційні методи завдяки своїй автоматизації, ефективності та здатності до глибокого аналізу даних. Використання принципів штучного інтелекту та машинного навчання дозволяє не тільки оптимізувати процеси ідентифікації та каталогізації книг, але й розробляти персоналізовані рекомендації, аналізувати тенденції читання та прогнозувати попит. Хоча ці системи забезпечують підвищену ефективність та розширений функціонал, вони також вимагають значних інвестицій та залучення кваліфікованих спеціалістів. Існуючі системи обліку, такі як SirsiDynix, Voyager і Koha, надають широкий спектр функцій та інтеграцію з інтелектуальними рекомендаційними системами, але вибір конкретного рішення залежить від різних факторів, у тому числі від розміру та типу бібліотеки. Розробка ефективних рекомендаційних систем у бібліотечному середовищі вимагає ретельного відбору важливих критеріїв, таких як релевантність та доступність, з використанням різноманітних даних, включаючи демографічні характеристики та історію використання користувачами.

При розробці системи рекомендацій ключовим є вибір адекватних методів і технік, що забезпечують створення доречних та точних рекомендацій. Такі системи використовують різні алгоритми, включаючи колаборативну фільтрацію, фільтрацію, базовану на вмісті, аналіз взаємодій та інші методи машинного навчання, щоб надати персоналізовані пропозиції користувачам. Важливо обрати такий метод, який найкраще відповідає цілям системи та доступним даним. Наприклад, колаборативна фільтрація відмінно підходить для аналізу взаємодій між користувачами та об'єктами для створення рекомендацій заснованих на цих

взаємодіях, тоді як фільтрація, орієнтована на зміст, фокусується на аналізі характеристик об'єктів. Ефективне планування рекомендаційної системи вимагає детального вивчення цільової аудиторії, наявних даних та цілей установи, для якої система створюється. Урахування цих факторів допоможе створити систему, що забезпечить користувачів якісними та відповідними рекомендаціями, тим самим підвищуючи їх задоволеність та активність використання системи.

При оцінці методів кластерного аналізу документів, особливо в бібліотечних системах, виявляється висока ефективність нечіткого кластерного аналізу з використанням косинусної відстані. Такий підхід дозволяє агрегувати елементи, такі як книги, засновуючи їх групування на їх взаємній подібності. Косинусна відстань виявляється особливо корисною при аналізі великих обсягів текстових даних. Процес кластеризації включає перетворення даних у векторний формат, визначення косинусної подібності, створення матриці відстаней та застосування методів ієрархічної кластеризації або K-Means. Нечітка кластеризація надає гнучкість, дозволяючи об'єктам належати до декількох груп з різною вагою, що відображає переплетення тем у книжкових колекціях бібліотек. Вибір конкретного методу кластеризації залежить від цілей проекту, наявних даних та особливостей документів, які підлягають класифікації.

Метод розробки рекомендацій до книжкового фонду, який інтегрує колаборативну фільтрацію і нечітку кластеризацію з косинусною відстанню за допомогою алгоритму K-Means, представляє собою всебічний і дієвий підхід до формування індивідуалізованих рекомендацій. Цей метод охоплює збір різноманітних даних про книги і читацькі переваги, розрахунок взаємної подібності книг за косинусною відстанню, а також використання додаткових фільтрів, що базуються на контенті, рейтингах і інших характеристиках. Розробка рекомендацій, що враховує ці аспекти,

забезпечує точність, своєчасність та відповідність пропозицій до потреб користувачів. Постійне оновлення інформаційної бази і оптимізація алгоритмів на підставі відгуків користувачів сприяють неперервному вдосконаленню системи рекомендацій, підвищуючи її ефективність та корисність у контексті бібліотечних послуг.

Проектування програмного забезпечення для системи рекомендацій книг в рамках інтелектуальної системи обліку вимагає обрання між мовами програмування Python і Ruby. З огляду на потребу в швидкому розробленні, Ruby, зокрема на фреймворку Ruby on Rails, стає переважним вибором через його простоту у використанні, безпеку і масштабованість. Для розробки фронтенду вибрані технології React.js та Tailwind CSS, які забезпечують гнучкість та стильний дизайн. Щодо середовища розробки, RubyMine, яке пропонує інтеграцію з Ruby on Rails, вибрано за його функціональність та зручність. Фронтенд структурований з використанням компонентів, таких як BookList, BookDetails, SearchBar, FilterBar та Recommendations, що забезпечує ефективну взаємодію. На бекенді використовуються моделі, пов'язані з моделлю книги, що забезпечує чітку організацію та взаємодію класів у рамках Rails додатку, гарантуючи технічну ефективність та високу якість користувацького інтерфейсу.

Створення тестової бази даних для системи рекомендацій книжкового фонду включає ретельне планування та організацію інформації в кількох ключових таблицях: книг, жанрів та користувачів. Таблиця книг зберігає інформацію про назви, жанри, авторів, дати публікації, ISBN коди, кількість сторінок, рейтинги та теги, що дозволяє збирати необхідні дані для аналізу та формування рекомендацій. Таблиця користувачів включає деталі, які допомагають у персоналізації рекомендацій, такі як паролі, імена, віки та історії прочитаних книг. Таблиця жанрів, поділена на категорії та оснащена коефіцієнтами для жанрів, відіграє ключову роль у точному відборі книг, відповідно до

інтересів користувачів. Ця тестова база даних є фундаментальною для ефективності системи рекомендацій, оскільки вона ретельно враховує різноманіття жанрів та індивідуальні потреби кожного користувача.

Процес впровадження методу для генерації рекомендацій книг у бібліотечних системах передбачає декілька кроків, спрямованих на точний відбір літератури. Спочатку відбувається ідентифікація жанрової групи вибраної книги, що дозволяє зосередитися на відповідному сегменті літератури. Для покращення аналізу, система виключає зайві слова з тегів книг, тим самим забезпечуючи точніше визначення їхньої тематики. Ключовою частиною процесу є порівняння книг на основі їх схожості за тегами, з урахуванням важливості уникнення повторень і встановлення мінімального порогу спільних тегів. Кожній книзі присвоюються вектори, що включають критерії, такі як ваговий коефіцієнт жанру та рейтинг. На заключному етапі, книги ранжуються за ступенем схожості та рейтингом, при цьому перевіряється, чи книга вже не була прочитана користувачем. Це дозволяє сформувати список з десяти найрекомендованіших книг, відповідних інтересам користувача, підвищуючи ефективність рекомендацій.

Тестування системи рекомендацій книг здійснюється через поєднання аналізу даних і реального використання користувачами. Цей підхід дозволяє оцінити функціональність системи та її відповідність запитам користувачів. Аналіз даних зосереджується на точності алгоритмів для визначення схожості між книгами, виявленні повторів, перевірці наявності вже прочитаних книг у рекомендаціях та відповідності підібраних книг реальним вподобанням. Тестування з користувачами включає налаштування персональних вподобань, додавання книг до списку прочитаних та отримання індивідуалізованих рекомендацій. Завдяки такому комплексному підходу до тестування, система демонструє високу

точність у підборі книг, що відповідають конкретним запитам користувачів, підтверджуючи її ефективність та корисність.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Su, X., & Wang, X. (2023). Intelligent library system based on matrix decomposition and SSA algorithm. In *Proceedings of the International Conference on Industrial Science and Engineering*.
2. Li, W., Han, Y., Koo, S., & Ho, N. (2015). AuRoSS: An Autonomous Robotic Shelf Scanning system. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*.
3. (2020). The Prospective of Library Spurred by Artificial Intelligence-China Survey. In *Proceedings of the IEEE International Conference on Artificial Intelligence and Computer Applications*.
4. (2021). An Intelligent Book Sorting Management System. In *Proceedings of the IEEE International Conference on Smart Grid and Energy Automation*.
5. Zhang, L., Guo, L., Liu, G., Zhang, X., Liu, L., Li, L., & Li, L. (2022). An RFID and Computer Vision Fusion System for Book Inventory using Mobile Robot. In *Proceedings of the IEEE International Conference on Computer Communications*.
6. Wang, D., Dai, Y., Xu, X., & Wang, W. (2022). Automatic inventory system of librarian books based on a deep learning algorithm with EAST and CRNN. In *Proceedings of the International Conference on Information Science and Machine Learning*.
7. (2022). Design of intelligent book sorting system based on PLC control. In *Proceedings of the IEEE International Conference on Science, Engineering, and Robotics*.
8. (2022). Design of personalized recommendation algorithm for mobile intelligent book management system based on cloud computing technology. In *Proceedings of the IEEE International Conference on Digital Science and Computer Applications*.

9. Chen, C. (2013). An intelligent mobile location-aware book recommendation system that enhances problem-based learning in libraries. *Interactive Learning Environments*.
10. Ji, X., Li, Y., & Liu, X. (2022). Library Intelligent Book Return Robot Design. In *Proceedings of the International Conference on Robotics and Artificial Intelligence*.
11. (2023). Study on Book Recommendation System. In *Proceedings of the IEEE International Conference on Advanced Computing and Technology for Human Advancement*.
12. (2020). Cloud Based Collaborative Filtering Algorithm for Library Book Recommendation System. In *Proceedings of the IEEE International Conference on Smart Systems and Inventive Technology*.
13. (2022). A Hybrid Book Recommendation System for University Library. In *Proceedings of the IEEE International Conference on Computer Science and Electronic Engineering*.
14. (2021). Library Book Intelligent Recommendation System Based on Artificial Intelligence. *Journal of Intelligent & Fuzzy Systems*.
15. Aggarwal C. Recommender Systems: The Textbook / Charu Aggarwal., 2016. – 519 с.
16. Когулько О.С. Використання методів колаборативної фільтрації для роботи рекомендаційної системи / Міжнародна науково-практична конференція «Математичне та імітаційне моделювання систем» (МОДС-2018) – м. Київ., 25-29 червня 2018 р. – С. 83-86.
17. A Comprehensive Overview of Recommender System and Sentiment Analysis [Електронний ресурс] – Режим доступу до ресурсу:
https://www.researchgate.net/publication/354724660_A_Comprehensive_Overview_of_Recommender_System_and_Sentiment_Analysis.

18. B. Sarwar Item-Based Collaborative Filtering Recommendation Algorithms / B. Evaluating Recommender Systems [Електронний ресурс] – Режим доступу: <https://medium.com/nerd-for-tech/evaluating-recommender-systems-590a7b87afa5>
19. Алгоритм нечіткої кластеризації fcm [Електронний ресурс] – Режим доступу: <https://studfile.net/preview/2429595/page:2/>
20. Алгоритми кластеризації [Електронний ресурс] – Режим доступу: <http://surl.li/nwjaf>
21. Аналіз методів перетворення слова в вектор фіксованої довжини для задачі класифікації текстів [Електронний ресурс] – Режим доступу: <http://surl.li/ocfcj>
22. Варіанти використання та сценарії (Use Cases and Scenarios) [Електронний ресурс] – Режим доступу: <https://www.maxzosim.com/use-cases-and-scenarios/>
23. Відстань між об'єктами (кластерами) і міри близькості груп об'єктів [Електронний ресурс] – Режим доступу: https://stud.com.ua/93356/statistika/vidstan_obyektami_klasterami_miri_blyzkosti_grup_obyektiv
24. Гайд з використання DRY principle: розуміння принципу, дублювання коду та створення абстракцій [Електронний ресурс] – Режим доступу: <https://dou.ua/forums/topic/35712/>
25. Інтелектуальні Інтегровані Системи [Електронний ресурс] – Режим доступу: <https://sites.google.com/a/kubg.edu.ua/integrovani-sistemi/rekomendacijni-sistemi/tipi-rekomendacijnih-sistem>
26. Колаборативна фільтрація [Електронний ресурс] – Режим доступу: <http://surl.li/nwoux>
27. Користувацькі сценарії: що це, як і навіщо їх потрібно будувати [Електронний ресурс] – Режим доступу:

28. Метод векторизації слів - word2vec [Електронний ресурс] – Режим доступу: <http://nlpx.net/archives/179>
29. Метричні методи класифікації та регресії. Алгоритм к найближчих сусідів [Електронний ресурс] – Режим доступу: http://www.andriystav.cc.ua/Downloads/MITER/Lecture_06.pdf
30. Обробка даних реальних користувачів для створення рекомендаційної системи. [Електронний ресурс] – Режим доступу: <https://dou.ua/forums/topic/39070/>
31. Загальні рекомендації з підготовки, оформлення, захисту та оцінювання випускних кваліфікаційних робіт здобувачів вищої освіти першого «бакалаврського» і другого «магістерського» рівнів / За ред. доц. М.І. Шинкарика. Тернопіль: ТНЕУ, 2018. 67 с.
32. Комар М.П., Саченко А.О., Васильків Н.М. Методичні рекомендації до виконання кваліфікаційної роботи з освітньо-професійної програми «Комп'ютерні науки» спеціальності 122 «Комп'ютерні науки» за другим (магістерським) рівнем вищої освіти. Тернопіль: ЗУНУ, 2021. 32 с.

ДОДАТОК А
АПРОБАЦІЯ ОТРИМАНИХ РЕЗУЛЬТАТІВ