

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра інформаційних технологій

«До захисту допущено»

В.о. завідувача кафедри

_____ Світлана ВАЩЕНКО

_____ 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня магістр

зі спеціальності 122 «Комп'ютерні науки»,

освітньо-професійної програми «Інформаційні технології проектування»

на тему: «Методи інформаційної технології формування рекомендацій читачу на основі інструментів інтелектуального аналізу даних»

Здобувачки групи ІТ.м-22 Погорілої Катерини Віталіївни

(шифр групи)

(прізвище, ім'я, по батькові)

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

(підпис)

_____ Катерина ПОГОРІЛА _____

(Ім'я та ПРІЗВИЩЕ здобувача)

Керівник доцент кафедри ІТ, к.т.н., доцент Володимир НАГОРНИЙ _____

(посада, науковий ступінь, вчене звання, Ім'я та ПРІЗВИЩЕ)

(підпис)

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра інформаційних технологій
Спеціальність 122 «Комп'ютерні науки»
Освітньо-професійна програма «Інформаційні технології проектування»

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри ІТ

Світлана ВАЩЕНКО

«_____» _____ 2023 р.

ЗАВДАННЯ

на кваліфікаційну роботу магістра студентіві

Погорлій Катерині Віталіївні

(прізвище, ім'я, по батькові)

1 Тема кваліфікаційної роботи Методи інформаційної технології формування рекомендацій читачу на основі інструментів інтелектуального аналізу даних

затверджена наказом по університету від «08» листопада 2023 р. № 1249-VI

2 Термін здачі студентом кваліфікаційної роботи «11» грудня 2023 р.

3 Вхідні дані до кваліфікаційної роботи Дослідити методи інформаційної технології формування рекомендацій читачу, створити та реалізувати інформаційну технологію, вхідними даними є запит користувача на отримання книг відповідно за введеним вподобанням

4 Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити) анотація, вступ, аналіз предметної області, постановка задачі та методи дослідження, моделювання інформаційної технології, реалізація інформаційної технології

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових слайдів презентації) актуальність, мета та задачі проекту, інструменти реалізації, контекстна діаграма у нотації IDEF0, діаграма декомпозиції першого рівня у нотації IDEF0, діаграма варіантів використання, діаграма послідовності, діаграма активності, практична реалізація, демонстрація роботи веб-додатку, висновки

6. Консультанти випускної роботи із зазначенням розділів, що їх стосуються:

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

Дата видачі завдання _____.

Керівник _____
(підпис)

Завдання прийняв до виконання _____
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Дослідження предметної області	06.11.23 – 11.11.23	
2	Формування мети і задач	12.11.23 – 14.11.23	
3	Аналіз проблем реалізації	15.11.23 – 16.11.23	
4	Виявлення вимог до проекту	17.11.23 – 19.11.23	
5	Планування робіт	20.11.23 – 21.11.23	
6	Вибір засобів реалізації	22.11.23 – 23.11.23	
7	Проектування інформаційної технології	24.11.23 – 26.11.23	
8	Розробка інформаційної технології	27.11.23 – 04.12.23	
9	Оформлення пояснювальної записки	05.12.23 – 11.12.23	

Магістрант _____ Катерина ПОГОРІЛА

Керівник роботи _____ к.т.н., доц. Володимир НАГОРНИЙ

АНОТАЦІЯ

Тема кваліфікаційної роботи магістра «Методи інформаційної технології формування рекомендацій читачу на основі інтелектуального аналізу даних».

Пояснювальна записка складається зі вступу, 4 розділів, висновків, списку використаних джерел із 30 найменувань, 2 додатків. Загальний обсяг роботи – 51 сторінка, у тому числі 36 сторінок основного тексту, 4 сторінки списку використаних джерел, 11 сторінок додатків.

Актуальність роботи полягає в дослідженні та впровадженні інформаційної технології формування рекомендацій. Це необхідно для ефективного вирішення проблеми обробки великого обсягу даних. Дослідження містить порівняння методів систем рекомендацій а також реалізацію трьох з них, таких як метод колаборативної фільтрації, метод контентної фільтрації та гібридний метод. Робота розкриває переваги кожного методу, а також створює основу для розвитку та оптимізації систем рекомендацій.

Мета роботи: розробити інформаційну технологію формування персоналізованих рекомендацій читачу з використанням методів та інструментів інтелектуального аналізу даних.

Ключові слова: інтелектуальний аналіз даних, колаборативна фільтрація, контентна фільтрація, гібридна фільтрація, рекомендаційна система, Python.

ЗМІСТ

ВСТУП.....	5
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	7
1.1 Актуальність проблеми.....	7
1.2 Методи рекомендаційних систем	8
2 ПОСТАНОВКА ЗАДАЧІ ТА МЕТОДИ ДОСЛІДЖЕННЯ.....	17
2.1 Мета та задачі дослідження.....	17
2.2 Вибір методів дослідження	18
3 МОДЕЛЮВАННЯ/ПРОЕКТУВАННЯ ОБ’ЄКТА ДОСЛІДЖЕННЯ.....	22
3.1 Структурно-функціональне моделювання.....	22
3.2 Моделювання варіантів використання рекомендаційної системи	23
3.3 Діаграми послідовності та активності.....	25
4 РЕАЛІЗАЦІЯ ВЕБ-ДОДАТКУ	27
4.1 Структура проекту.....	27
4.2 Розробка веб-додатку	28
4.3 Приклад використання веб-додатку	32
ВИСНОВКИ.....	34
СПИСОК ЛІТЕРАТУРИ.....	35
ДОДАТОК А – ПЛАНУВАННЯ РОБІТ	39
ДОДАТОК Б – ЛІСТИНГ ПРОГРАМНОГО КОДУ	45

ВСТУП

Зі стрімким розвитком технологій та широкими можливостями застосування інформаційних технологій, Інтернет поступово став важливим джерелом доступу людей до інформації. Щодня у світі з'являються сотні мільйонів одиниць мережевої інформації, і люди поступово переходять від епохи інформаційного перевантаження до епохи інформаційного дефіциту. [1]

Інформаційна плутанина часто призводить до того, що люди не можуть знайти корисні дані під час пошуку інформації. Крім того, це призводить до того, що людей приваблюють нерелевантні та безладні дані. Зменшення інформаційного перевантаження та вирішення інформаційної плутанини стало актуальною темою для досліджень. В результаті цього з'явилися такі технології, як класифікаційний каталог, пошукова система та рекомендаційна система. [2]

Перевантаження великою кількістю даних може також стати перешкодою для читача під час пошуку необхідної літератури. Використання інтелектуального аналізу даних для формування рекомендації забезпечує ефективний і якісний відбір книжкових рекомендацій.

Об'єкт дослідження: формування персоналізованих рекомендацій читачу на основі інструментів інтелектуального аналізу даних.

Предмет дослідження: методи формування персоналізованих рекомендацій читачу на основі інструментів інтелектуального аналізу даних.

Мета: розробити інформаційну технологію формування персоналізованих рекомендацій читачу з використанням методів та інструментів інтелектуального аналізу даних.

Основні задачі:

- Провести аналіз предметної області та визначити актуальність роботи.
- Визначити функціональні вимоги.

- Провести моделювання бізнес-процесів та проектування інформаційної технології.
- Реалізувати інформаційну технологію у вигляді веб-додатку.

Практичне значення: розробка та порівняння різних методів рекомендацій для книг сприяє покращенню використання систем та методів, забезпечуючи читачам персоналізовані та цілеспрямовані рекомендації, що підвищує якість отриманих рекомендацій та загальну задоволеність користувача.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Актуальність проблеми

Інтелектуальний аналіз даних - це процес, який використовує різноманітні інструменти аналізу даних для виявлення закономірностей і взаємозв'язків у даних, які, в свою чергу, можуть бути використані для прогнозування. [3]

Рекомендаційні системи були розроблені як інтелектуальні алгоритми, які можуть генерувати результати у вигляді рекомендацій для користувачів. Вони зменшують накладні витрати, пов'язані з вибором найкращого серед великої кількості.[4]

Персоналізовані рекомендаційні технології це засіб, що допомагає користувачу швидко зрозуміти в якій інформації він буде зацікавлений швидше за все. Він може забезпечити різних користувачів персоналізованими послугами, що задовільнить саме їх потреби та інтереси. Для цього не потрібно мати явні і точні вимоги користувача, наприклад як під час використання пошукової системи. Модель будується на основі аналізу історії поведінки користувача та іншої корисної інформації, а потім рекомендує користувачам інформацію, яка найкраще відповідає їхнім інтересам та потребам, на основі моделі. [1]

Проблема накопичення інформації впливатиме на ефективність використання часу та якість рекомендацій, оскільки вона може спричинити людську похибку. Хороша система - це система, яка може спростити роботу і може працювати більш ефективно, щоб підтримувати прискорення роботи, не створюючи проблем.[5]

Сьогодні все більше компаній застосовують рекомендовані технології до інтернет-продуктів. Приклади такого застосування зустрічаються нам кожного дня. Наприклад, домашня сторінка популярної нині платформи короткого відео - це стрічка, що складається з рекомендованих моделей, яка ефективно збільшує тривалість перебування і частоту користувачів.[2] А рекомендаційна система на основі алгоритму спільної фільтрації та правил асоціацій, розроблена Amazon, забезпечує близько 30% продажів. [6]

1.2 Методи рекомендаційних систем

Кожна рекомендаційна система складається з двох сутностей, одна з яких - користувач, а інша - товар. Користувачем може бути будь-який покупець або споживач будь-якого товару чи послуги, який отримує рекомендації. Вхідними даними для алгоритму рекомендацій може бути база даних користувачів і товарів, а вихідними - рекомендації.[4]

Повна рекомендаційна система містить три основні елементи: ресурс користувача, ресурс товару та алгоритм рекомендацій. У моделі користувача аналізуються інтереси споживача, аналогічно, у моделі товару аналізуються характеристики товару. Потім характеристики споживача зіставляються з характеристиками товару, щоб оцінити, які товари рекомендувати за допомогою алгоритму рекомендацій. Продуктивність цього алгоритму впливає на продуктивність всієї системи[8]

Існує багато методів, які використовуються для надання рекомендацій. Вони класифікуються на основі різних джерел знань. Джерела знань включають характеристики користувачів, такі як вік і стать, характеристики товарів/послуг, такі як ключові слова, жанр і дані про вподобання користувачів, такі як рейтинг, історія покупок тощо. Ці дані про вподобання користувачів створюють профіль користувача, який відіграє важливу роль у рекомендаціях. Існуючі методи рекомендацій використовують ці характеристики користувачів і товарів для рекомендацій, але їх недостатньо для надання ефективних рекомендацій. Профілювання користувачів та вдосконалення існуючих методів є головним викликом сьогодення. [9]

Рекомендаційна система фільтрує дані за допомогою багатьох алгоритмів і рекомендує користувачам найбільш підходящі елементи. Вона починає роботу з фіксації попередніх записів покупця і на основі цього рекомендує товари, які користувачі, ймовірно, будуть купувати. Фільтрація - це один з методів відбору ознак,

в якому невелика частина даних береться як набір і використовується для подальшого аналізу.[10]

Найбільш відомими методами є наступні: рекомендація на основі асоціативних правил, колаборативна фільтрація, фільтрація на основі контенту, соціальна фільтрація та гібридна рекомендація. [1]

1.2.1 Рекомендація на основі асоціативних правил

Для реалізації цього методу потрібно встановити відповідні зв'язки між предметами та моделями інтересів користувачів, використовуючи різні правила асоціацій, а потім спрогнозувати інтереси користувачів на основі цих зв'язків, надавши рекомендації. Переваги цього методу полягають в простоті та легкості в реалізації, а також в універсальності у різних галузях та гарній ефективності при використанні у реальному часі. Основна проблема це те, що щойно доданий об'єкт важко знайти за правилами в системі, і його не можна рекомендувати через відсутність даних про поведінку користувачів щодо нього (так звана проблема холодного старту).[1]

1.2.2 Колаборативна фільтрація

Суть цього методу в знаходженні релевантності між елементами або користувачами, аналізуючи поведінку користувачів, а потім відповідно рекомендації користувачам. Це найперший запропонований і найпоширеніший метод у рекомендаційних системах. Його основна ідея - груповий досвід. Зазвичай його теж можна розділити на два методи: колаборативна фільтрація на основі користувачів і колаборативна фільтрація на основі об'єктів.[1]

1.2.2.1.Колаборативна фільтрація на основі користувачів

Цей метод спочатку знаходить набір користувачів зі схожими інтересами з цільовим користувачем відповідно до даних про поведінку користувачів, а потім відфільтровує ті елементи, яким надають перевагу схожі користувачі, але які не переглядаються цільовим користувачем у всіх наборах елементів, і нарешті рекомендує ці елементи цільовому користувачеві.

Основною перевагою є те, що точність є високою у випадку повного та багатого набору даних без вилучення вмісту елементів. Але натомість цей метод не може вирішити проблему холодного старту. А ще зі збільшенням кількості користувачів у системі цей метод потребує підтримки величезної матриці схожості користувачів, що збільшує рекомендований час розрахунку та знижує ефективність системи.[1]

1.2.2.2. Колаборативна фільтрація на основі об'єктів

Цей метод виявляє схожість у товарах, а потім рекомендує схожі товари на основі існуючого вибору користувачів. Метод рекомендацій колабораційної фільтрації на основі товарів не вимагає врахування історичної поведінки нових користувачів, коли вони стикаються з новими користувачами. Як тільки нові користувачі взаємодіють з одним товаром, їм можна рекомендувати схожі на нього товари. Також нова поведінка користувача призведе до змін у результатах рекомендацій у реальному часі. Але обчислення та оновлення таблиці досить ресурсоємне та неефективне. [1]

Наприклад, щоб вирішити проблему розрідженості підходу колаборативної фільтрації, можна збудувати модель ключових слів, яка враховує як записи про тиражі книг, так і дані про ключові слова книг. [11]

К-найближчий сусід - це алгоритм машинного навчання, який використовується для пошуку схожих користувачів серед кластерів з багатьох користувачів за їхніми вподобаннями, оцінками або відгуками, залишеними кожним конкретним користувачем, і робить прогнози відповідно до найкоротшої відстані між користувачами. Алгоритм К-найближчого сусіда вимірює відстань для визначення

"близькості" екземплярів і видає результат. Колаборативна фільтрація рекомендує елементи на основі обчислення схожих елементів користувача, який використовує алгоритм сусідства (K-найближчих сусідів). У техніці колаборативної фільтрації використовується матриця оцінок, щоб згрупувати користувачів зі схожими інтересами.[12]

Існує два методи збору даних для колаборативної фільтрації: явний і неявний. Неявний метод отримує дані на основі поведінки або історії користувача, хоча його точність дещо нижча, ніж явного методу, коли клієнта запитують безпосередньо, але він має перевагу, оскільки може відображати його мінливі тенденції.[13]

В останні роки колаборативні рекомендаційні системи застосовуються в електронних бібліотеках і стають основною темою досліджень у цій галузі. У електронній бібліотеці вподобання користувачів можуть змінюватися за короткий час зі зміною напрямів їхніх досліджень та вимог, тому для вивчення користувацького середовища впроваджується технологія колаборативної фільтрації, заснована на онтології.[14]

1.2.3 Фільтрація на основі контенту

Цей метод видобуває та аналізує контент об'єктів рекомендацій, отримує інтереси користувачів на основі їхньої історичної поведінки та рекомендує користувачам об'єкти, які найкраще відповідають їхнім моделям інтересів у контенті. Хоча він не потребує даних інших користувачів, він все одно може точно відображати інтереси. Він вирішує проблеми холодного старту та розрідженості, які присутні в колаборативній фільтрації. Також присутні певні обмеження, наприклад деякі правила між різними об'єктами не можуть бути вивчені інструментами машинного навчання, або деякі особливості вмісту об'єктів дуже важко виділити. Також страждає від проблеми пропонування дуже схожих рекомендацій користувачам.[1]

Фільтрація на основі вмісту не може з'ясувати якість товару. Для подолання цієї проблеми використовуються системи колаборативної фільтрації.[4]

Оскільки алгоритми фільтрації на основі контенту не обов'язково вимагають оцінок користувачів для побудови свого представлення продукту, рекомендації особливо корисні для наборів даних, де багато продуктів мають відносно невелику кількість оцінок або взагалі не мають оцінок. [15]

1.2.4 Соціальна фільтрація

Перевага цього методу полягає в тому, що він не вимагає взаємодії між користувачами та об'єктами, але при цьому може надавати рекомендації для нових користувачів або нових об'єктів та може вирішити проблему холодного старту. Цей метод використовує інформацію з соціальних мереж користувачів для аналізу їхніх уподобань, а потім вибірково рекомендує їм улюблені товари їхніх друзів. Але недоліком цього методу є те, що в соціальних мережах все ще існує нестача даних.[1]

1.2.5 Рекомендації на основі даних

Замість того, щоб використовувати рейтинги користувачів, цей підхід вивчає поведінкові правила або моделі з історії збережених даних і надає рекомендації на основі вивчених правил. [3]

1.2.6 Демографічний метод рекомендацій

Цей метод використовує лише інформацію про користувачів. Ця методика знаходить взаємозв'язок між користувачами на основі їх демографічного профілю. Рекомендуються товари, яким віддають перевагу користувачі зі схожим демографічним профілем. Демографічний метод страждає від проблеми "холодного старту" для нового

товару, оскільки новому товару не надавав перевагу жоден користувач з тим самим демографічним профілем. Конфіденційність є основною проблемою при зборі демографічної інформації про користувачів. Це пов'язано з двома причинами:

- Важко отримати релевантну рекомендацію, коли людина шукає товар для іншої людини, яка належить до іншого демографічного профілю.
- Користувачі неохоче надають свою особисту інформацію в Інтернеті.[9]

1.2.7 Гібридна рекомендація

Суть методу полягає в тому, щоб обирати різні методи рекомендацій та використовувати відповідні стратегії для їх змішування відповідно до визначених сценаріїв застосування. В основному для реалізації використовують такі стратегії змішування: зважений синтез, перемикання вперед-назад, комбінація ознак, каскад, гібрид метарівня, доповнення ознак тощо, щоб інтегрувати різні методи рекомендацій.[1] Нарешті, гібридні підходи поєднують підходи на основі контенту та спільної роботи, які мають взаємодоповнюючі сильні та слабкі сторони, що дає кращі результати. [15]

Деякі гібридні підходи до рекомендацій:

- Окреме виконання процедур та об'єднання результатів.
- Використання деяких принципів фільтрації контенту з використанням колаборативної фільтрації.
- Використання деяких принципів колаборативної фільтрації у рекомендації контент-фільтрації
- Використання в рекомендації як контент-фільтрації, так і спільної фільтрації[8]

Системи фільтрації на основі контенту рекомендують користувачеві елементи на основі його минулих уподобань. На відміну від них, системи колаборативної фільтрації ґрунтують свої рекомендації для конкретного користувача на минулих вподобаннях інших користувачів, які поділяють його смаки. [3]

Загальні проблеми, притаманні всім рекомендаційним системам це розрідженість, холодний старт і конфіденційність користувачів. [1] Також багато методів ігнорують широко поширений консенсус про те, що зашумлені дані можуть вводити в оману алгоритми рекомендацій. [16]

Використання кластерів для зменшення кількості елементів і користувачів було запропоновано для вирішення проблеми великих обчислень, однак це знизить якість рекомендацій. Іншими словами, якщо метод порівнюватиме користувача з невеликою вибіркою, подібність не буде точною. Крім того, розбиття товарів на проміжки обмежить рекомендації конкретними типами товарів.[17]

Поєднуючи інформацію в онлайн-відгуках користувачів з їхніми основними персональними даними та поведінковими характеристиками в Інтернеті, можна краще вирішити проблеми розрізнених даних про користувачів та "холодного старту" системи в процесі рекомендацій і отримати більш точні персоналізовані рекомендації.[18]

З метою покращення ефекту рекомендацій все більше дослідників починають вивчати модель рекомендацій на основі графів знань. Цей метод рекомендацій не тільки інтегрує зовнішню багату семантичну інформацію сутностей елементів, але також вводить інформацію про асоціації між елементами, що може ефективно полегшити проблему розрідженості даних і холодного старту.[2] Графи знань мають природні переваги в розширенні інформації про об'єкти та зміцненні зв'язків між об'єктами. Це може забезпечити потужну і багату довідкову базу для рекомендаційних систем. Спочатку дослідники безпосередньо застосували атрибути сутностей у графах знань до алгоритму рекомендацій. Потім вони розробили рекомендації на основі графів знань, які допомагали рекомендаціям, тобто впровадили навчання на основі представлення графів знань. Цей метод може забезпечити глибші та довші асоціації між об'єктами. [2]

З розвитком комп'ютерних технологій, технологій штучного інтелекту та інших галузей технологія глибокого навчання широко використовується в розпізнаванні мови, обробці зображень тощо. розпізнавання мови, обробки зображень тощо. Ця технологія дозволяє не тільки точно отримувати взаємозв'язок між користувачами та об'єктами у

різноманітних даних, але також може перетворювати абстрактні коди на високорівневу інформацію про дані. Тому все більше і більше дослідників проводять дослідження моделей мереж глибокого навчання, що застосовуються дослідження мережевих моделей глибокого навчання, застосованих до рекомендаційних систем. [19]

Результати порівняння найпопулярніших методів рекомендаційних систем розміщені в таблиці 1.1.

Таблиця 1.1. – Порівняльна характеристика найпопулярніших методів рекомендаційних систем

	Колаборативна фільтрація	Контентна фільтрація	Асоціативні правила	Гібридні рекомендації
Персоналізованість	+	-	-	+
Гнучкість	+	+	+	+
Ефективність в нових ситуаціях	+	+	+	+
Вирішення проблеми холодного старту	-	-	-	+-
Незалежність від оцінок	-	-	-	+-
Ефективність з великими обсягами даних	-	-	-	-

Джерело: побудовано автором

Вибір методів рекомендацій залежить від конкретних характеристик системи, користувачів та доступної інформації. Наразі не існує ідеального методу рекомендацій, але в цьому випадку варто відкинути метод асоціативних правил з двох причин:

- Книжковий каталог може бути дуже великим, і асоціативні правила можуть генерувати велику кількість правил, що ускладнює їхню обробку та управління.
- Асоціативні правила можуть бути занадто специфічними і орієнтованими на конкретні покупки, що буде обмежувати варіативність рекомендацій.

Враховуючи комплексність інтересів користувачів у книжковому сегменті, колаборативна, контентна та гібридна рекомендації можуть надати більш влучні та персоналізовані рекомендації для читачів.

2 ПОСТАНОВКА ЗАДАЧІ ТА МЕТОДИ ДОСЛІДЖЕННЯ

2.1 Мета та задачі дослідження

Метою дослідження є створення інформаційної технології формування персоналізованих рекомендацій читачу з використанням методів та інструментів інтелектуального аналізу даних.

Основними задачами проекту є:

- Проведення аналізу предметної області та визначення актуальності роботи. Це допоможе визначити тенденції та нагальні проблеми в області персоналізованих рекомендацій для читачів.
- Визначення функціональних вимог потрібно для визначення того, як технологія буде взаємодіяти з користувачами та обробляти введені ними дані.
- Проведення моделювання бізнес-процесів та проектування інформаційної технології допоможе визначити послідовність дій під час надання рекомендацій.
- Реалізація інформаційної технології у вигляді веб-додатку. Це забезпечить можливість користувачеві зручно отримувати рекомендації книг.

Розроблена інформаційна технологія має обробляти вхідні дані користувача і формувати рекомендацію, відображаючи її у веб-інтерфейсі.

Вимоги до веб-додатку:

- Можливість введення вхідних даних користувача.
- Здатність надавати рекомендації на основі колаборативного методу.
- Можливість генерації рекомендацій, використовуючи контентний метод.
- Реалізація гібридного методу для надання рекомендацій, який комбінує колаборативний та контентний підходи.

У додатку А представлено планування робіт з розробки інформаційної технології у вигляді деталізації проекту методом SMART, WBS-діаграми, OBS-діаграми, діаграми Ганта та матриці ризиків.

2.2 Вибір методів дослідження

Для цієї роботи буде використано три найпопулярніших методи рекомендаційних систем, що використовуються для рекомендацій книг з метою порівняння, тому немає необхідності обирати найдоцільніший. Але варто зазначити, що розробник може вибрати найкращий алгоритм, враховуючи структурні обмеження, такі як тип, своєчасність і надійність даних про доступність, допустимий обсяг пам'яті та процесорних ресурсів. [20]

Надання рекомендацій на основі контентної фільтрації має певну послідовність дій: обробка даних, векторизація, обчислення косинусу подібності і власне надання рекомендацій. В першому пункті забезпечується очищення даних та заповнення пропущених даних. Це класичний початок реалізації будь-якої рекомендаційної системи.

Для процесу векторизації було використано статистичний показник TF-IDF, що представлений формулою (2.1):

$$W_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right) \quad (2.1)$$

де $tf_{i,j}$ - це кількість разів, коли слово i з'являється в наборі ключових слів товару j ;

N - загальна кількість об'єктів;

df_i - кількість об'єктів, які містять слово i . [27]

Косинусна подібність вимірює кут між двома векторами, спроектованими в багатовимірному просторі. Розрахунок представлений формулою (2.2). Він порівнює два документи в нормалізованому масштабі. Косинусна схожість - це один із способів обчислення схожості тексту. Це допоможе визначити, наскільки два текстових документи близькі один до одного з точки зору їхнього контексту або значення. [27]

$$\text{Cos}\theta = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\| \|\vec{b}\|} \quad (2.2)$$

де $\vec{a} \cdot \vec{b}$ це добуток двох векторів;

$\|\vec{a}\|$ – це евклідова норма вектору.

Для колаборативної фільтрації потрібно визначити схожість між елементами, аналізуючи матрицю оцінювання користувачем елементів, щоб знайти набір сусідніх елементів, найбільш схожих на рекомендований користувачем набір. Потім ми сортуємо список результатів, що цікавлять користувача, за цими позиціями відповідно до можливої оцінки, яку поставив користувач. Для визначення схожості теж використовується косинусна подібність, що розраховується за формулою (2.2). Після обчислення схожості між елементами алгоритм колаборативної фільтрації на основі елементів повинен знайти набір найближчих сусідів цільового елемента. У цій множині сусідів для цільового елемента генерується множина найближчих сусідів, яка впорядкована за спаданням відповідно до розміру схожості. Останнім етапом відбувається розрахунок прогнозу користувача, щодо товару шляхом підсумовування оцінок користувачів для схожих товарів за формулою (2.3). [26]

$$P_{u,i} = \frac{\sum_{\text{all similar items}, N} (S_{i,N} \cdot R_{u,N})}{\sum_{\text{all similar items}, N} (|S_{i,N}|)} \quad (2.2)$$

Поєднання двох методів рекомендацій задля реалізації гібридного методу рекомендацій було досягнуто за допомогою методу зважених сум. Колаборативна та контентна фільтрації були реалізовані незалежно один від одного. Остаточна схожість була зваженою сумою схожостей, отриманих в результаті колаборативної фільтрації та рекомендацій на основі вмісту. [26]

Спочатку більшість рекомендацій оцінювали та ранжували за їхньою прогностичною силою - здатністю точно передбачати вибір користувача. Однак зараз широко поширена думка, що точні прогнози мають вирішальне значення.[20] Також задоволеність користувачів і похідні від неї, такі як випадковість, різноманітність і довіра, все частіше розглядаються як важливі.[21]

Точність достатньо легко виміряти за умови, якщо користувач має змогу оцінити рекомендацію. Для цього можна використати обчислення середньоквадратичної або абсолютної помилки за формулами (2.4 - 2.5):

$$RMSE = \sqrt{\frac{1}{|\tau|} \sum_{(u,i) \in \tau} (\hat{r}_{ui} - r_{ui})^2}, \quad (2.4)$$

$$MAE = \frac{1}{|\tau|} \sum_{(u,i) \in \tau} |\hat{r}_{ui} - r_{ui}|, \quad (2.5)$$

де \hat{r}_{ui} – прогнозовані рейтинги для набору даних τ пар користувач-об'єкт (u, i) ;

r_{ui} – справжні рейтинги. [22]

Довіра користувача до рекомендацій зазвичай вимірюється шляхом проведення онлайн тестування щодо їх досвіду користування рекомендаційною системою. Також пов'язаним фактором є кількість постійних користувачів, оскільки якщо немає довіри до системи, то люди не повертаються до користування нею. [23, с. 285]

Рекомендації повинні бути різноманітними, тому важливо додавати до списку рекомендацій об'єкти з низькою ймовірністю вибору, щоб забезпечити новизну

інформації для користувача. В найпростішому варіанті вона визначається за формулою (2.6):

$$novelty(i) = -\log_2 p(i) \quad (2.6)$$

де $p(i)$ – це ймовірність потрапляння елемента до списку рекомендацій. [24]

Наразі найкращими метриками, які можна застосувати для оцінки впровадженої рекомендаційної системи є точність та новизна, тож на них і потрібно спиратися під час реалізації та порівняння методів рекомендаційних систем.

3 МОДЕЛЮВАННЯ/ПРОЕКТУВАННЯ ОБ'ЄКТА ДОСЛІДЖЕННЯ

3.1 Структурно-функціональне моделювання

Структурно-функціональне моделювання – це методологія, що використовується для розуміння структури та функцій системи. Це допомагає виокремити основні функції та визначити їх зв'язок. [25]

Контекстна діаграма – це графічне зображення системи та її зовнішнього середовища. Використовується для виділення основних компонентів системи та їх взаємодії з оточенням.[25] Контекстна IDEF0 діаграма для інформаційної технології формування персоналізованих рекомендацій книжок зображена на рисунку. 3. 1.



Рисунок 3.1 - Контекстна діаграма в нотації IDEF

Джерело: побудовано автором

Входи моделі: інформація про книгу, вміст книги, вподобання користувача.

Виходи: сформовані рекомендації книг.

Контролі: метод надання рекомендацій, швидкість виконання, набір даних.

Механізми: користувач, програмне забезпечення, рекомендаційна система.

Декомпозиція на функціональні блоки:

- Вибір методу рекомендаційної системи;
- Формулювання запиту користувача;
- Формування рекомендацій книги;
- Відображення рекомендованої книги.

Також разом з контекстною діаграмою часто створюють діаграму декомпозиції, що має декілька переваг, таких як: зменшення складності системи чи завдання, полегшення аналізу та розуміння кожного компоненту. Діаграма декомпозиції для рекомендованої системи книг представлена на рисунку. 3.2.

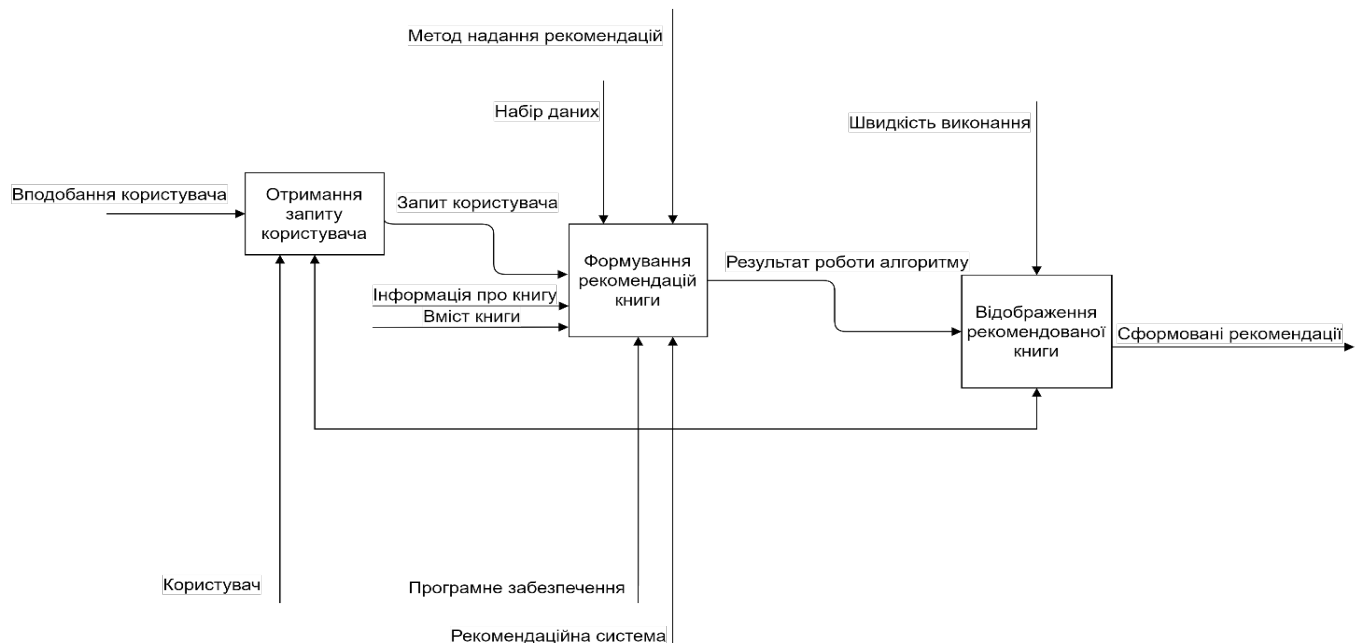


Рисунок 3.2 - Діаграма декомпозиції
Джерело: побудовано автором

3.2 Моделювання варіантів використання рекомендаційної системи

Наглядним представленням функцій проекту є розробка Use case діаграми[25], що зображена на рисунку. 3.3. Опис акторів наведено в таблиці 3.1, опис варіантів використання в таблиці 3.2.

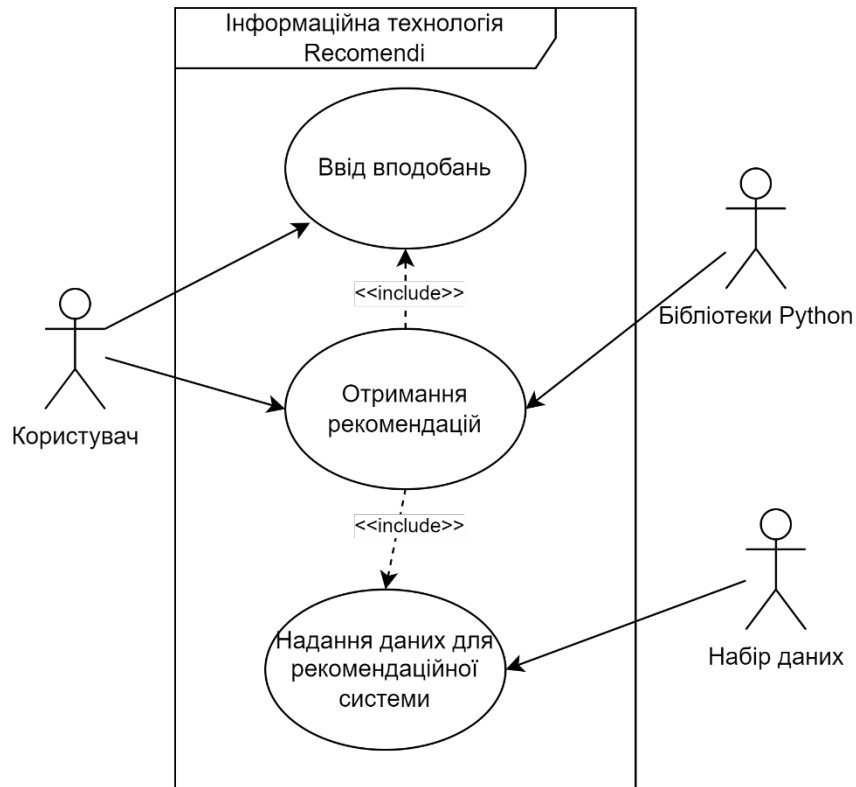


Рисунок 3.3 - Use case діаграма
Джерело: побудовано автором

Таблиця 3.1. – Опис акторів

Актор	Опис
Користувач	Користувач рекомендаційної системи має змогу отримати рекомендацію після вводу власних вподобань
Набір даних	На основі набору даних рекомендаційна система формує рекомендації для користувача
Бібліотеки Python	За допомогою бібліотек Python (Pandas, Sklearn) відбувається зчитування даних та обраховуються дані для формування рекомендацій

Джерело: побудовано автором

Таблиця 3.2. – Опис варіантів використання

Варіант використання	Опис
Ввід вподобань	Користувач має змогу ввести власні вподобання для отримання рекомендацій
Надання даних для рекомендаційної системи	Набір даних надає інформацію для подальшого використання і формування рекомендацій
Отримання рекомендацій	Користувач має змогу переглянути вивід рекомендацій після виконання запиту

Джерело: побудовано автором

3.3 Діаграми послідовності та активності

Діаграма послідовності відноситься до типу діаграм, які відображають як об'єкти в системі взаємодіють між собою.[25] Діаграма послідовності зображена на рисунку.

3.4.

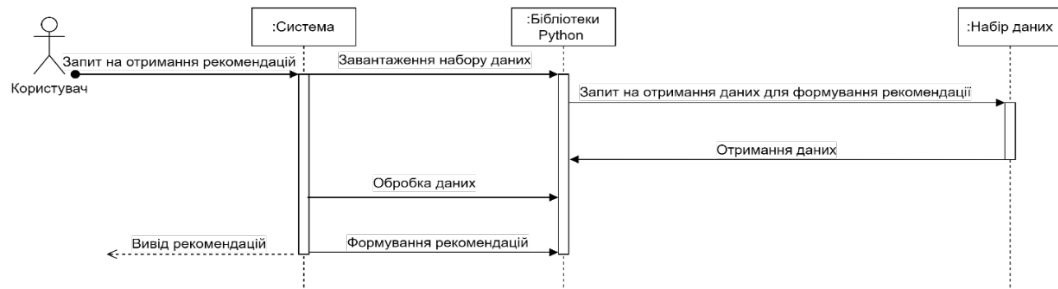


Рисунок 3.4 - Sequence діаграма

Джерело: побудовано автором

Після того, як користувач виконав запит на отримання рекомендацій відбувається декілька послідовних подій. На діаграмі зображена послідовність роботи системи з бібліотеками Python та набором даних. Після отримання запиту система надсилає запит на завантаження набору даних до бібліотеки, в свою чергу вона отримує їх. Маючи всі дані, система починає їх обробку, а потім може вивести рекомендації для користувача.

Діаграма активності це тип UML діаграми, який ілюструє робочий процес системи. Вона представляє виконання дій і рішень від початку до кінця.[25] Діаграма активності зображена на рисунку 3.5.

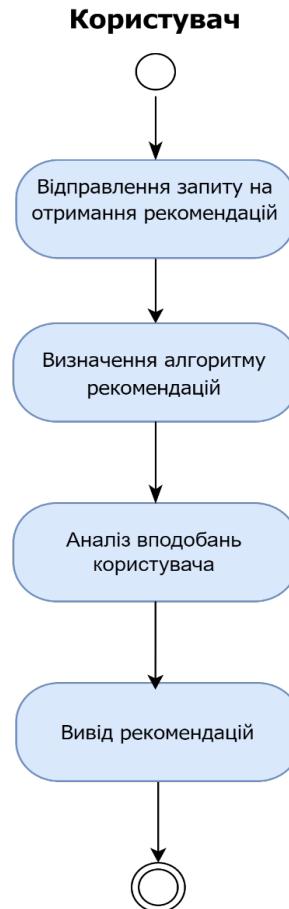


Рисунок 3.5 – Activity діаграма

Джерело: побудовано автором

Оскільки дана інформаційна технологія має досить просту логіку взаємодії, діаграма активності виглядає досить лінійно. Початок діаграми це запит користувача на отримання рекомендацій, далі виконується визначення алгоритму та аналіз вподобань користувача. Закінчується процес виведенням сформованих рекомендацій на екран.

4 РЕАЛІЗАЦІЯ ВЕБ-ДОДАТКУ

4.1 Структура проекту

Програмна реалізація веб-додатку відбувалась у середовищі розробки PyCharm за допомогою веб-фреймворку Django. Стили веб сторінки реалізовані за допомогою Bootstrap.[28]

Шаблон Django - це текстовий документ або рядок Python, розмічений за допомогою мови шаблонів Django. Деякі конструкції розпізнаються та інтерпретуються механізмом шаблонів. Основними з них є змінні та теги. [29]

В реалізованому проекті існує три шаблони, список яких зображений на рисунку 4.1:

Base.html – шаблон, котрий містить базові Html теги та посилання на Bootstrap;

Recommendations.html – шаблон, котрий забезпечує виведення сформованих рекомендацій користувача на сторінку;

User_preferences.html – шаблон, який містить форму, що заповнює користувач перш ніж отримати рекомендації. Заповнення цієї форми забезпечує персоналізованість рекомендацій.

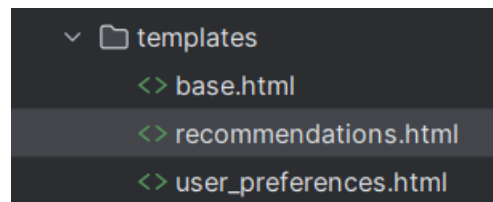


Рисунок 4.1 – Директорія з шаблонами Django

Джерело: побудовано автором (знімок з екрану)

Тека з назвою datasets містить дані, які використовуються для надання рекомендацій. Їх було отримано з сайту Kaggle. [30]

Файли urls.py та views.py типові для Django і забезпечують переходи між шаблонами. Файл forms.py містить поля форми для вводу побажань користувача. В свою чергу recommendations.py – це файл, який містить реалізацію всіх методів

рекомендаційної системи, залучених в реалізації веб-додатку. Дерево проекту зображене на рисунку 4.2.

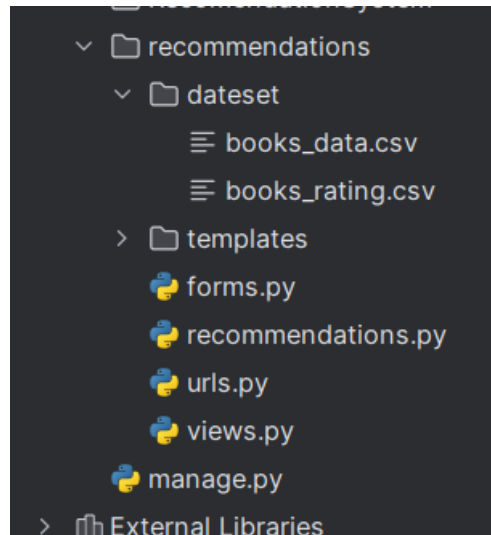


Рисунок 4.2 – Загальна структура проекту

Джерело: побудовано автором (знімок з екрану)

4.2 Розробка веб-додатку

Основою для формування рекомендацій є дані. Для реалізації додатку було взято набір даних «Amazon Books Reviews»[30], структуру цього датасету зображено на рисунку 4.3.

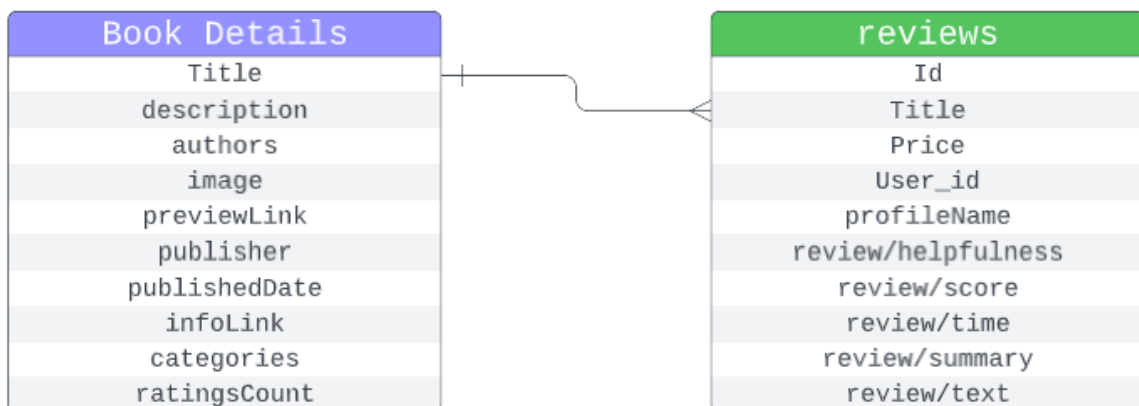


Рисунок 4.3 – Структура датасету

Джерело: [30]

Набір даних складається з двох файлів – деталі про книгу (Book Details) і відгуки (reviews). Для отримання рекомендацій методом контентної фільтрації було б достатньо тільки першого файлу, але для методу колаборативної фільтрації необхідно мати відгуки користувачів для формування. В кодї зчитування даних виконується за допомогою бібліотеки pandas, використовуючи метод read_csv:

```
bookDetails=pd.read_csv('recommendations/dataset/books_data.csv',
encoding='utf-8', sep=',', usecols=['Title', 'authors', 'categories',
'description']).drop_duplicates()
rating=pd.read_csv('recommendations/dataset/books_rating.csv',
encoding='utf-8', sep=',', nrows=80000).drop_duplicates()
mergedData = pd.merge(bookDetails, rating, on='Title').
```

Зчитані дані використовуються для формування рекомендацій. Для формування рекомендацій мною було реалізовано три методи: collaborative, content, hybrid. З особливостей втілення алгоритму можна зазначити обрахування косинусної подібності за допомогою бібліотеки sklearn.metrics.pairwise:

```
cosine_similarities=linear_kernel(user_vector, tfidf_matrix).flatten().
```

А також обрахування статистичного показника TF-IDF за допомогою sklearn.feature_extraction.text :

```
tfidf_vectorizer = TfidfVectorizer(stop_words='english')
tfidf_matrix = tfidf_vectorizer.fit_transform(merged_data['features'])
```

Оскільки веб-додаток не збирає інформацію про користувача, який надсилає запит, то тестування можливе тільки на основі даних з датасету. Для обрахування оцінок було взято трьох користувачів і сформовано приблизний запит на основі їх відгуків. Розмір рекомендацій для обчислення був обмежений до 10 елементів. Точність в цьому разі відповідає за відсоток співпадінь рекомендацій з вже прочитаними користувачем книгами і розраховується за формулою (4.1). Новизна навпаки відображає відсоток нових для користувача книг і представлена формулою (4.2). Результати обрахунків наведено в таблицях 4.1 – 4.3.

$$precision = \frac{|H \cap R|}{|R|} \times 100 \quad (4.1)$$

$$novelty = \left(1 - \frac{|H \cap R|}{|R|}\right) \times 100 \quad (4.2)$$

де H – це книги, що присутні в історії відгуків користувача;

R – рекомендовані книги.

За результатами запиту «object-oriented programming» для користувача з ID «A3F7XLGCTTUMWP» можна зробити висновок, що жодна з рекомендованих книг ще не була прочитана користувачем. Тому новизна рекомендації становить 100%. Даний результат спричинено наявністю великої кількості літератури на дану тематику в заданому датасеті та невеликою кількістю відгуків від читача, що значно розширює кількість варіантів.

Таблиця 4.1. – Результат оцінки рекомендацій для користувача «A3F7XLGCTTUMWP»

	Колаборативний метод	Контентний метод	Гібридний метод
Точність	0%	0%	0%
Новизна	100%	100%	100%

Джерело: побудовано автором

Наступним запитом є «The Rabbi's Cat» для користувача з ID «A3F7XLGCTTUMWP». Отриманий результат є протилежним до першого, оскільки запитом є конкретна назва книги, прочитаної читачем.

Таблиця 4.2. – Результат оцінки рекомендацій для користувача «A3F7XLGCTTUMWP»

	Колаборативний метод	Контентний метод	Гібридний метод
Точність	100%	100%	100%
Новизна	0%	0%	0%

Джерело: побудовано автором

Останнім запитом є «Medicine» для користувача з ID «A23OLHL5RPQKLP». Колаборативний метод надав повністю нові для користувача рекомендації, в той час як контентний та гібридний методи надали 10% вже знайомих користувачу книг. Тобто користувач вже був зацікавленим і залишав відгук про 1 з 10 книг, запропонованих контентним та гібридним методом.

Таблиця 4.3. – Результат оцінки рекомендацій для користувача «A23OLHL5RPQKLP»

	Колаборативний метод	Контентний метод	Гібридний метод
Точність	0%	10%	10%
Новизна	100%	90%	90%

Джерело: побудовано автором

За результатами оцінювання можна зробити висновок, що для більш загальних запитів користувач отримує рекомендації з високим рівнем новизни, що забезпечує наявність широкого вибору для користувача. В свою чергу ті запити, які мають в собі конкретну назву, забезпечують бажану рекомендацію.

Для покращення якості оцінювання можна провести оптимізацію параметрів рекомендаційних систем. Цього можна досягти шляхом емпіричного порівняння рекомендацій за умови зміни параметрів методу обчислення TF-IDF, алгоритмів ранжування та підходу до обробки тексту (наприклад, застосування стемінгу). Також для підвищення точності можна збільшити кількість даних, що використовується для створення рекомендаційної системи, оптимізувати та розширити метрики.

В праці [20] зазначається, що найкращим показником ефективності рекомендацій залишається відгук користувача, для впровадження ефективної і якісної рекомендаційної системи потрібно враховувати додавання систему відгуків користувача та постійний збір даних.

4.3 Приклад використання веб-додатку

Практична реалізація веб-додатку формування персоналізованих рекомендацій читачу надає користувачеві змогу отримати список рекомендованих книг після запиту з його вподобаннями.

На рисунку 4.4 зображена головна сторінка веб-додатку, котра містить форму введення власних вподобань. Можна вказати текстові вподобання – це може бути конкретна назва книги або ключові слова, перелічені через кому, або крапку з комою. Також є можливість обрати методи рекомендаційної системи, за допомогою яких будуть формуватися рекомендації. За замовчуванням обрана колаборативна фільтрація. Для отримання рекомендацій має бути обраний хоча б один метод. Після натиснення кнопки «Submit» надходить запит з вподобаннями користувача і починається формування рекомендацій.

User Preferences

User Preference

- Collaborative Filtering
- Content Filtering
- Hybrid Filtering

Рисунок 4.4 – Сторінка для надсилання запиту користувачем

Джерело: побудовано автором (знімок з екрану)

Після того, як рекомендації сформовано користувач може побачити список з обраними методами рекомендацій. Кожен метод повертає список з максимум трьох рекомендацій читачеві. На рисунку 4.5 зображено виведення рекомендацій за умови, що користувачем було обрано три методи рекомендаційних систем. В свою чергу на рисунку 4.6 зображено варіант використання, коли користувач вибрав тільки два методи.

Recommendations

Collaborative recommendations:

- A Philadelphia Catholic in King James's Court
- The Rabbi's Cat
- Professional collection catalog

Content recommendations:

- Guide to Owning a Burmese Cat (Popular Cat Library)
- The New Cat Handbook (New Pet Handbooks)
- Searle's Cats

Hybrid recommendations:

- Professional collection catalog
- The Rabbi's Cat
- Searle's Cats

Рисунок 4.5 – Виведення результатів рекомендацій

Джерело: побудовано автором (знімок з екрану)

Recommendations

Collaborative recommendations:

- A Philadelphia Catholic in King James's Court
- The Rabbi's Cat
- Professional collection catalog

Content recommendations:

- Guide to Owning a Burmese Cat (Popular Cat Library)
- The New Cat Handbook (New Pet Handbooks)
- Searle's Cats

Рисунок 4.6 – Виведення результатів рекомендацій за обраним методом

Джерело: побудовано автором (знімок з екрану)

ВИСНОВКИ

В ході виконання кваліфікаційної роботи магістра був проведений детальний аналіз предметної області, зроблено огляд існуючих методів рекомендаційних систем, зокрема їх переваг та недоліків, та визначено проблемні аспекти. Результати аналізу вказують на актуальність теми дослідження зі зростом попиту на персоналізовані рекомендації.

Також було визначено та сформульовано функціональні вимоги до інформаційної технології формування персоналізованих рекомендацій читачу. Це допомогло відобразити бізнес-цілі проекту, що в подальшому вплине на досягнення поставленої мети.

Після формування функціональних вимог було проведено планування робіт, що включає в себе деталізацію мети методом SMART, розробку WBS та OBS діаграм, розробку діаграми Ганта для визначення часових проміжків проекту та матрицю ризиків.

Наступним етапом було проведено структурно-функціональне моделювання та моделювання варіантів використання рекомендаційної системи. Проектування інформаційної технології забезпечити ефективну та логічну структуру під час реалізації.

Розробка була проведена за допомогою мови програмування Python (бібліотеки pandas та sklearn), дані для створення та тестування рекомендаційної системи були взяті з сайту Kaggle. Результати розробки методів рекомендаційних систем були оцінені за допомогою вирахування точності та новизни наданих рекомендацій. Практичною цінністю розробки є реалізація та оцінка методів рекомендаційних систем для формування книжкових рекомендацій читачу.

Результатом кваліфікаційної роботи магістра є інформаційна технологія, що реалізує формування персоналізованих рекомендацій читачу.

СПИСОК ЛІТЕРАТУРИ

1. Li M., Wang L. A Survey on Personalized News Recommendation Technology. *IEEE Access*. 2019. Vol. 7. P. 145861–145879. DOI: <https://doi.org/10.1109/access.2019.2944927> (date of access: 29.11.2023).
2. Research of Personalized Recommendation Technology Based on Knowledge Graphs / X. Yang et al. *Applied Sciences*. 2021. Vol. 11, no. 15. P. 7104. DOI: <https://doi.org/10.3390/app11157104> (date of access: 29.11.2023).
3. Bendakir N., Aïmeur E. Using Association Rules for Course Recommendation.
4. Rajpurkar S., Bhatt D., Malhotra P. Book Recommendation System. *International Journal for Innovative Research in Science & Technology*. 2015. Vol. 11, no. 1. P. 314–316.
5. Hikmatyar M., Ruuhwan. Book Recommendation System Development Using User-Based Collaborative Filtering. *Journal of Physics: Conference Series*. 2020. Vol. 1477. P. 032024. DOI: <https://doi.org/10.1088/1742-6596/1477/3/032024> (date of access: 29.11.2023).
6. Xiao L., Lu Q., Guo F. Mobile Personalized Recommendation Model based on Privacy Concerns and Context Analysis for the Sustainable Development of M-commerce. *Sustainability*. 2020. Vol. 12, no. 7. P. 3036. DOI: <https://doi.org/10.3390/su12073036> (date of access: 29.11.2023).
7. A group recommendation system for online communities / J. K. Kim et al. *International Journal of Information Management*. 2010. Vol. 30, no. 3. P. 212–219. DOI: <https://doi.org/10.1016/j.ijinfomgt.2009.09.006> (date of access: 29.11.2023).
8. Rana A., Deeba K. Online Book Recommendation System using Collaborative Filtering (With Jaccard Similarity). *Journal of Physics: Conference Series*. 2019. Vol. 1362. P. 012130. DOI: <https://doi.org/10.1088/1742-6596/1362/1/012130> (date of access: 29.11.2023).

9. Chandak M., Girase S., Mukhopadhyay D. Introducing Hybrid Technique for Optimization of Book Recommender System. *Procedia Computer Science*. 2015. Vol. 45. P. 23–31. DOI: <https://doi.org/10.1016/j.procs.2015.03.075> (date of access: 29.11.2023).
10. Product Recommendation System a Comprehensive Review / J. Sharma et al. *IOP Conference Series: Materials Science and Engineering*. 2021. Vol. 1022. P. 012021. DOI: <https://doi.org/10.1088/1757-899x/1022/1/012021> (date of access: 29.11.2023).
11. Enhancing the Performance of Library Book Recommendation System by Employing the Probabilistic-Keyword Model on a Collaborative Filtering Approach / N. Ifada et al. *Procedia Computer Science*. 2019. Vol. 157. P. 345–352. DOI: <https://doi.org/10.1016/j.procs.2019.08.176> (date of access: 29.11.2023).
12. Neighborhood-based approach of collaborative filtering techniques for book recommendation system / N. Sivaramakrishnan et al. *International Journal of Pure and Applied Mathematics*. 2019. Vol. 119, no. 12.
13. Jooa J., Bangb S., Parka G. Implementation of a Recommendation System Using Association Rules and Collaborative Filtering. *Procedia Computer Science*. 2016. Vol. 91. P. 944–952. DOI: <https://doi.org/10.1016/j.procs.2016.07.115> (date of access: 29.11.2023).
14. Personalized Book Recommendation Based on Ontology and Collaborative Filtering Algorithm / L. Cui et al. *The Open Cybernetics and Systemics Journal*. 2014. Vol. 8, no. 1. P. 632–637. DOI: <https://doi.org/10.2174/1874110x01408010632> (date of access: 29.11.2023).
15. Book Recommendation System / J. Cho et al. 2016.
16. Exploring Clustering-Based Reinforcement Learning for Personalized Book Recommendation in Digital Library / X. Wang et al. *Information*. 2021. Vol. 12, no. 5. P. 198. DOI: <https://doi.org/10.3390/info12050198> (date of access: 29.11.2023).

17. Alluhaidan A. Recommender System Using Collaborative Filtering Algorithm. *ScholarWorks@GVSU*. URL: <https://scholarworks.gvsu.edu/cistechlib/155/> (date of access: 29.11.2023).
18. Chen S., Lv X., Gou J. Personalized Recommendation Model: An Online Comment Sentiment Based Analysis. *International Journal of Computers Communications & Control*. 2020. Vol. 15, no. 1. DOI: <https://doi.org/10.15837/ijccc.2020.1.3764> (date of access: 29.11.2023).
19. Hou D. Personalized Book Recommendation Algorithm for University Library Based on Deep Learning Models. *Journal of Sensors*. 2022. Vol. 2022. P. 1–6. DOI: <https://doi.org/10.1155/2022/3087623> (date of access: 29.11.2023).
20. Shani G., Gunawardana A. Evaluating Recommendation Systems. *Recommender Systems Handbook*. Boston, MA, 2010. P. 257–297. DOI: https://doi.org/10.1007/978-0-387-85820-3_8 (date of access: 29.11.2023).
21. Tintarev N., Masthoff J. A Survey of Explanations in Recommender Systems. *2007 IEEE 23rd International Conference on Data Engineering Workshop*, Istanbul, 17 April 2007. 2007.
22. Meleshko Y. Методи оцінки якості роботи рекомендаційних систем. Системи управління, навігації та зв'язку. Збірник наукових праць. 2018. Т. 5, № 51. С. 92–97. DOI: <https://doi.org/10.26906/sunz.2018.5.092> (дата звернення: 05.12.2023).
23. *Recommender systems handbook* / ed. by F. Ricci et al. Boston, MA : Springer US, 2011. DOI: <https://doi.org/10.1007/978-0-387-85820-3> (date of access: 05.12.2023).
24. Vargas S. Novelty and diversity enhancement and evaluation in recommender systems and information retrieval. *SIGIR '14: the 37th international ACM SIGIR conference on research and development in information retrieval*, Gold Coast Queensland Australia. New York, NY, USA, 2014. DOI: <https://doi.org/10.1145/2600428.2610382> (date of access: 05.12.2023).
25. Larman C. *Applying UML and patterns: an introduction to object-oriented analysis and design and iterative development* (3rd edition). 3rd ed. Prentice Hall PTR, 2004. 704 p.

26. Hybrid recommendation system combining collaborative filtering and content-based recommendation with keyword extraction / S. Zhang et al. *Applied and computational engineering*. 2023. Vol. 2, no. 1. P. 927–939. DOI: <https://doi.org/10.54254/2755-2721/2/20220579> (date of access: 12.12.2023).
27. A content-based movie recommendation system / Y. Yuan et al. *Theoretical and natural science*. 2023. Vol. 2, no. 1. P. 56–66. DOI: <https://doi.org/10.54254/2753-8818/2/20220152> (date of access: 12.12.2023).
28. Bootstrap. Bootstrap · The most popular HTML, CSS, and JS library in the world. URL: <https://getbootstrap.com/> (date of access: 09.12.2023).
29. Django. Django Project. URL: <https://docs.djangoproject.com/en/5.0/topics/templates/> (date of access: 09.12.2023).
30. Amazon books reviews. Kaggle: Your Machine Learning and Data Science Community. URL: <https://www.kaggle.com/datasets/mohamedbakheta/amazon-books-reviews> (date of access: 09.12.2023).

ДОДАТОК А – ПЛАНУВАННЯ РОБІТ

Деталізація мети проекту методом SMART. Продуктом дипломного проекту є веб сторінка з системою рекомендації, що надає персоналізовані рекомендації книг.

Результати деталізації методом SMART розміщені у таблиці А.1.

Таблиця А.1 – Деталізація мети методом SMART

Specific (конкретна)	Розробити систему рекомендацій, яка надає персоналізовані рекомендації книг для користувачів, використовуючи три методи рекомендацій - колаборативний, контентний та гібридний.
Measurable (вимірювана)	Розрахувати коефіцієнти точності рекомендацій для кожного методу, зменшити відсоток невдалих рекомендацій.
Achievable (досяжна)	Використання існуючі API та методи для рекомендацій.
Relevant (реалістична)	Врахування інтересів та поведінки користувачів у розробці алгоритмів рекомендацій.
Time-framed (обмежена у часі)	На розробку проекту необхідно 12 тижнів, за умови роботи по 5 годин на день.

Джерело: побудовано автором

Планування змісту структури робіт IT-проекту (WBS). WBS – це ієрархічна декомпозиція загального обсягу робіт, для успішної реалізації проекту. WBS структура допомагає організувати і визначити загальний обсяг проекту, розбиваючи його на менші

частини. Це допомагає розподіляти ресурси та відстежувати прогрес. Структура WBS розміщена на рисунку А.1.

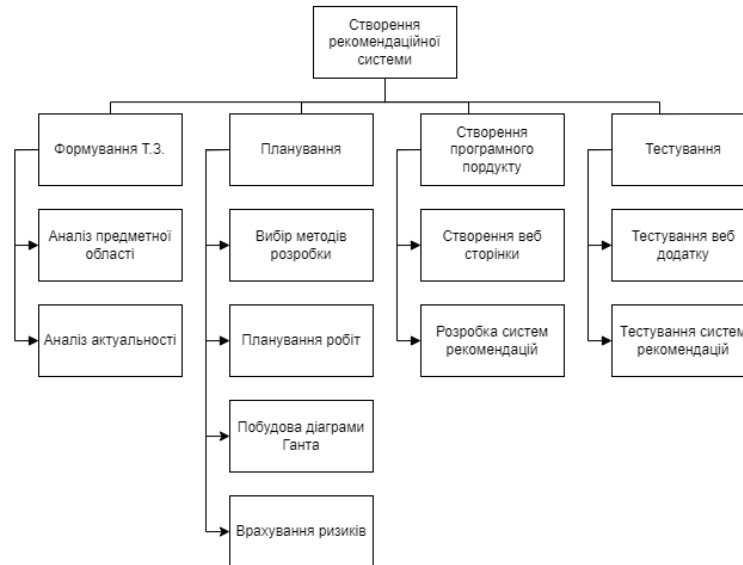


Рисунок А.1 – Структура WBS
Джерело: побудовано автором

Організаційна структура проекту (OBS). OBS – це ієрархічне представлення організації проекту. Це допомагає визначити структуру проекту та відповідальних осіб, які виконують той чи інший етап імплементації проекту. Це допомагає визначити обов’язки і ієрархію учасників проекту. Структура OBS відображена на рисунку А.2.

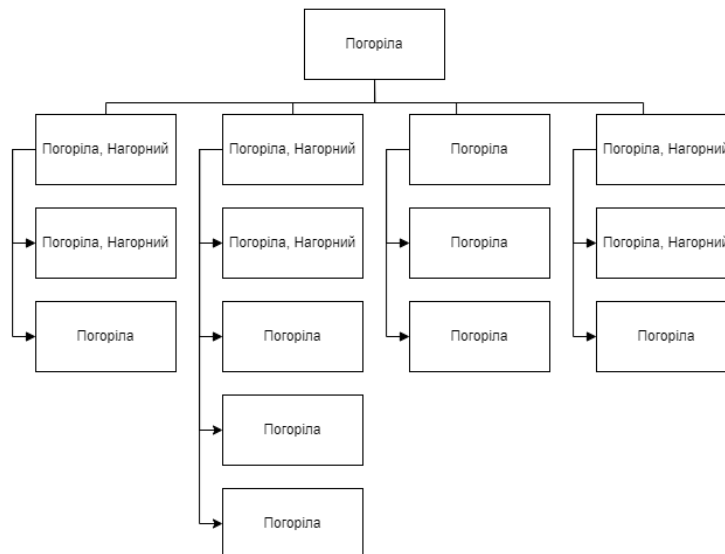


Рисунок А.2 – Структура OBS
Джерело: побудовано автором

Побудова календарного графіка виконання ІТ-проекту. Для побудови календарного графіку виконання проекту розповсюдженою метрикою є Діаграма Ганта. Це гістограма, яка відображає завдання проекту у відповідних часових рамках. Діаграма розміщена на рисунку А.3.

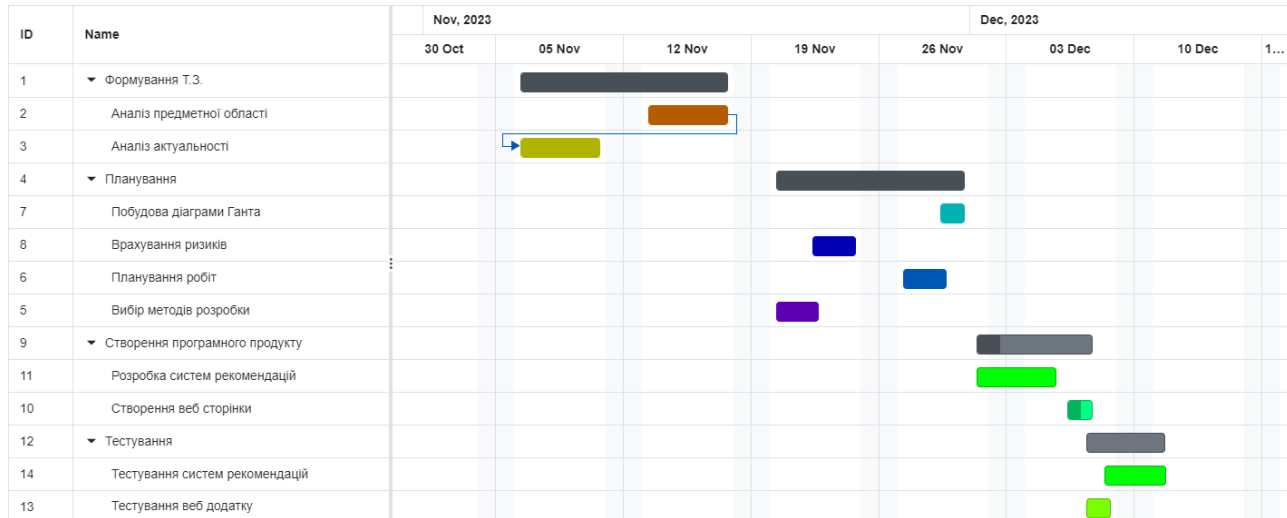


Рисунок А.3 – Діаграма Ганта

Джерело: побудовано автором

Управління ризиками. До основних ризиків розробки рекомендаційної системи книг є:

- неправильно обраний алгоритм рекомендацій;
- збільшення обсягу даних;
- недостатня кількість даних;
- зміна вподобань читачів;
- збільшення очікувань щодо якості рекомендацій;
- відсутність кваліфікованого програміста;
- зміна строків виконання роботи;
- зміна функціональних вимог;
- зростання вимог до проекту.

Оцінка ймовірності виникнення та величини ризиків розміщена в таблиці Б.2.

Таблиця Б.2. Ймовірність виникнення і величина ризику

№	Ризики	Виникнення	Втрати
1	Неправильно обраний алгоритм рекомендацій	3	2
2	Збільшення обсягу даних	2	4
3	Недостатня кількість даних	3	2
4	Зміна вподобань читачів	2	4
5	Збільшення очікувань щодо якості рекомендацій	3	3
6	Відсутність кваліфікованого програміста	2	5
7	Зміна строків виконання роботи	2	4
8	Зміна функціональних вимог	3	3
9	Зростання вимог до проекту	4	3

Джерело: побудовано автором

Аналізуючи ризики за ймовірністю їх виникнення, можемо їх розділити на:

- Ігноровані
 - Відсутні
- Незначні
 - Збільшення обсягу даних
 - Зміна вподобань читачів
 - Відсутність кваліфікованого програміста
 - Зміна строків виконання роботи
- Помірні
 - Неправильно обраний алгоритм рекомендацій
 - Недостатня кількість даних
 - Збільшення очікувань щодо якості рекомендацій
 - Зміна функціональних вимог
- Істотні
 - Зростання вимог до проекту
- Критичні
 - Відсутні

Класифікація ризиків за рівнем впливу:

- Прийнятні
 - Відсутні
- Виправдані
 - Збільшення очікувань щодо якості рекомендацій
 - Зміна функціональних вимог
 - Зростання вимог до проекту
 - Збільшення обсягу даних
 - Зміна вподобань читачів
 - Зміна строків виконання роботи
- Неприпустимі

ДОДАТОК Б – ЛІСТИНГ ПРОГРАМНОГО КОДУ

views.py

```

from django.shortcuts import render
from .forms import UserPreferencesForm
from .recommendations import collaborative, content, hybrid

def user_preferences(request):
    if request.method == 'POST':
        form = UserPreferencesForm(request.POST)

        if form.is_valid():
            userPreference = form.cleaned_data['preference']
            isCollaborative = form.cleaned_data['collaborative']
            isContent = form.cleaned_data['content']
            isHybrid = form.cleaned_data['hybrid']

            recommendations = {}

            if isCollaborative:
                recommendations['Collaborative'] = collaborative(userPreference)[:3]

            if isContent:
                recommendations['Content'] = content(userPreference)[:3]

            if isHybrid:
                recommendations['Hybrid'] = hybrid(userPreference, 0.5, 0.5)[:3]

            return render(request, 'recommendations.html', {'recommendations': recommendations})

        else:
            form = UserPreferencesForm()

    return render(request, 'user_preferences.html', {'form': form})

```

recommendations.py

```

import re

import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import linear_kernel

bookDetails = pd.read_csv('recommendations/dataset/books_data.csv', encoding='utf-8', sep=',',
                           usecols=['Title', 'authors', 'categories', 'description']).drop_duplicates()

rating = pd.read_csv('recommendations/dataset/books_rating.csv', encoding='utf-8', sep=',',

```

```

nrows=80000).drop_duplicates()
mergedData = pd.merge(bookDetails, rating, on='Title')

def collaborative(user_preference):
    mergedData['features'] = mergedData[['Title', 'authors', 'categories', 'review/text']].astype(str).agg(' '.join,
                                                                                                       axis=1)

    user_item_matrix = pd.pivot_table(
        mergedData,
        index='User_id', columns='Title', values='review/score', fill_value=0
    )
    user_preferences = re.split(r'[;,]+', user_preference)
    filtered_books = []

    for term in user_preferences:
        filtered_books.extend(user_item_matrix.loc[:, user_item_matrix.columns.str.contains(term,
case=False)].columns)

    filtered_books = list(set(filtered_books))

    book_avg_ratings = user_item_matrix[filtered_books].mean(axis=0)
    recommendations = book_avg_ratings.sort_values(ascending=False).index

    scores = book_avg_ratings[recommendations]
    max_score = scores.max()
    normalized_scores = scores / max_score

    return list(zip(recommendations[:5], normalized_scores[:5]))

def content(user_preference):
    merged_data = mergedData[['Title', 'description']].drop_duplicates(subset=['Title', 'description'])
    merged_data['features'] = (merged_data['Title'] + ' ' + merged_data['description']).fillna("")
    tfidf_vectorizer = TfidfVectorizer(stop_words='english')
    tfidf_matrix = tfidf_vectorizer.fit_transform(merged_data['features'])

    user_preferences = re.split(r'[;,]+', user_preference)
    matching_indices = set()

    for term in user_preferences:
        matching_indices.update(
            merged_data[merged_data['features'].str.contains(term, case=False, na=False)].index
        )

    user_vector = tfidf_vectorizer.transform([user_preference])
    cosine_similarities = linear_kernel(user_vector, tfidf_matrix).flatten()

    scores = pd.Series(cosine_similarities, index=merged_data.index)
    scores = scores[scores.index.isin(matching_indices)]

```

```

scores.sort_values(ascending=False, inplace=True)

recommended_books = [
    (merged_data.loc[i, 'Title'], scores.loc[i])
    for i in scores.index
]

return recommended_books[:5]

def hybrid(user_preference, weight_collaborative, weight_content):
    collaborative_recommendations = collaborative(user_preference)
    content_recommendations = content(user_preference)

    collaborative_df = pd.DataFrame(collaborative_recommendations, columns=["Title",
"Collaborative_Score"])
    content_df = pd.DataFrame(content_recommendations, columns=["Title", "Content_Score"])
    collaborative_df["Collaborative_Score"] /= collaborative_df["Collaborative_Score"].max()
    content_df["Content_Score"] /= content_df["Content_Score"].max()

    merged_df = pd.merge(collaborative_df, content_df, on="Title", how="outer").fillna(0)

    merged_df["Combined_Score"] = (
        weight_collaborative * merged_df["Collaborative_Score"] +
        weight_content * merged_df["Content_Score"]
    )

    merged_df = merged_df.sort_values(by="Combined_Score", ascending=False)

    return merged_df[["Title", "Combined_Score"]].head(5).values.tolist()

def calculate_precision(actual_items, recommended_items):
    intersection = set(actual_items) & set(recommended_items)
    precision = len(intersection) / len(recommended_items)
    return precision

def calculate_novelty(recommended_items, user_history):
    novelty = 1 - len(set(recommended_items) & set(user_history)) / len(recommended_items)
    return novelty

def test():
    user_preference = "Medicine"
    user_id = "A23OLHL5RPQKLP"

    actual_items = get_actual_items_for_user(user_id)
    collaborative_recs = collaborative(user_preference)

```



```

collaborative_recs_titles = [title for title, score in collaborative_recs]

content_recs = content(user_preference)
content_recs_titles = [title for title, score in content_recs]

precision_collaborative = calculate_precision(actual_items, collaborative_recs_titles)
novelty_collaborative = calculate_novelty(collaborative_recs_titles, actual_items)

precision_content = calculate_precision(actual_items, content_recs_titles)
novelty_content = calculate_novelty(content_recs_titles, actual_items)

print(f"Precision for Collaborative: {precision_collaborative}")
print(f"Novelty for Collaborative: {novelty_collaborative}")

print(f"Precision for Content: {precision_content}")
print(f"Novelty for Content: {novelty_content}")

hybrid_recs = hybrid(user_preference, 0.5, 0.5)
hybrid_recs_titles = [title for title, score in hybrid_recs]

precision_hybrid = calculate_precision(actual_items, hybrid_recs_titles)
novelty_hybrid = calculate_novelty(hybrid_recs_titles, actual_items)

print(f"Precision for Hybrid: {precision_hybrid}")
print(f"Novelty for Hybrid: {novelty_hybrid}")

```

```

def get_actual_items_for_user(user_id):
    user_ratings = rating[rating['User_id'] == user_id]
    actual_items = user_ratings['Title'].tolist()

    return actual_items

```

forms.py

```

from django import forms

```

```

class UserPreferencesForm(forms.Form):
    preference = forms.CharField(label='User preference', max_length=100)
    collaborative = forms.BooleanField(label='Collaborative filtering', initial=True, required=False)
    content = forms.BooleanField(label='Content filtering', initial=False, required=False)
    hybrid = forms.BooleanField(label='Hybrid filtering', initial=False, required=False)

```

base.html

```

<!DOCTYPE html>
<html lang="en">
<head>

```

```

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet"
integrity="sha384-
T3c6CoIi6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwwykc2MPK8M2HN"
crossorigin="anonymous"> <!-- ... (existing content) ... -->
</head>
<body>
  <div class="container">
    {% block content %} {% endblock %}
  </div>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-
C6RzsynM9kWDrMNeT87bh95OGNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL"
crossorigin="anonymous"></script></body>
</html>

```

recommendations.html

```

{% extends 'base.html' %}

{% block content %}
<div class="container mt-5">
  <h2 class="mb-4">Recommendations</h2>
  {% if recommendations %}
    {% for method, method_recommendations in recommendations.items %}
    <h3>{{ method }} recommendations:</h3>
    <ul>
      {% for recommendation in method_recommendations %}
      <li>{{ recommendation }}</li>
      {% endfor %}
    </ul>
    {% endfor %}
  {% else %}
    <p>No recommendations available.</p>
  {% endif %}
</div>
{% endblock %}

```

user_preferences.html

```

{% extends 'base.html' %}

{% block content %}
<div class="container mt-5">
  <h2 class="mb-4">User Preferences</h2>
  <form method="post" class="mb-4">
    {% csrf_token %}
    <div class="mb-3">
      <label for="{{ form.preference.id_for_label }}" class="form-label">User Preference</label>
      {{ form.preference }}
    </div>
  </form>
</div>

```

```

<div class="form-check mb-3">
  {{ form.collaborative }}
  <label class="form-check-label" for="{{ form.collaborative.id_for_label }}">Collaborative
Filtering</label>
</div>
<div class="form-check mb-3">
  {{ form.content }}
  <label class="form-check-label" for="{{ form.content.id_for_label }}">Content Filtering</label>
</div>
<div class="form-check mb-3">
  {{ form.hybrid }}
  <label class="form-check-label" for="{{ form.hybrid.id_for_label }}">Hybrid Filtering</label>
</div>
<button type="submit" class="btn btn-primary">Submit</button>
</form>
</div>
{% endblock %}

```