Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Системотехніки
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

рівень вищої освіти другий (магістерський)

Розробка та дослідження рекомендаційної системи на основі контентної фільтрації
(тема)

Виконав:
студент 2 курсу, групи ІТПм-22-1
Линник О. О.
(прізвище, ініціали)

Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)

Освітня програма Інформаційні технології проектування
(повна назва освітньої програми)

Керівник проф. Іванов В.Г.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ І.В. Гребеннік
(підпис) (прізвище, ініціали)

2024 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук

Кафедра Системотехніки

Рівень вищої освіти другий (магістерський)

Спеціальність 122 Комп'ютерні науки
(код і повна назва)

Тип програми освітньо-професійна

Освітня програма Інформаційні технології проектування
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«___» _____20___ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Линнику Олександру Олеговичу
(прізвище, ім'я, по батькові)

1. Тема роботи: Розробка та дослідження рекомендаційної системи на основі контентної фільтрації
затверджена наказом по університету від 20.11 2023 р. № 1373Ст
2. Термін подання студентом роботи до екзаменаційної комісії: 18.01.2024 р
3.Вихідні дані до роботи: дослідити та розробити рекомендаційну систему на основі контентної фільтрації

4. Перелік питань, що потрібно опрацювати в роботі: Вступ. Опис сучасного стану розвитку рекомендаційних систем. Аналіз застосування досліджуваних методів в існуючих системах Рекомендаційна система на основі контенту Проектування системи рекомендацій на основі контенту Розробка системи рекомендацій. Висновки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій: типи рекомендаційних систем, етапи виконання колаборативної фільтрації, візуальне представлення методу K-NN та правила, візуалізація алгоритму фільтрації на основі контенту, візуальне представлення набору даних, оброблений набір даних, результат виконання фільтрації, ключові слова з описів та рейтинг схожості рекомендованих книг.

6. Консультанти розділів роботи

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на на кваліфікаційну роботу	16.10.2023	Виконано
2	Аналіз предметної області та постановка задачі	18-28.11.2023	Виконано
3	Аналіз методів щодо вирішення поставленої задачі	29-01.12.2023	Виконано
4	Огляд літератури	02-14.12.2023	Виконано
5	Теоретичне дослідження методу вирішення задачі	14-25.12.2023	Виконано
6	Розробка рекомендаційної системи	25-30.12.2023	Виконано
7	Оформлення пояснювальної записки	01-10.01.2024	Виконано
8	Представлення на рецензування	18.01.2024	Виконано

Дата видачі завдання 16.10.2023 р.

Студент _____
(підпис)

Керівник роботи _____ проф. Іванов В.Г.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 61 с., 25 рис., 4 табл., 3 дод., 14 джерел.

РЕКОМЕНДАЦІЙНА СИСТЕМА, K-NN, КОСИНУСНА ПОДІБНІСТЬ, K-НАЙБЛИЖЧИХ-СУСІДІВ, TF-IDF, ФІЛЬТРАЦІЯ КОНТЕНТУ.

Об'єкт дослідження – методи рекомендаційної системи на основі контентної фільтрації.

Предмет дослідження – рекомендаційна система на основі контенту.

Мета роботи – розробка та дослідження рекомендаційної системи на основі контенту. Система призначена для забезпечення користувачам зручного вибору релевантного контенту.

Методи дослідження – розробка рекомендаційної системи на основі контенту, порівняння з іншими рекомендаційними системами та дослідження методів системи.

У роботі представлено порівняння рекомендаційних систем, опис етапів системи та опис методів для кожного етапу, таких як методи вагування та методи для знаходження подібності. Також побудова самої рекомендаційної системи включно з попередньою обробкою даних.

ABSTRACT

Explanatory note: 61 p., 25 fig., 4 tabl., 3 ann., 14 sources.

RECOMMENDATION SYSTEM, K-NN, COSINE SIMILARITY, K-NEAREST NEIGHBORS, TF-IDF, CONTENT FILTERING.

The object of the research is the methods of content-based recommendation systems.

The subject of the research is a content-based recommendation system.

The goal of the work is to develop and investigate a content-based recommendation system. The system aims to provide users with convenient selection of relevant content.

Research methods include developing a content-based recommendation system, comparing it with other recommendation systems, and investigating system methodologies.

The paper presents a comparison of recommendation systems, describes the stages of the system, and outlines methods for each stage, such as weighting methods and similarity finding methods. It also involves building the recommendation system itself, including data preprocessing.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

k-NN – k-Nearest Neighbors – метод k-найближчих сусідів;

Термін – ключове слово у векторному представленні;

Документ – набір ключових слів;

TF-IDF – Term Frequency-Inverse Document Frequency;

НД – навчальні документи;

ТД – тестові документи;

Sklearn – Scikit-learn – бібліотека для машинного навчання.

ВСТУП

Системи рекомендацій — це інтелектуальні інструменти, які забезпечують користувачів індивідуальними рекомендаціями щодо продуктів, контенту або послуг. Основною метою цих систем є підвищення задоволення користувачів і стимулювання їх активності на платформі. Завдяки аналізу попередніх взаємодій та переваг користувачів, системи прагнуть надати контент, який найімовірніше буде їм цікавим. Такі системи знаходять застосування у найрізноманітніших сферах, включно з електронною комерцією, стрімінговими сервісами, соціальними мережами, освітою, медіа бізнесом та багатьма іншими. Вони сприяють поліпшенню користувацького досвіду, збільшують залученість користувачів і час, проведений на платформі, і забезпечують більш персоналізований підхід до представлення контенту та пропозицій.

Суттєвий вплив на розвиток рекомендаційних систем зробило зростання обчислювальних потужностей і застосування алгоритмів машинного навчання, особливо техніки глибокого навчання. Ці методи дозволили обробляти великі обсяги даних і враховувати складні взаємозв'язки, що покращило точність рекомендацій і здатність систем швидко адаптуватися до змін у поведінці користувачів.

Системи рекомендацій можна вважати драйвером індивідуального досвіду користувачів на електронних платформах. Використовуючи багато даних від взаємодії користувачів, таких як історія переглядів, вподобайки, записи покупок і пошукові запити, ці системи формують складне розуміння індивідуальних уподобань, що далі використовується для прогнозування інтересів і запропонування нового контенту, який відповідає особистим уподобанням користувача.

Застосування різноманітних алгоритмів і методик, від колаборативного фільтрування та на основі контенту до більш складних гібридних підходів,

системи рекомендацій захоплюють широкий спектр вподобань користувачів. Ці системи не тільки покращують індивідуальні враження, але й сприяють зростанню бізнесу, підвищуючи конверсію продажів, лояльність клієнтів та персоналізацію маркетингових зусиль.

Неперервне вдосконалення алгоритмів рекомендацій і зростаючий тренд прийняття рішень на основі даних, підкреслюють важливість цих систем. Оскільки приватність даних та етичні аспекти набирають все більшої актуальності, розробка прозорих і справедливих систем рекомендацій, які поважають згоду користувачів і мінімізують упередженість, стає пріоритетним напрямком у цій галузі. Синтез технологій та принципів дизайну, орієнтованих на користувача, визначає напрямок, в якому будуть розвиватись системи рекомендацій, до такого майбутнього, де персоналізований цифровий досвід є не лише зручністю, але і очікуванням.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Опис сучасного стану розвитку рекомендаційних систем

Історія рекомендаційних систем розпочалася в 1990-х роках, коли стали зрозумілі можливості використання комп'ютерів для надання персоналізованих рекомендацій користувачам. Перші системи базувалися на простих алгоритмах фільтрації, що враховували взаємодії користувачів з контентом або товарами.

У 1992 році була представлена система "Tapestry" в Xerox PARC, яка дозволяла користувачам обирати та рекомендувати ресурси, використовуючи систему тегів. Це була система, яка допомагає користувачам електронної пошти протидіяти спам-листам, дана система дозволяє отримувати листи лише від ресурсів, на які були підписані користувачи. В середині 1990-х було багато досліджень, спрямованих на вирішення проблем рекомендацій, та виникли перші комерційні системи, такі як система Amazon, яка використовувала колаборативний підхід для рекомендацій.

На початку 2000-х років виникла проблема інформаційного перенасичення, зокрема у сфері електронної пошти, де користувачі отримували величезну кількість небажаних повідомлень, відомих як спам. Саме тоді з'явилося інтерес до використання рекомендаційних систем для боротьби із спамом. Ефективні рішення були спрямовані на визначення та фільтрацію небажаного контенту, щоб полегшити життя користувачам і зробити їхні поштові скриньки більш управляючими. Перші рекомендаційні системи для електронної пошти в основному використовували евристичні методи, такі як фільтри на основі ключових слів або шаблонів, щоб визначити, чи є повідомлення спамом чи ні. Згодом було впроваджено більш складні технології, такі як байєсівські фільтри та машинне навчання, для автоматичного вивчення

та адаптації до нових типів спаму. Рекомендаційні системи, зокрема ті, які виявляють патерни в користувацькому поведінці та аналізують відгуки на зазначені як спам повідомлення, допомагають автоматизувати індивідуальний фільтр для кожного користувача.

Цей розвиток став важливим кроком у використанні рекомендаційних систем для розв'язання практичних завдань, а не лише для розваг та комерційних цілей. Такий підхід не тільки полегшив життя користувачам електронної пошти, але й встановив основи для подальшого розвитку рекомендаційних систем у різних сферах інтернет-технологій.

На сьогоднішній день рекомендаційні системи є ключовим елементом в інформаційному ландшафті, що обслуговує різноманітні сфери від роздрібної торгівлі та розваг до освіти та досліджень. Основною метою рекомендаційних систем є забезпечення користувачів персоналізованими пропозиціями, що відповідають їхнім уподобанням та потребам. Ваш досвід взаємодії з платформами, такими як YouTube, Netflix, Amazon, або соціальними мережами, є прикладом впровадження рекомендаційних систем у реальному житті.

Останні роки відзначаються значними досягненнями у сфері рекомендаційних систем, зокрема завдяки розвитку глибокого навчання (deep learning) та зростанням обчислювальної потужності. Моделі глибокого навчання дозволяють ефективно аналізувати складні зв'язки та залежності в даних, що покращує точність рекомендацій та здатність систем адаптуватися до змін у користувацькому поведінці.

Однак існують виклики, пов'язані з етикою та конфіденційністю даних у рекомендаційних системах. Збір та обробка великих обсягів особистих даних може викликати обурення користувачів та породжувати питання щодо безпеки. Також важливо враховувати можливість формування "фільтру пузиря", коли користувачі обмежуються інформацією, що відповідає їхнім попереднім

переглядам. Саме через це, деякі системи побудовані так, що інколи пропонують випадкові елементи, які можуть не підходити активному користувачеві.

У майбутньому можна очікувати подальший розвиток рекомендаційних систем через поєднання методів машинного навчання, збагачених даних, та врахування соціокультурних аспектів. Наприклад, рекомендаційні системи можуть стати більш контекстуалізованими, враховуючи не лише особисті вподобання, але й контекст використання та соціальний вплив.

1.2 Аналіз застосування досліджуваних методів в існуючих системах

У сучасних рекомендаційних системах спостерігається значний вплив передових методів і технологій, що визначає їхню високу ефективність та широкий спектр застосування. На рисунку 1.1 представлені типи рекомендаційних систем. Розглянемо наступні типи рекомендаційні системи:

- основані на контенті;
- основані на знанні;
- системи з використанням глибокого навчання;
- факторизаційні машини;
- колаборативна фільтрація;
- гібридні системи.

Розглянемо більш детально, рекомендаційні системи на основі контенту - це підхід до рекомендацій, де система враховує характеристики та описи об'єктів та користувачів. В основі цього методу лежить розуміння контенту та його відповідність вподобанням користувача. Наприклад, якщо користувач позитивно реагує на певний жанр кінофільму, система може рекомендувати інші фільми того ж самого жанру. Такі рекомендаційні системи використовують різноманітні методи для аналізу об'єктів. Вони можуть включати аналіз ключових слів для врахування семантичного змісту, тематичний аналіз для ідентифікації основних

тем, аналіз засобами природної мови для розуміння контексту тексту, аналіз зображень та відео для врахування вмісту медіафайлів, і аналіз метаданих для додаткової інформації про об'єкти. Ці методи допомагають побудувати детальні профілі об'єктів та персоналізувати рекомендації для кожного користувача, забезпечуючи більш точні та цільовані рекомендації. Такий підхід рекомендаційних систем особливо ефективний, коли важко визначити схожість між користувачами на основі їхніх інтересів.

Дані системи є описані в книзі "Recommender Systems" [2]. Системи рекомендацій, засновані на знаннях, особливо корисні для предметів, які не купуються дуже часто. Наприклад, це можуть бути такі речі, як нерухомість, автомобілі, запити на туризм, фінансові послуги чи дорогі товари розкоші. У таких випадках може бути недостатньо рейтингів для процесу рекомендацій. Оскільки ці товари купують рідко і мають різноманітні деталізовані параметри, складно зібрати достатню кількість рейтингів для конкретної реалізації (тобто комбінації параметрів) даного товару. Ця проблема також стикається в контексті проблеми холодного старту, коли для процесу рекомендацій недостатньо рейтингів. Крім того, при роботі з такими товарами природа вподобань споживачів може змінюватись з часом. Наприклад, модель автомобіля може значно змінитись протягом кількох років, внаслідок чого можуть змінитися й вподобання. У інших випадках може бути складно повністю охопити інтереси користувача за допомогою історичних даних, таких як рейтинги. Конкретний товар може мати характеристики, які відображають його різноманітні властивості, і користувач може цікавитися лише товарами з певними властивостями. Наприклад, автомобілі можуть мати кілька марок, моделей, кольорів, опцій двигуна та внутрішнього оздоблення, і інтереси користувачів можуть бути регульовані дуже конкретною комбінацією цих параметрів. Таким чином, у таких випадках область товарів складна за своїми різноманітними властивостями, і складно пов'язати достатню кількість рейтингів із великою

кількістю наявних комбінацій. Цікаво зауважити, що як системи на основі знань, так і системи на основі контенту значно залежать від характеристик предметів. Оскільки системи на основі знань використовують характеристики контенту, вони спадковують деякі недоліки систем на основі контенту. Наприклад, так само, як системи на основі контенту, рекомендації в системах на основі знань часом можуть бути очевидними, оскільки вони не використовують оцінки інших користувачів. Фактично, системи на основі знань іноді розглядають як "близьких родичів" систем на основі контенту. Основна відмінність полягає в тому, що системи на основі контенту вчаться на основі минулої поведінки користувачів, тоді як системи рекомендацій на основі знань рекомендують на основі активного визначення користувачем їх потреб та інтересів. Відмінність систем представлено в таблиці 1.1.

Системи, що використовують глибоке навчання, застосовують нейронні мережі для аналізу великої кількості даних та моделювання складних взаємозв'язків. Це дозволяє системі здійснювати більш точний аналіз користувацьких уподобань та забезпечувати персоналізовані рекомендації на основі глибокого розуміння контенту та користувацьких попередніх взаємодій.

Факторизаційні машини використовують матричні розклади для аналізу та передбачення відгуків користувачів. Матричний розклад використовується для розкладання матриці оцінок користувачів та об'єктів (наприклад, фільмів чи товарів) на дві матриці нижнього рангу. Це дозволяє зберігати інформацію про користувачів та об'єкти у вигляді меншого об'єму даних, сприяючи вирішенню проблеми розрідженості даних та забезпечуючи більш ефективний аналіз. За допомогою цього методу система може враховувати складні взаємодії між користувачами та об'єктами. Наприклад, якщо користувачі здійснюють покупки в інтернет-магазині, факторизаційні машини можуть передбачати їхні вподобання на основі раніше придбаних товарів та взаємодії з іншими користувачами.

Колаборативна фільтрація використовує взаємодію та зворотний зв'язок між користувачами для створення рекомендацій. Якщо два користувачі мають схожі вподобання або взаємодіють із схожими об'єктами, система може рекомендувати об'єкти, які один з них сподобаються, але інший ще не переглядав чи не використовував. Це допомагає рекомендаційній системі знаходити підходящі об'єкти, враховуючи схожість користувачів.

Гібридні системи об'єднують різні методи рекомендацій для забезпечення більш широкого та точного спектру рекомендацій. Це може включати поєднання фільтрації на основі змісту та колаборативної фільтрації. Наприклад, гібридна система може використовувати фільтрацію на основі змісту для рекомендацій фільмів та колаборативну фільтрацію для рекомендацій музики. Такий підхід дозволяє системі бути більш гнучкою та адаптивною до різних сценаріїв використання.

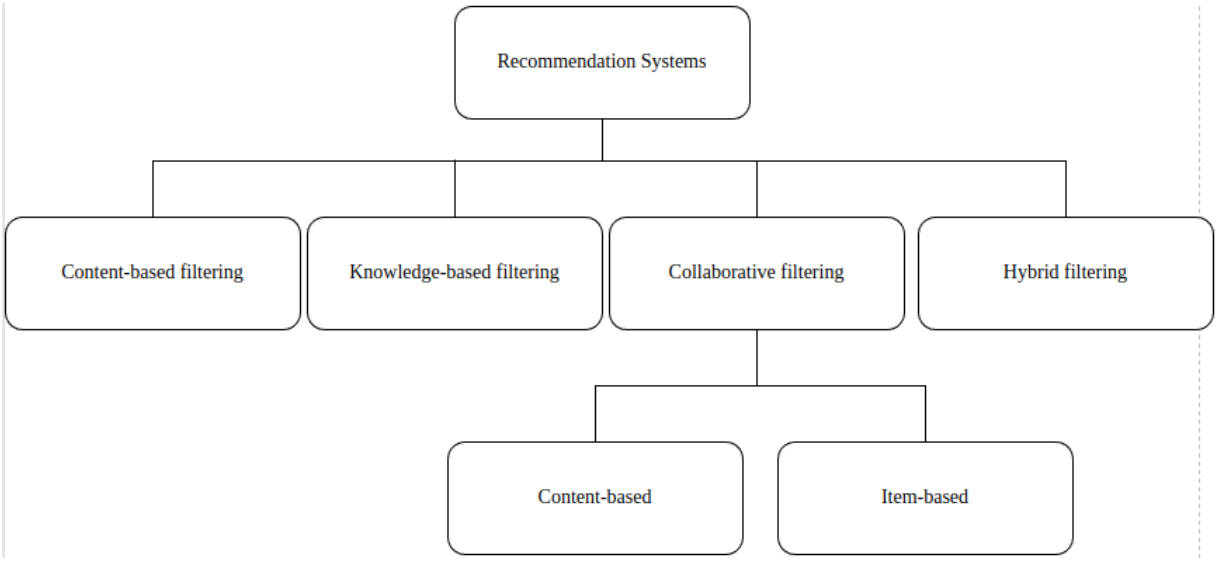

Рисунок 1.1 – Типи рекомендаційних систем

Таблиця 1.1 – Відмінність рекомендаційних систем

Тип рекомендаційної системи	Ціль
Колаборативна фільтрація	Генерація рекомендації на основі колаборативного підходу, які використовують відгуки та дії подібних користувачів або активного користувача.
Основана на контенті	Генерація рекомендації на основі контенту, який активний користувач обрав у попередніх відгуках та діях.
Основана на знанні	Генерація рекомендацій на основі явних вказівок активного користувача щодо виду контенту.

1.2.1 Колаборативна фільтрація

Основна ідея колаборативної фільтрації полягає у використанні взаємодій користувачів з контентом (товарами, фільмами, тощо) для передбачення та рекомендацій нових об'єктів. На рисунку 1.2 представлені етапи даної системи. Процес роботи колаборативної фільтрації виглядає наступним чином:

- збір даних - початково збираються дані про взаємодії користувачів з об'єктами, такі як оцінки, перегляди, покупки, тощо;
- побудова матриці взаємодій - створюється матриця, де кожен рядок відповідає користувачеві, кожен стовпчик - об'єкту, а значення в ячейці - оцінка або взаємодія користувача з об'єктом;

- визначення схожості - використовуються методи для визначення схожості між користувачами або об'єктами. Це може бути косинусна схожість, кореляція, методи машинного навчання тощо;
- прогнозування оцінок - на основі схожості обчислюються прогнози оцінок для користувачів або об'єктів, які ще не мали взаємодій;
- рекомендації - об'єкти з найвищими прогнозованими оцінками рекомендуються користувачам, що допомагає підбирати об'єкти, які можуть їм сподобатися;
- оновлення моделі - після отримання нових даних або взаємодій, модель періодично оновлюється для покращення рекомендацій.

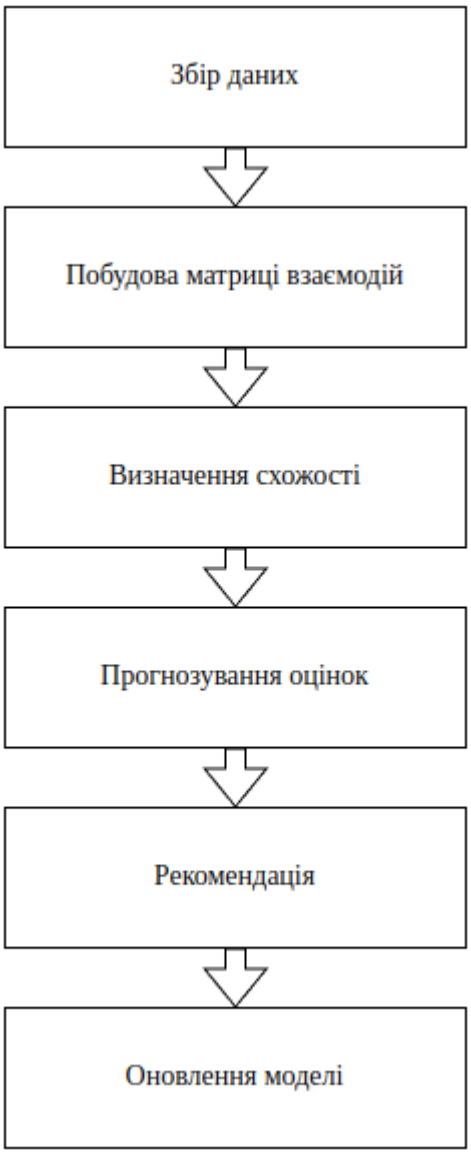

Рисунок 1.2 – Етапи виконання колаборативної фільтрації

1.2.2 Колаборативна фільтрація на основі користувачів

Розглянемо типи колаборативної фільтрації:

- колаборативна фільтрація на основі користувачів;
- колаборативна фільтрація на основі об’єктів(контенту).

Тип колаборативної фільтрації на основі користувачів визначає схожість між користувачами на основі їхніх взаємодій та вподобань. Якщо користувач А подібний до користувача В у своїх виборах та діях, то система рекомендує користувачеві А об'єкти, які сподобалися користувачеві В, і навпаки. Це враховує особисті вподобання та подібність між користувачами. Схожість між користувачами зазвичай визначається за допомогою таких методів, як: косинусна схожість, підрахунок кореляцій, або із застосуванням методів машиного навчання.

Косинусна схожість використовується для визначення схожості між двома користувачами на основі кута між їхніми векторами у просторі предметів (об'єктів, які вони вибирали чи оцінювали). Чим менший кут між векторами, тим більше схожість. Косинусна схожості для користувачів u та v розраховується за формулою (1.1).

$$(u, v) = \frac{\sum_i u_i \times v_i}{\sqrt{\sum_i u_i^2 \times \sum_i v_i^2}} \tag{1.1}$$

де (u, v) - подібність;

u_i та v_i - оцінки, які користувачі u та v поставили об'єктам.

Підрахунок кореляцій вимірює ступінь лінійної залежності між оцінками користувачів. Визначається, наскільки зміни в оцінках одного користувача відповідають змінам іншого. Кореляція розраховується за формулою (1.2).

$$(u, v) = \frac{\sum_i (u_i - \bar{u}) \times (v_i - \bar{v})}{\sqrt{\sum_i ((u_i - \bar{u}))^2 \times \sum_i ((v_i - \bar{v}))^2}} \quad 1.2)$$

де (u, v) - подібність;

u_i та v_i - оцінки, які користувачі u та v поставили об'єктам;

$\bar{u}$ та $\bar{v}$ - середні оцінки користувачів u та v відповідно.

Алгоритми машинного навчання використовують методи навчання з учителем чи без нього для прогнозування вподобань користувачів на основі їхніх попередніх дій у системі. Найпоширенішим є метод k-NN(k-найближчих сусідів).

1.2.3 Колаборативна фільтрація на основі контенту

На відміну від фільтрування на основі користувачів, тип колаборативної фільтрації за контентом визначає схожість між об'єктами (товарами, фільмами і т.д.). Якщо об'єкт А подібний до об'єкта В у тому, які користувачі вибирали їх, то система рекомендує об'єкти, схожі на ті, які вже сподобалися користувачам. Схожість за контентом зазвичай знаходиться за наступними методами: аналіз засобами природної мови, аналіз зображень та відео, аналіз метаданих.

Для текстової інформації в об'єктах (наприклад, описи товарів або відгуки користувачів) застосовується обробка природної мови. Вона включає в себе векторизацію тексту, виявлення тем та семантичний аналіз, що дозволяє враховувати контент і рекомендувати подібні об'єкти за текстовим описом.

Для візуального контенту, такого як зображення або відео, можуть використовуватися методи комп'ютерного зору та обробки зображень. Зокрема,

застосовують алгоритми витягнення характеристик, розпізнавання об'єктів і класифікації, щоб знайти подібні об'єкти.

При аналізі метаданих враховуються додаткові дані про об'єкти, такі як категорії, теги, атрибути, що допомагають у визначенні подібності. Ці дані можуть бути використані для порівняння об'єктів та врахування їхніх характеристик.

1.2.4 Рекомендаційна система на основі контенту

Методи, що ґрунтуються на вмісті, мають кілька переваг у наданні рекомендацій для нових елементів, коли недостатньо даних про оцінки цього елемента. Це через те, що інші елементи з подібними характеристиками можуть бути оцінені активним користувачем. Тому навчена модель зможе використовувати ці оцінки разом з характеристиками елементів для надання рекомендацій навіть тоді, коли немає історії оцінок для цього елемента.

Методи на основі контенту мають такі недоліки:

- у багатьох випадках методи, що ґрунтуються на вмісті, надають очевидні рекомендації через використання ключових слів чи контенту. Наприклад, якщо користувач ніколи не взаємодіяв з елементом із певним набором ключових слів, такий елемент не має шансів на рекомендацію. Це через те, що побудована модель специфічна для конкретного користувача, і не використовується загальне знання спільноти подібних користувачів. Цей феномен спричиняє зменшення різноманітності рекомендованих елементів, що є небажаним;
- хоча методи, що ґрунтуються на вмісті, ефективні у наданні рекомендацій для нових елементів, вони не ефективні у наданні рекомендацій для нових користувачів. Це через те, що модель навчання для цільового користувача потребує використання його історії оцінок. Насправді, для забезпечення

надійних прогнозів без перенавчання зазвичай потрібна велика кількість наявних оцінок для цільового користувача.

Отже, методи, що ґрунтуються на вмісті, мають різні компроміси порівняно з системами колаборативної фільтрації.

Хоча вище описується звичайний підхід до методів рекомендаційних систем заснованованих на контенті, іноді використовується більш широкий підхід. Наприклад, користувачі можуть вказувати відповідні ключові слова у своїх власних профілях. Ці профілі можуть бути порівняні з описами елементів для надання рекомендацій. Такий підхід не використовує оцінок у процесі рекомендацій, тому він корисний при холодному старті.

1.2.5 Проблема холодного старту

Одна з основних проблем у рекомендаційних системах полягає в тому, що початкова кількість доступних оцінок є відносно невеликою. У таких випадках стає складніше застосовувати традиційні моделі колаборативної фільтрації. Так як, колаборативна фільтрація основується на схожості між користувачами та їх взаємодією з об’єктами, то коли в дану систему додається новий користувач, він має спочатку зробити декілька взаємодій, щоб мати історію оцінок, по якій вже можна буде розрахувати схожість з іншими користувачами, та побудувати рекомендації для нього.

Тому було розроблено ряд специфічних методів для полегшення проблеми холодного старту у контексті систем рекомендацій, наприклад ключові слова в профілях користувачів, що мінімізує дану проблему для методів основаних на контенті. Методи, що ґрунтуються на контенті є більш надійними у випадку холодного старту, але інколи схожий контент все ж таки не завжди може бути доступний.

1.2.6 Висновки до аналізу рекомендаційних систем

Рекомендаційні системи є важливим інструментом для забезпечення персоналізованих рекомендацій користувачам у різних областях, таких як електронна комерція, медіа, соціальні мережі тощо. Одними з ключових підходів є колаборативна фільтрація та методи засновані на контенті.

У колаборативній фільтрації виділяють два типи: фільтрація на основі користувачів та фільтрація на основі об'єктів. Перший тип враховує схожість між користувачами, другий - між об'єктами. Для визначення схожості використовуються різні методи, включаючи косинусну схожість для порівняння векторів користувачів/об'єктів, а також кореляційний аналіз для вимірювання лінійної залежності між оцінками.

З використанням методів, що ґрунтуються на контенті, пов'язані різні переваги та обмеження порівняно з системами колаборативної фільтрації. Однак, зазначений підхід може включати ширший спектр методів, де користувачі можуть самі вказувати ключові слова у своїх профілях для отримання рекомендацій.

Однією з головних переваг над методами колаборативної фільтрації є те, що дані методи на основі контенту мінімізують одну з її найбільших проблем - холодний старт, а також надають більш персоналізовані рекомендації.

1.3 Постанова задачі

Провести теоретичний та практичний аналіз системи рекомендацій, заснованої на методах контентної фільтрації. Важливо ретельно розглянути ключові етапи таких систем, оглянути та порівняти різноманітні існуючі методи системи на основі фільтрації контенту. На основі аналізу необхідно спроектувати та програмно реалізувати систему, яка рекомендує елементи

користувачеві, на основі контентної фільтрації. Протестувати дану систему, знайти точність схожості рекомендованих елементів.

Процес розробки передбачає декілька ключових етапів: предобробки даних, ефективне вилучення, вагування та трансформація ознак, а також підготовка набору даних для видачі точних та релевантних рекомендацій. Програмно реалізувати рекомендаційну систему за допомогою мови програмування Python, а також з використанням можливостей бібліотеки Scikit-learn, що дозволить експлуатувати перевірені на практиці сценарії аналізу даних. Протестувати систему на наборі даних [5], використовуючи існуючі елементи для пошуку рекомендацій.

Кінцевою метою є розробка рекомендаційної системи, здатної забезпечити підбір контенту, адаптованого до індивідуальних вподобань та потреб користувачів.

2 ТЕОРЕТИЧНЕ ДОСЛІДЖЕННЯ

2.1 Рекомендаційна система на основі контенту

Системи на основі контенту мають певні основні компоненти, які залишаються незмінними у різних варіаціях таких систем. Оскільки системи на основі контенту працюють з різноманітними описами елементів та інформацією про користувачів, потрібно перетворити ці різні типи структурованої інформації в стандартизовані описи. У більшості випадків вибір перетворення описів елементів на ключові слова є бажаним. Отже, системи на основі контенту в основному, працюють у текстовому домені. Багато природних застосувань систем на основі контенту також орієнтовані на текст. Наприклад, системи рекомендацій статей часто базуються на контенті і також є текстово-орієнтованими системами. Загалом, методи класифікації тексту та регресійне моделювання залишаються найбільш використовуваними інструментами для створення систем рекомендацій на основі контенту.

Основні компоненти систем на основі контенту включають попередню обробку та навчання, які проводяться до використання системи, а також передбачення, що здійснюється у реальному часі. Частини, що виконуються до запуску системи, стосуються створення узагальненої моделі, яка часто є моделлю класифікації або регресії. Ця модель потім застосовується для генерації рекомендацій користувачам під час їх взаємодії з системою. Розглянемо різні компоненти систем на основі контенту.

Передобробка та вилучення ознак — системи на основі контенту застосовуються у різноманітних сферах, таких як веб-сторінки, описи товарів, новини, музичні особливості та інше. У більшості випадків ознаки вилучаються з цих різних джерел для перетворення їх у векторне представлення простору ключових слів. Це перший крок будь-якої системи рекомендацій на основі

контенту і сильно залежить від конкретної області. Однак, правильне вилучення найінформативніших ознак є ключовим для ефективної роботи будь-якої системи рекомендацій на основі контенту.

Навчання профілів користувачів на основі контенту — як було вже зазначено, модель на основі контенту є специфічною для конкретного користувача. Тому для передбачення інтересів користувача до елементів створюється модель, специфічна для цього користувача, на основі його минулої історії покупок чи оцінок елементів. Для досягнення цієї мети використовується зворотний зв'язок користувача, який може виявлятися у вигляді попередніх визначених оцінок, наприклад вподобайки — явний зворотний зв'язок, або активності користувача, наприклад взаємодія з контентом — неявний зворотний зв'язок. Такий зворотний зв'язок використовується разом із характеристиками елементів для створення навчальних даних. На цих даних створюється модель навчання. Ця стадія часто не дуже відрізняється від моделювання класифікації або регресії, в залежності від того, чи є зворотний зв'язок категоріальним, наприклад бінарний вибір елемента, чи числовим (оцінки чи частота покупок). Отримана модель називається профілем користувача, оскільки концептуально пов'язує інтереси користувача, його оцінки з характеристиками елементів.

Фільтрації та рекомендації: на цьому етапі використовується навчена модель з попереднього кроку для надання рекомендацій щодо елементів для конкретних користувачів. Важливо, щоб цей етап був дуже ефективним, оскільки передбачення мають здійснюватися в реальному часі.

2.2 Передобробка та вилучення ознак

Перша фаза у всіх моделях на основі контенту - це вилучення розрізняючих ознак для представлення елементів. Розрізняючі ознаки - це ті, які мають високий прогностичний характер для інтересів користувача. Ця фаза

сильно залежить від конкретного застосування. Наприклад, система рекомендацій статей буде значно відрізнятися від системи рекомендацій продуктів.

2.2.1 Вилучення ознак

На етапі вилучення ознак отримують описи різних елементів. Хоча можна використовувати будь-який тип представлення, такий як багатовимірне представлення даних, найбільш поширеним підходом є отримання ключових слів з основних даних. Цей вибір зумовлений тим, що неструктуровані текстові описи часто широко доступні у різних областях і залишаються більш природним способом опису елементів. У багатьох випадках елементи можуть мати кілька полів, які описують різні аспекти елемента. Наприклад, інтернет-магазин, який спеціалізується на продажі побутової техніки, може мати детальні специфікації продуктів та ключові слова, які описують функції, бренди та моделі приладів. У деяких випадках ці специфікації можуть бути перетворені в набір характеристик у формі "мішка слів". У інших ситуаціях, можна використовувати безпосередньо структуроване представлення даних. Це стає необхідним, коли характеристики товарів включають числові параметри, такі як об'єм, потужність або вибір з обмеженого набору опцій такі як, тип управління, клас енергоефективності.

Різні поля повинні бути адекватно зважені для полегшення їх використання у процесі класифікації. Зважування ознак тісно пов'язане з вибором ознак: у першому випадку це м'яка версія останнього. У випадку вибору ознак атрибути включаються чи не включаються в залежності від їхньої важливості, тоді як у зважені ознак їм надаються різні ваги залежно від їхньої значущості.

2.2.2 Представлення та очищення ознак

Цей процес особливо важливий, коли використовується неструктурований формат для представлення. Фаза вилучення ознак може визначити мішки слів з неструктурованих описів продуктів або веб-сторінок. Однак ці представлення потребують очищення та перетворення в відповідний формат для обробки. Далі розглянемо етапи у процесі очищення.

Вилучення стоп-слів: багато тексту, що видобувається з описів елементів у вільній формі, містить багато слів, які не є специфічними для елемента, але є загальною частиною будь-якого мовного словника. Такі слова зазвичай є високочастотними словами. Наприклад, слова, такі як "або", "як", "до" і "що", не будуть особливо конкретними для даного елемента. У додатку для рекомендації фільмів дуже поширені такі слова у синопсисі. Загалом артиклі, прийменники, сполучники та займенники розглядаються як стоп-слова. У більшості випадків існують стандартизовані списки стоп-слів у різних мовах.

Злиття: під час злиття різні варіації одного слова об'єднуються. Наприклад, однина і множина слова чи різні часи одного слова об'єднуються. У деяких випадках вилучаються спільні корені з різних слів. Наприклад, слова "синенький" і "синюватий" консолідуються в загальний корінь "син". Звісно, злиття іноді може мати негативний вплив, оскільки слово "син" має власний відмінок. Для злиття існує багато готових інструментів, доступних для використання.

Вилучення фраз: Ідея полягає у виявленні слів, які часто зустрічаються разом у описах. Наприклад, фразеологізм "серце з перцем" має інший зміст порівняно зі своїми складовими словами. Існують словники, які були написані вручну, для вилучення фраз, хоча також можуть використовуватися автоматизовані методи.

Після виконання цих кроків, ключові слова перетворюються у векторне представлення. Кожне слово також називається терміном. У векторному представленні, документи подаються у вигляді мішків слів разом з їх частотами. І хоч вживання слів з високою частотою може виглядати привабливим, такий підхід, як правило, не є оптимальним. Пояснюється це тим, що слова, що зустрічаються найчастіше, зазвичай містять менше важливої інформації, для відрізнення текстів. Тому такі слова часто знижують вагу. Це схоже на принцип стоп-слів, крім того, що це виконується шляхом зниження ваги слова, а не повністю його вилучення.

Спосіб зниження ваги слів досягається за допомогою поняття інверсної частоти документів. Інверсна частота документів для і-го терміна обернено пропорційна кількості документів, де цей термін зустрічається.

$$idf_i = ln(\frac{n}{n_i}) \tag{2.1}$$

де idf_i - інверсна частота документів для і-го терміну;

n - кількість документів у колекції.

Крім того, потрібно пильнувати, щоб надмірне входження одного слова в колекції не отримувало занадто великої ваги. Наприклад, коли описи елементів збираються з ненадійних джерел або відкритих платформ, таких як Інтернет, вони можуть містити значну кількість спаму. Для досягнення цієї мети до частот може опціонально застосовуватися функція згладжування, така як квадратний корінь або логарифм, перед обчисленням подібності.

$$f(v_i) = \sqrt{v_i} \tag{2.2}$$

де $f(v_i)$ - функція згладжування;

v_i - частота входження слова.

$$f(v_i) = ln(v_i) \tag{2.3}$$

де $f(v_i)$ - функція згладжування;

v_i - частота входження слова.

Згладжування частоти є необов'язковим і часто виключається. Нормалізована частота для і-го слова визначається шляхом поєднання інверсивної частоти документів з функцією згладжування:

$$N(v_i) = f(v_i)idf_i \tag{2.4}$$

де $N(v_i)$ - нормалізована частота;

$f(v_i)$ - функція згладжування;

df_i - інверсна частота документів.

Ця модель популярно відома як модель TF-IDF, де TF відображає частоту терміну, а IDF - інверсну частоту документів.

2.2.3 Збір уподобань користувачів

Окрім вмісту про елементи, також потрібно збирати дані про уподобання користувачів для процесу рекомендацій. Збір даних відбувається до початку використання системи, тоді як рекомендації визначаються в реальному часі, коли конкретний користувач взаємодіє з системою. Користувач, для якого в даний момент виконується передбачення, називається активним користувачем. Вподобання користувача поєднуються з вмістом для створення передбачень в реальному часі. Дані про уподобання користувачів можуть мати одну з наступних форм:

- оцінки: у цьому випадку користувачі вказують оцінки, що вказують на їхні уподобання стосовно елемента. Оцінки можуть бути бінарними - таблиця 2.1, інтервальними - таблиця 2.2, або порядковими. У рідкісних випадках оцінки можуть бути навіть дійсними числами. Характер оцінки має значний вплив на модель, яка використовується для навчання профілів користувачів;
- неявний зворотний зв'язок - вказує на дії користувача, такі як купівля або перегляд елемента. У більшості випадків неявний зворотний зв'язок зафіксовує лише позитивні вподобання користувача, але не відображає негативних оцінок;
- текстові відгуки: у багатьох випадках користувачі можуть висловлювати свої думки у формі текстових описів. У таких випадках неявні оцінки можуть бути здобуті з цих відгуків. Ця форма вилучення оцінок вже пов'язана з галуззю аналізу думок і виявлення настроїв.
- приклади: користувачі можуть вказувати приклади елементів, які їх цікавлять. Такі приклади можуть використовуватись як неявний зворотний зв'язок з класифікаторами найближчих сусідів або класифікаторами Роккіо. Однак коли використовується пошук подібності разом із ретельно

розробленими функціями корисності, ці методи більше пов'язані з системами рекомендацій на основі прикладів.

В усіх вищезазначених випадках уподобання користувача до елементу остаточно перетворюються на унітарну, бінарну, інтервальну або реальну оцінку. Цю оцінку також можна розглядати як мітка класу або залежної змінної, яка в підсумку використовується для навчальних цілей.

Таблиця 2.1 – Візуалізація бінарної оцінки

	Слово 1	Слово 2	Слово 3	Оцінка
Елемент 1	Негативна	Позитивна	Позитивна	Позитивна
Елемент 2	Позитивна	Негативна	Негативна	Негативна
Елемент 3	Негативна	Негативна	Позитивна	?

Таблиця 2.2 – Візуалізація інтервальної оцінки

	Слово 1	Слово 2	Слово 3	Оцінка
Елемент 1	9	9		9
Елемент 2			3	3
Елемент 3			3	?

2.2.4 Навчальний відбір та вагування

Мета відбору та вагування полягає у тому, щоб в векторному представленні залишалися лише найінформативніші слова. Фактично, багато відомих систем рекомендацій явно підтримують використання обмеження на кількість ключових слів. Експериментальні результати, які проводилися у різних галузях, вказують на те, що кількість видобутих слів повинна бути десь між 60 та 250. Основна ідея полягає в тому, що шуми часто призводять до перенавчання, отже, їх слід попередньо видаляти. Це особливо важливо, враховуючи той факт, що кількість документів, доступних для вивчення конкретного користувацького профілю, часто не дуже велика. Коли кількість документів для навчання невелика, схильність моделі до перенавчання збільшується. Тому важливо зменшити розмір простору ознак.

Є два відмінні аспекти включення інформативності ознак у представлення документів. Перший - це відбір ознак, що відповідає видаленню слів. Другий - це вагування ознак, яке передбачає надання більшого значення словам. Варто зазначити, що видалення стоп-слів та використання оберненої частоти документів є прикладами відбору та вагування ознак відповідно. Проте ці методи відбору та вагування ознак є методами без вчителя, оскільки не включають в себе відгуки користувачів.

Методи обчислення інформативності ознак можуть використовуватися або для жорсткого відбору ознак, або для евристичного зважування ознак за допомогою функції обчисленої кількісної характеристики інформативності. Міри, які використовуються для оцінки інформативності ознак, також різняться в залежності від того, чи рейтинг користувача трактується як числове або категоріальне значення. Наприклад, в контексті бінарних оцінок, або оцінок з

невеликою кількістю дискретних значень, має сенс використовувати категоріальні, а не числові представлення.

2.3 Навчання профілів користувачів та фільтрація

Навчання профілів користувачів тісно пов'язане з проблемою моделювання класифікації та регресії. Коли рейтинги трактуються як дискретні значення, наприклад "вподобайка" чи "дизлайк", проблема схожа на те, що відбувається у текстовій класифікації. З іншого боку, коли рейтинги трактуються як набір числових значень, проблема подібна до задачі регресійного моделювання. Крім того, проблему навчання можна поставити як у структурованих, так і в неструктурованих областях. Для узгодженості будемо вважати, що описи елементів виглядають у формі документів. Проте підхід легко узагальнюється до будь-якого типу багатовимірних даних, оскільки текст є спеціальним типом багатовимірних даних.

У кожному випадку ми припускаємо, що у нас є набір навчальних документів — НД, які позначені певним користувачем. Цього користувача також називають активним, коли він отримує рекомендацію від системи. Навчальні документи відповідають описам елементів, які були вилучені на етапі попередньої обробки та відбору ознак. Крім того, навчальні дані містять рейтинги, що надані цим документам активним користувачем. Ці документи використовуються для створення навчальної моделі. Зазначимо, що мітки, надані іншими користувачами (крім активного користувача), не використовуються в процесі навчання. Отже, навчальні моделі специфічні для конкретних користувачів і не можуть бути використані для довільно обраних користувачів. Це відрізняється від традиційного колаборативного фільтрування, в якому методи, такі як матрична факторизація, побудовані на одній моделі для

всіх користувачів. Навчальна модель для конкретного користувача відображає профіль користувача.

Мітки на документах відповідають числовим, бінарним або унікальним рейтингам. Припустимо, що і-й документ у НД має рейтинг, позначений Рі. Також маємо набір тестових документів, які не мають міток — ТД. Важливо відзначити, що як НД, так і ТД специфічні для певного (активного) користувача. Тестові документи можуть відповідати описам елементів, які потенційно можуть бути рекомендовані користувачеві, але які ще не були придбані або оцінені користувачем. У галузях, таких як рекомендації статей, документи в ТД можуть відповідати потенційним веб-документам для рекомендацій активному користувачу. Точне визначення ТД залежить від галузі, але окремі документи в ТД вилучаються аналогічно до тих у НД. Навчальна модель на НД використовується для предоставлення рекомендацій з ТД активному користувачу. Як і у випадку колаборативного фільтрування, модель може використовуватися для надання передбачуваного значення рейтингу або для складання впорядкованого списку рекомендацій топ-k.

Ця проблема відразу ж нагадує проблеми класифікації та регресійного моделювання у текстовій області.

2.3.1 Класифікація за найближчим сусідством

Основна задача полягає в пошуку подібності контенту, існує багато методів пошуку, такі як косинусна схожість формула(1.1). Розглянемо метод k-найближчих сусідів, який найбільш ефективний з перелічених раніше методів та відносно стійкий до шумів. На рисунку 2.1 візуально представленно метод K-NN.

k-Найближчих Сусідів (k-NN) є методом навчання без вчителя, тобто це алгоритм, який не навчається на даних, а використовується

безпосередньо для визначення схожості між об'єктами на основі їх характеристик. Так як модель k-NN взагалі не навчається на даних, то він просто запам'ятовує характеристики об'єктів у наборі даних. Коли потрібно зробити рекомендацію для нового об'єкта, модель знаходить k-найбільш схожих об'єктів у вже відомому наборі даних за допомогою певних метрик схожості, наприклад, евклідової відстані. Потім вона робить рекомендації на основі того, що сподобалося чи було відмічено користувачами для цих k найбільш схожих об'єктів. Відсутність етапу навчання робить k-NN досить простим та зручним методом, але залишається проблема холодного старту, та необхідність обрати оптимальне значення k.

Найближчий сусід (nearest neighbor) - один з найпростіших методів класифікації, який можна реалізувати досить прямолінійно. Першим кроком є визначення функції схожості, яка використовується в класифікаторі найближчого сусіда. Найбільш поширеною функцією схожості є косинусна функція. Нехай (u, v) - пара документів, в яких нормалізовані частоти слова i задані відповідно u_i та v_i в двох документах. Зауважте, що ці частоти нормалізовані або зважені за допомогою зважування без вчителя TF-IDF або навчальних методів. Тоді косинусна міра визначається за допомогою цих нормалізованих частот за формулою(1.1).

Косинусна схожість часто використовується в текстовій області через її здатність адаптуватися до різних довжин основних документів. Коли цей підхід застосовується до інших типів структурованих і багатовимірних даних, використовуються інші функції схожості/відстані, такі як евклідова відстань і манхеттенська відстань. Для реляційних даних з категоріальними атрибутами доступні різноманітні вимірювання схожості на основі відповідності.

Ця функція схожості корисна для прогнозування оцінок для елементів, в яких відомі уподобання користувача. Для кожного документа в ТД

використовується косинусна схожість для визначення його k-найближчих сусідів в НД. Для кожного елемента в ТД обчислюється середнє значення оцінки для k сусідів. Це середнє значення є передбачуваною оцінкою для відповідного елемента в ТД. Додатковий евристичний аспект полягає в тому, що кожну оцінку можна зважити значенням схожості. У випадках, коли оцінки розглядаються як категоріальні значення, визначається кількість голосів за кожне значення оцінки, та передбачається значення оцінки з найбільшою частотою. Документи в ТД потім ранжуються за передбаченим значенням оцінки, і верхні елементи рекомендуються користувачеві.

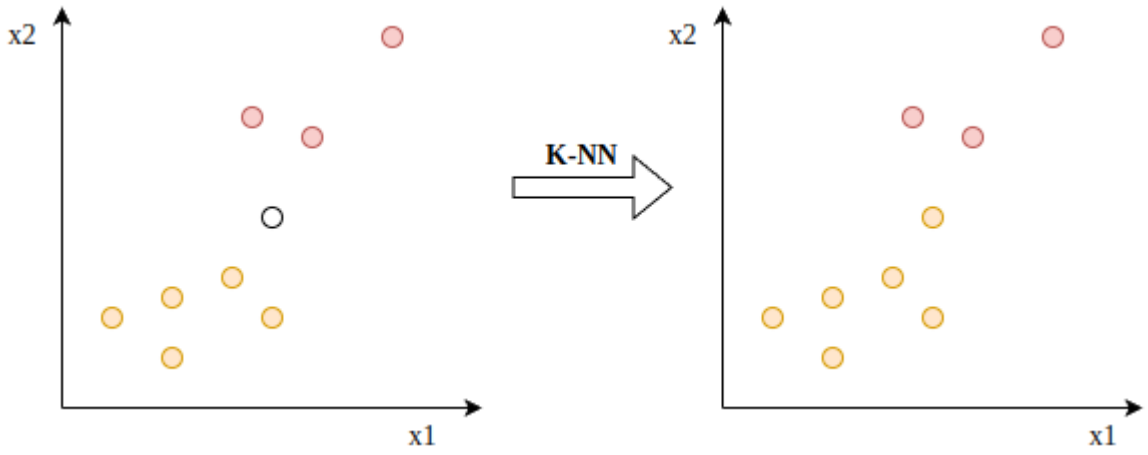

Рисунок 2.1 – Візуальне представлення методу K-NN

2.3.2 Класифікатор на основі правил

Класифікатори на основі правил у системах рекомендацій на основі контенту використовують набір правил для призначення рейтингів чи класів до конкретних елементів. Ці правила визначають умови, які вказують, який рейтинг чи клас має бути призначений кожному елементу. Наприклад, якщо елемент має певні ключові слова, схожість з попередніми вподобаннями користувача, або

визначені атрибути, то йому може бути призначений певний рейтинг чи клас. Ці правила допомагають системі автоматично призначати рейтинги чи класи новим елементам, що рекомендуються користувачам.

Класифікатори на основі правил у системах рекомендацій на основі контенту подібні до класифікаторів на основі правил у колаборативній фільтрації. У правилах елемент-елемент у колаборативній фільтрації як передумови, так і наслідки правил відповідають рейтингам елементів. Основна відмінність полягає в тому, що передумови правил у колаборативній фільтрації відповідають рейтингам різних елементів, у той час як передумови правил у методах на основі контенту відповідають наявності конкретних ключових слів у описах елементів. Отже, правила мають наступний вигляд:

- якщо елемент містить набір ключових слів A, тоді ставимо рейтинг — подобається;
- якщо елемент містить набір ключових слів B, тоді ставимо рейтинг — не подобається.

Таким чином, передумова правила вважається "виконаною" для певного представлення ключових слів елемента, якщо всі ключові слова в передумові містяться в цьому рядку. Наслідки відповідають різним рейтингам, які, для спрощення, ми вважаємо бінарними - подобається чи не подобається. Рядок вважається "виконаним" для наслідка того правила, якщо значення рейтингу в наслідку відповідає рейтингу цього рядка.

Перший крок - використати активний профіль користувача (тобто навчальні документи) для отримання всіх правил на бажаному рівні підтримки та достовірності. Як і в усіх методах, основаних на контенті, правила є специфічними для активного користувача. Наприклад, у випадку таблиці 2.3 активний користувач, схоже, зацікавлений в Елемент 1 та Елемент 2. В такому

випадку, прикладом відповідного правила, може бути таке, як представлено на рисунку 2.1.

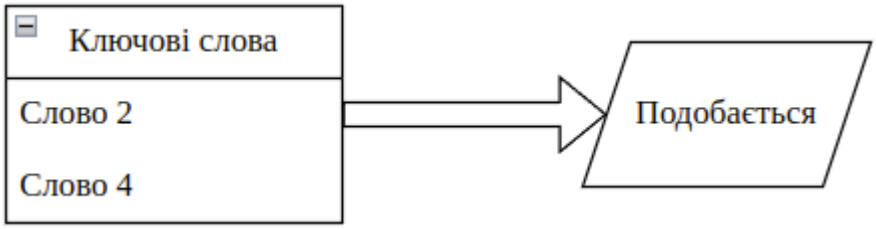

Рисунок 2.2 – Візуалізація правила

Таблиця 2.3 – Візуалізація оцінок активного профілю користувача

Ключові слова / Елементи	Слово 1	Слово 2	Слово 3	Слово 4	Подобається або Не подобається
Елемент 1	0	1	0	1	П
Елемент 2	1	1	0	1	П
Елемент 3	1	0	1	0	НП
Тест 1	0	1	0	0	?
Тест 2	1	0	1	0	?

Отже, основна ідея полягає в тому, щоб знайти всі такі правила для даного активного користувача. Потім, для цільових елементів, для яких інтереси користувача невідомі, визначається, які правила спрацьовують. Правило спрацьовує на опис цільового елемента, якщо ключові слова передумови

включені в опис. Як тільки всі спрацьовані правила визначені для активного користувача, середній рейтинг у наслідках цих правил видається як рейтинг цільового елемента. Існує багато різних евристик для поєднання рейтингів у наслідках. Наприклад, можна вибрати вагу рейтингу з достовірністю правила при обчисленні середнього. Якщо ж жодне правило не спрацьовує, потрібно використовувати типові евристики. Наприклад, можна визначити середній рейтинг активного користувача по всім елементам і також визначити середній рейтинг цільового елемента у всіх користувачів. Зважене середнє цих двох значень видається. Отже, загальний підхід до класифікації на основі правил можна описати наступним чином:

- фаза навчання: визначаються всі відповідні правила з профілю користувача на бажаному рівні мінімальної підтримки та достовірності з набору навчальних даних НД;
- фаза тестування: для кожного опису елемента у ТД, визначаються спрацьовані правила та середній рейтинг. Ранжуються елементи у ТД на основі цього середнього рейтингу.

Однією з переваг систем на основі правил є висока рівень інтерпретованості, яку вони надають. Наприклад, для рекомендованого елемента можна використовувати ключові слова в передумові спрацьованих правил, щоб дати рекомендацію цільовому користувачеві щодо того, чому йому може сподобатися певний елемент.

Отже, було розглянуто теоретичну складову рекомендаційної системи на основі контенту, методологію якої основано на аналізі вмісту та користувацьких даних. Визначено, що ключові компоненти такої системи включають попередню обробку даних, вилучення ознак елементів для створення стандартизованих векторних описів, навчання користувацьких профілів та систему фільтрації та рекомендацій у реальному часі.

Зазначено, що системи на основі контенту надають перевагу текстовому домену, а методи класифікації текстів та регресійне моделювання розглядаються як основні інструменти моделювання для генерації рекомендацій. Для оцінки та передбачення інтересів користувачів використовуються специфічні моделі, засновані на історії взаємодії користувача з певними елементами (покупки, оцінки і т.д.) та зворотному зв'язку. Елементи систем виконують різноманітні завдання, починаючи від предобробки та вилучення ознак до побудови моделей і прогнозування. Описано важливість правильного вилучення ознак для ефективної роботи системи.

У підсумку, система на основі контенту обґрунтовує свою ефективність здатністю враховувати оцінки та інтереси специфічних до кожного користувача, що дозволяє генерувати точні та актуальні рекомендації. Однак, системи, засновані на контенті, також мають потенційні обмеження, що включає труднощі з "холодним стартом" та обмежену різноманітність рекомендацій, які вимагають постійного оновлення контенту та оцінок.

3 ПРАКТИЧНЕ ДОСЛІДЖЕННЯ

3.1 Проектування системи рекомендацій на основі контенту

Фільтрація на основі контенту є однією з основних стратегій, яка використовується в системах рекомендацій для запропонування продуктів, новин, або іншого контенту користувачам залежно від їхніх переваг. Даний алгоритм аналізує атрибути об'єктів та рекомендує користувачам ті, що найбільш подібні тим, які вони раніше оцінили позитивно.

Алгоритми фільтрації на основі контенту використовують детальну інформацію про об'єкти, фокусуючись на властивостях або описах цих об'єктів. Двома словами, для кожного об'єкта створюється профіль, що складається з вагових ключових слів й на їх основі робляться рекомендації. Візуалізація алгоритму представлено на рисунку 3.1. Розглянемо процес фільтрації:

- вилучення ознак: спочатку відбувається вилучення ознак з описів елементів, тобто збір ключових слів, характеристик тощо;
- представлення та очищення ознак: попередній етап може визначити мішки слів з неструктурованих описів елементів, але ці дані ще треба очистити та перетворити в відповідний формат. У процесі відбувається вилучення стоп-слів, злиття, вилучення фраз. Після виконання цих методів ключові слова перетворюються у векторне представлення. Мішки слів з їх частотами представляють собою документи, далі потрібно зменшити вагу слів у яких занадто висока частота, бо вони містять менше корисної інформації, зазвичай для цього використовується підхід TF-IDF;
- збір уподобань користувачів: збираються оцінки від користувачів, це можуть бути як явні оцінки, наприклад вподобайки, так і не явні, такі як активність користувача з елементом;

- навчальний відбір та вагування: відбувається процес обробки векторних представлень, щоб залишити тільки найінформативніші слова. Це досягається за допомоги відбору ознак, коли видаляються непотрібні слова, та вагуванням слів;
- фільтрація: нові елементи порівнюються з профілем активного користувача, його навчальними документами, які були отримані на попередніх етапах, і система рекомендує елементи, що мають найбільшу схожість. Для цього використовуються алгоритми для пошуку схожості, такі як косинусна схожість, k-NN, класифікатор на основі правил тощо.

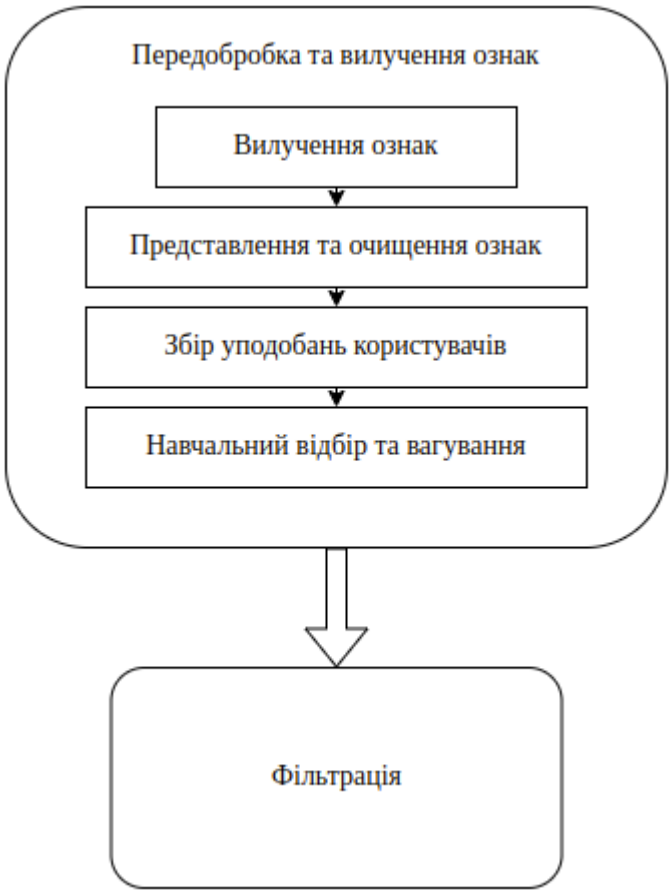

Рисунок 3.1 – Візуалізація алгоритму фільтрації на основі контенту

Рекомендаційна система на основі контенту персоналізує рекомендації, враховуючи специфічні особливості об'єктів, без необхідності аналізу великих масивів поведінка інших користувачів, як у колаборативній фільтрації. Це дозволяє робити рекомендації швидкими, влучними, унікальними для кожного користувача.

3.2 Програмна реалізація

Для програмної реалізації використана мова програмування Python, яка має низку переваг перед іншими мовами програмування, коли мова йде про рекомендаційні системи. З головних переваг можна виділити простоту, швидкість розробки та наявність бібліотек, які надають багато готових методів для розрахунків.

3.2.1 Бібліотека Scikit-learn

Scikit-learn, далі Sklearn, це відкрите програмне забезпечення для машинного навчання, побудоване на мові програмування Python. Вона широко використовується для створення моделей машинного навчання, включаючи класифікацію, регресію, кластеризацію та рекомендаційні системи. Бібліотека використовується в багатьох областях, включаючи аналіз даних тощо. Також дана бібліотека особлива тим що має інструменти для створення моделей рекомендаційних систем, як на основі контенту, так й інших. Загалом Sklearn пропонує широкий спектр алгоритмів машинного навчання та статистичних інструментів, що дозволяє ефективно створювати й оцінювати моделі. Для рекомендаційних систем на основі контенту ця бібліотека дозволяє використовувати алгоритми класифікації та регресії для прогнозування вподобань користувачів щодо предметів на основі їхніх характеристик.

Основні переваги даної бібліотеки перед альтернативними полягають у простоті використання та широкому спектрі доступних алгоритмів. Одна з головних переваг – це простота інтеграції. Sklearn дозволяє легко впроваджувати алгоритми машинного навчання в проекти завдяки зрозумілому інтерфейсу та докладній документації. Крім того, вона містить широкий набір алгоритмів для класифікації, регресії, кластеризації та інших завдань машинного навчання.

Завдяки гнучкості інтеграції з Python, Sklearn відмінно працює з іншими бібліотеками для аналізу даних та обробки інформації. Вона має також вбудовані інструменти для оцінки та налаштування параметрів моделей, що дозволяє вибирати найкращі алгоритми для конкретних задач.

Дана бібліотека користується популярністю у багатьох галузях, від фінансів до медицини, завдяки своїй універсальності та можливостям. Це інструмент, який спрощує роботу з машинним навчанням та дозволяє швидко створювати та налаштовувати моделі для різних завдань.

3.2.2 Набір даних

У якості набору даних використовується "7k Books" [5] , який знаходиться у відкритому доступі. Цей набір даних містить близько 7000 записів та має наступні атрибути: isbn13, isbn10, title, subtitle, authors, categories, thumbnail, description, published_year, average_rating. Представлені атрибути містять назви книг, описи, категорії тощо, що достатньо для побудови рекомендаційної системи на основі контенту. На рисунку 3.2 представлено візуалізацію даного набору.

title	subtitle	authors	categories	thumbnail	description
Gilead		Marilynne Robinson	Fiction	http://books.google.com/books/content?id=KQZCPgAACAAJ&printsec=frontcover&img=1&zoom=1&source=gbs_ap...	A NOVEL THAT READERS and critics have been eagerly anticipating for over a decade, Gilead is an asto...
Spider's Web	A Novel	Charles Osborne;Agatha Christie	Detective and mystery stories	http://books.google.com/books/content?id=gA5GPgAACAAJ&printsec=frontcover&img=1&zoom=1&source=gbs_ap...	A new 'Christie for Christmas' -- a full-length novel adapted from her acclaimed play by Charles Osb...
The One Tree		Stephen R. Donaldson	American fiction	http://books.google.com/books/content?id=OmQawwEACAAJ&printsec=frontcover&img=1&zoom=1&source=gbs_ap...	Volume Two of Stephen Donaldson's acclaimed second trilogy featuing the compelling anti-hero Thomas ...
Rage of angels		Sidney Sheldon	Fiction	http://books.google.com/books/content?id=FKo2TgANz74C&printsec=frontcover&img=1&zoom=1&source=gbs_ap...	A memorable, mesmerizing heroine Jennifer -- brilliant, beautiful, an attorney on the way up until t...
The Four Loves		Clive Staples Lewis	Christian life	http://books.google.com/books/content?id=XhQ5XsFcpGIC&printsec=frontcover&img=1&zoom=1&source=gbs_ap...	Lewis' work on the nature of love divides love into four categories; Affection, Friendship, Eros and...
The Problem of Pain		Clive Staples Lewis	Christian life	http://books.google.com/books/content?id=Kk-uVe5QK-	"In The Problem of Pain, C.S. Lewis, one of the most

Рисунок 3.2 – Візуальне представлення набору даних

3.2.3 Розробка системи

Спочатку відбувається імпорт потрібних для роботи бібліотек. Зокрема Pandas, яка потрібна для роботи з набором даних, Sklearn – пропонує багато інструментів які допомагають з розрахунками для рекомендаційної системи.

Також відбувається завантаження набору даних за допомоги read_csv() з бібліотеки Pandas. Імпорт бібліотек та завантаження набору даних представлено на рисунку 3.3.

```
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
from gensim.parsing.preprocessing import remove_stopwords
from nltk.tokenize import RegexpTokenizer

dataset = pd.read_csv('books.csv')
```

Рисунок 3.3 – Імпорт бібліотек та завантаження набору даних

На рисунку 3.4 зображено завантажений набір даних який ще не був ніяк оброблений. Він наразі має 6810 записів та 12 атрибутів.

```
[8]: dataset
```

[8]:

	isbn13	isbn10	title	subtitle	authors	categories	thumbnail	description	published_year	average_rating	num_pages	ratings_count
0	9780002005883	0002005883	Gilead	NaN	Marilynne Robinson	Fiction	http://books.google.com/books/content?id=KQZCP...	A NOVEL THAT READERS and critics have been eag...	2004.0	3.85	247.0	361.0
1	9780002261982	0002261987	Spider's Web	A Novel	Charles Osborne;Agatha Christie	Detective and mystery stories	http://books.google.com/books/content?id=gA5GP...	A new 'Christie for Christmas' -- a full-lengt...	2000.0	3.83	241.0	5164.0
2	9780006163831	0006163831	The One Tree	NaN	Stephen R. Donaldson	American fiction	http://books.google.com/books/content?id=OmQaw...	Volume Two of Stephen Donaldson's acclaimed se...	1982.0	3.97	479.0	172.0
3	9780006178736	0006178731	Rage of angels	NaN	Sidney Sheldon	Fiction	http://books.google.com/books/content?id=FKo2T...	A memorable, mesmerizing heroine Jennifer -- b...	1993.0	3.93	512.0	29532.0
4	9780006280897	0006280897	The Four Loves	NaN	Clive Staples Lewis	Christian life	http://books.google.com/books/content?id=XhQ5X...	Lewis' work on the nature of love divides love...	2002.0	4.15	170.0	33684.0
...	...	...	...	...	...	...	...	...	...	...	...	...
6805	9788185300535	8185300534	I Am that	Talks with Sri Nisargadatta Maharaj	Sri Nisargadatta Maharaj;Sudhakar S. Dikshit	Philosophy	http://books.google.com/books/content?id=Fv_JP...	This collection of the timeless teachings of o...	1999.0	4.51	531.0	104.0
6806	9788185944609	8185944601	Secrets Of The Heart	NaN	Khalil Gibran	Mysticism	http://books.google.com/books/content?id=XcrVp...	NaN	1993.0	4.08	74.0	324.0
6807	9788445074879	8445074873	Fahrenheit 451	NaN	Ray Bradbury	Book burning	NaN	NaN	2004.0	3.98	186.0	5733.0

Рисунок 3.4 – Завантажений набір даних

За допомоги видалення непотрібних атрибутів, таких як посилання, підзаголовки тощо, та записів з пустими значеннями, що заважають фільтрації, для атрибутів, які будуть використовуватись для надання рекомендацій, відбувається підготовка набору даних. Після даної обробки залишається 6548 записів, та 8 атрибутів. Даний процес зображено на рисунку 3.5.

```
[2]: dataset.drop(columns=['isbn13','isbn10','subtitle','thumbnail'],inplace=True)
     dataset.dropna(subset=['title', 'description'],inplace=True)

[3]: dataset.shape

[3]: (6548, 8)

[4]: dataset.columns

[4]: Index(['title', 'authors', 'categories', 'description', 'published_year',
            'average_rating', 'num_pages', 'ratings_count'],
           dtype='object')
```

Рисунок 3.5 – підготовка набору даних

Для того щоб однакові слова, але які починаються з головної, або з малої букви не рахувалися за різні слова, всі слова назв та описів приводяться до нижнього регістру. На рисунку 3.6 зображено переведення до слів до нижнього регістру.

```
[8]: dataset['title'] = dataset['title'].apply(lambda s: s.lower())
     dataset['description'] = dataset['description'].apply(lambda s: s.lower())
```

Рисунок 3.6 – Лістинг коду для переведення слів до нижнього регістру

Стоп-слова – це слова, які не мають корисної інформації, та не можуть являтися ключовими словами, ці слова тільки заважають при фільтрації та

можуть бути причиною зниження точності рекомендацій. На рисунку 3.7 зображено лістинг-коду, де представлена функція, яка завантажує список стоп-слів з бібліотеки, та перевіряє кожне слово в тексті, і якщо знаходить стоп-слово, то видаляє його з тексту, та при завершенні віддає текст без стоп-слів.

```
[103]: def rm_stopwords(txt):
           return remove_stopwords(str(txt))
```

Рисунок 3.7 – Видалення стоп-слів

Для подальшої роботи зі словами, слова потрібно відокремити від тексту, уникаючи знаків пунктуації, пробілів тощо. Для цього написана функція, яка за допомоги регулярного виразу відокремлює слова в тексті. Дана функція зображена на рисунку 3.8.

```
[51]: def rm_punct(txt):
          regexp = r'\w+'
          return " ".join(RegexpTokenizer(regexp).tokenize(str(txt)))
```

Рисунок 3.8 – Функція видалення непотрібних символів

Ці функції застосовуються до значення головних атрибутів, по яким й проводиться рекомендація, в цьому випадку, це атрибут опису. На рисунку 3.9 представлено лістинг коду застосування цих функцій.

```
dataset['description'] = dataset['description'].apply(rm_stopwords)
dataset['description'] = dataset['description'].apply(rm_punct)
```

Рисунок 3.9 – Застосування функцій для обробки даних

На цьому етапі завершується обробка даних. Загалом проведено 3 маніпуляції з даними: переведення тексту в нижній регістр, видалення стоп-слів, та відокремлення слів від інших символів, таких як знаки пунктуації. На рисунку 3.10 зображено набір даних після їх обробки.

title	authors	categories	description
gilead	Marilynne Robinson	Fiction	novel readers critics eagerly anticipating dec...
spider's web	Charles Osborne;Agatha Christie	Detective and mystery stories	new christie christmas full length novel adapt...
the one tree	Stephen R. Donaldson	American fiction	volume stephen donaldson s acclaimed second tr...
rage of angels	Sidney Sheldon	Fiction	memorable mesmerizing heroine jennifer brillia...
the four loves	Clive Staples Lewis	Christian life	lewis work nature love divides love categories...
...	...	...	...
journey to the east	Hermann Hesse	Adventure stories	book tells tale man goes wonderful amazing jou...
the monk who sold his ferrari: a fable about f...	Robin Sharma	Health & Fitness	wisdom create life passion purpose peace inspi...
i am that	Sri Nisargadatta Maharaj;Sudhakar S. Dikshit	Philosophy	collection timeless teachings greatest sages i...
the berlin phenomenology	Georg Wilhelm Friedrich Hegel	History	volume edition ofhegel s philosophy subjective...

Рисунок 3.10 – Оброблений набір даних

Щоб зіставити назви з індексами, для полегшення подальшої роботи, щоб мати змогу працювати з безпосередньо індексами, виконується функція, з бібліотеки Pandas, яка зображена на рисунку 3.11.

```
[109]: book_to_index = pd.Series(dataset.index, index=dataset['title']).drop_duplicates()
```

Рисунок 3.11 – Зіставлення назв з індексами

Далі використано метод TF-IDF, який застосовується щоб визначити релевантність кожного слова з опису. Якщо слово дуже часто зустрічається в коллекції документів, то вага цього слова знижується, бо зазвичай такі слова несуть менше корисної інформації. Розраховується TF-IDF за формулою 2.4. Також на цьому етапі відбувається перетворення слів у вектор. На рисунку 3.12 зображено застосування даного методу за допомоги бібліотеки Sklearn.

```
[110]: tfidfVector = TfidfVectorizer(stop_words='english')
       book_matrix = tfidfVector.fit_transform(dataset['description'])
```

Рисунок 3.12 – Застосування методу TF-IDF

На рисунку 3.13 показана розмірність матриці, яка означає, що матриця має 6548 записів, тобто кількість книг, та 30335 атрибутів. Атрибути в даній матриці вказуються на кількість ознак, що в нашому випадку являється словами, які доступні в описах до книг.

```
[114]: book_matrix.shape

[114]: (6548, 30335)
```

Рисунок 3.13 – Розмірність матриці

Для розрахунку рекомендації використовується функція recommend, яка приймає в якості параметрів назву та матрицю, і віддає схожі за цими параметрами елементи, тобто рекомендації. Спочатку дана функція шукає індекс елементу, за його назвою, а потім бере опис даного елементу за раніше отриманим індексом, для цього кроку використовується допоміжна функція get_desc. Дана операція зображена на рисунку 3.14.

```
def get_desc(title):
    id = book_to_index[title]
    return [dataset['description'][id]]
```

Рисунок 3.14 – Пошук індексу та опису за назвою

Наступним кроком є отримання TF-IDF вектору для тестових даних, тобто для тієї книги, за якої відбувається пошук рекомендацій. А також, розраховуємо косинусну схожість за допомоги інструментів з бібліотеки Sklearn та сортуємо. Лістинг коду приведено на рисунку 3.15.

```
book_test_matrix = tfidfVector.transform(book_test_desc)

sim_scores = cosine_similarity(book_test_matrix, book_matrix).tolist()[0]
sim_scores = sorted(enumerate(sim_scores), key=lambda i: i[1], reverse=True)
```

Рисунок 3.15 – Розрахунок тестової матриці та знаходження подібностей

Далі функція обирає топ 5 елементи за розрахунком косинусної схожості, та отримує індекси цих елементів. Лістинг коду наведено на рисунку 3.16.

```
sim_scores = sim_scores[1:6]
book_indexes = [i[0] for i in sim_scores]
```

Рисунок 3.16 – Отримання індексів рекомендованих елементів

Останній крок даної функції це пошук елементів за цими індексами, та повернення їх з функції. На рисунку 3.17 наведено данний процес.

```
return [dataset['title'][i] for i in book_indexes]
```

Рисунок 3.17 – Пошук за індексом та повернення рекомендацій з функції

На рисунку 3.18 представлений кінцевий вигляд функції, яка виконує пошук рекомендацій за назвою книги.

```
def recommend(title, book_matrix = book_matrix):
  book_test_desc = get_desc(title)

  book_test_matrix = tfidfVector.transform(book_test_desc)

  sim_scores = cosine_similarity(book_test_matrix, book_matrix).tolist()[0]
  sim_scores = sorted(enumerate(sim_scores), key=lambda i: i[1], reverse=True)

  sim_scores = sim_scores[1:6]
  book_indexes = [i[0] for i in sim_scores]

  return [dataset['title'][i] for i in book_indexes]
```

Рисунок 3.18 – Функція для пошуку рекомендацій

На рисунку 3.19 продемонстрований виклик даної функції, вона надає 5 рекомендацій за книгою Rage of angels. Також на даному рисунку зображені

відповідні ключові слова з описів рекомендованих книг, які використовувались для фільтрації.

```
rec_titles = recommend('rage of angels')
rec_titles
```

```
['deception',
 'what looks like crazy on an ordinary day',
 'shadowmarch',
 'the black dahlia',
 'voice of the gods']
```

```
list(map(lambda t: get_desc(t), rec_titles))
```

```
[['famous writer mistress meet room bed talk play games other sex tell lies work complaint
rs heart'],
 ['remarkable debut novel sizzles sensuality crackles life affirming energy moves reader la
erstanding decade luxe living atlanta ava johnson returned tiny idlewild michigan fabulous
new beginning because ten plus years left problems big city invaded sleepy community childl
portantly ava johnson inexplicabley undeniably falling love'],
 ['mass market paperback tad williams triumphant return high fantasy'],
 ['january 15 1947 torture ravished body beautiful young woman los angeles vacant lot victi
on bucky bleichert lee blanchard warrants squad cops friends rivals love woman obsessed dal
lly postwar hollywood core dead girl s twisted life past extremes psyches into region total
 ['unable avoid drawn terrible conflict auraya protector siyee fears unable meet conditions
my gods now immortal wilds deterred quest powerful long buried secrets deadly adversaries s
```

Рисунок 3.19 – Результат виконання фільтрації

На рисунку 3.20 зображено ключові слова з описів рекомендованих книг, які виділені кольорами, якщо вони мають дублікати в описах інших рекомендованих книгах. Це демонструє ключові слова, які враховувалися при фільтрації, та також те що фільтрація була виконана правильним чином, та результатом рекомендації є схожі за описом книги.

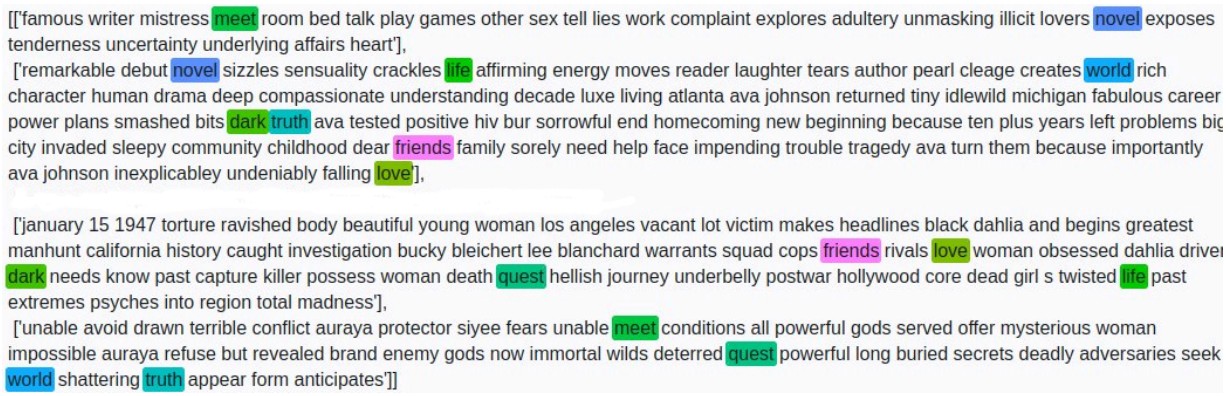

Рисунок 3.20 – Ключові слова з описів рекомендованих книг

За допомоги методів для знаходження схожості, які представлені на рисунку 3.15, було отримано результати точності рекомендацій, які представлені на рисунку 3.21. Перша строка даного списку є елемент, по якому відбувається рекомендації, тому в нього відповідне значення, яке дорівнює 1.

```
sim_sc('taken at the flood')
```

```
[(53, 1.0000000000000002),
 (2443, 0.2524878161725561),
 (86, 0.20230878432676278),
 (44, 0.16358739561866986),
 (45, 0.14845300991102625),
 (55, 0.14536696748024278),
```

Рисунок 3.21 – Рейтинг схожості

З даного рейтингу можна зробити висновок, що в даному наборі даних є тільки 2 книги, які мають достатню схожість для рекомендації. Решта книг з схожістю менше ніж 0.20, занадто різні для рекомендації, отже нерелевантні для активного користувача.

Дана функція recommend, є головною для пошуку рекомендацій, вона є універсальною, це означає що вона може бути використана на різних наборах даних та у різних областях, потрібно лише мати на увазі, що вона ніяк не обробляє метадані набору, тобто атрибути, пусті значення тощо, а також не проводить предообробку даних, тобто переведення в нижній регістр, видалення зайвих символів тощо. Цими операціями займаються інші відповідні функції, які виконуються заздалегідь.

ВИСНОВКИ

В даній роботі було проведено теоретичне та практичне дослідження рекомендаційної системи на основі контенту, яка використовує аналіз метаданих елементів. Ключовими компоненти такої системи є попередня обробка даних, вилучення та очищення ознак елементів для створення стандартизованих векторних описів, вагування ознак, для знаходження слів з високим рівне інформативності, навчання користувацьких профілів та саму фільтрацію.

Система працює переважно з текстовими описами та ключовими словами, а її компоненти здійснюють весь спектр дій, від попередньої обробки даних до побудови моделей і роблення прогнозів. Особливий наголос робиться на якісному вилучені та очищенні ознак для підвищення ефективності рекомендацій.

В практичній частині роботи було проаналізовано підхід до створення системи рекомендацій на основі контенту і описана її програмна реалізація за допомогою мови програмування Python та бібліотеки Scikit-learn. Використовувалася модель TF-IDF для визначення ваги термінів у тексті. Процес розробки включав етапи очищення даних, обробку метаданих, вилучення та перетворення особливостей, а також використання методів для забезпечення точних та релевантних рекомендацій.

Основними кроками створення системи рекомендацій є вилучення ознак з контенту, їх очищення, а потім перетворення у відповідний формат. Стоп-слова та інші нерелевантні символи видаляються з тексту для збільшення точності моделі. Використовувана модель здатна навчатися з векторних представлень, обираючи ключові слова з важливими значеннями, використовуючи TF-IDF.

Ключовою перевагою розглянутого підходу є можливість запуску рекомендацій на основі аналізу характеристик об'єктів без необхідності враховувати поведінку інших користувачів, що забезпечує персоналізації

рекомендацій і ефективність системи. Бібліотека Scikit-learn виявилася ідеальною для таких завдань, завдяки своїй простоті, еластичності та широкому вибору інструментів для роботи з машинним навчанням.

Тестування розробленої системи на наборі даних "7k Books" підтвердило ефективність методу, що базується на фільтрації на основі контенту, надаючи користувачам релевантні рекомендації. Однак, було виявлено, що рекомендаціями для книги "Taken at the flood" є лише дві книги з набору, які мали достатньо високий рівень схожості, що вказує на потенційну потребу в більш детальному аналізі або доопрацюванні алгоритму для покращення його здатності знаходити схожий контент, наприклад це може бути взяття більш широкого спектру ознак, а не тільки описи.

Загалом, отриманий досвід і результати роботи системи рекомендацій підтверджують, що мова програмування Python і бібліотеки машинного навчання здатні забезпечувати ефективний засіб для побудови персоналізованих рекомендаційних систем на основі аналізу контенту. Сама ж система в свою чергу, показала простоту побудови, гнучкість в виборі методів, та ефективність рекомендацій. Робота підтверджує, що рекомендаційна система заснована на контенті є ефективною завдяки здатності адаптуватися до індивідуальних уподобань користувачів і створювати точні рекомендації. Але такі системи стикаються з певними труднощами, зокрема проблема "холодного старту", хоч дана проблема й не така виражена, як в інших системах, зокрема колаборативній фільтрації, та обмежена різноманітність рекомендацій, що потребують регулярного оновлення контенту та користувацьких оцінок.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Jannach, G., Adomavicius, G., Tuzhilin, A. Recommender Systems Handbook. Нью-Йорк : Springer, 2011. 845 с.
2. Aggarwal, C. C. Recommender Systems: The Textbook. Нью-Йорк : Springer, 2016. 518 с.
3. Sarwar, B., Karypis, G., Konstan, J., Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. Proceedings of the 10th International Conference on World Wide Web. Міннеаполіс : University of Minnesota, 2001. 295 с.
4. Melville P.,Mooney R., Nagarajan R. Content-Boosted Collaborative Filtering for Improved Recommendations. Остін : University of Texas, 2002. 192с.
5. Kaggle dataset "7k Books". URL: https://www.kaggle.com/datasets/dylanjcastillo/7k-books-with-metadata/data (дата звернення 10.12.2023)
6. Pazzani, Michael J.; Billsus, Daniel. Content-based recommendation systems. In: The adaptive web: methods and strategies of web personalization. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007. p. 325-341.
7. Wang, Donghui, et al. A content-based recommender system for computer science publications. Knowledge-Based Systems, 2018, 157: 1-9.
8. Thorat, Poonam B.; Goudar, Rajeshwari M.; BARVE, Sunita. Survey on collaborative filtering, content-based filtering and hybrid recommendation system. International Journal of Computer Applications, 2015, 110.4: 31-36.
9. Ko, Hyeyoung, et al. A survey of recommendation systems: recommendation models, techniques, and application fields. Electronics, 2022, 11.1: 141.
10. LÜ, Linyuan, et al. Recommender systems. Physics reports, 2012, 519.1: 1-49.
11. Song, Yading; Dixon, Simon; Pearce, Marcus. A survey of music recommendation systems and future perspectives. In: 9th international symposium on computer music modeling and retrieval. 2012. p. 395-410.

12. Yeremenko, O., Perova, I., Litovchenko, O., Miroshnychenko, N. Framework for Developing a System for Monitoring Human Health in the Combined Action of Occupational Hazards Using Artificial Intelligence and IoT Technologies. Lecture Notes on Data Engineering and Communications Technologies, 2021, 83, pp. 401–410.
13. Бодянский Е.В., Руденко О.Г. Искусственные нейронные сети: архитектуры, обучение, применение. Харьков : ТЕЛЕТЕХ, 2004. 372 с.
14. Nechyporenko, A. A new intelligence-based approach for rhinomanometric data processing / Yerokhin, A., Nechyporenko, A., Babii, A., Turuta, O. // 2016 IEEE 36th International Conference on Electronics and Nanotechnology, ELNANO 2016: Conference Proceedings. – 2016. Vol. 7493047. – P. 198–201.