
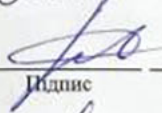
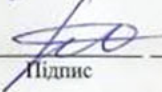


КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА


на тему Рекомендаційна система підбору телесеріалів з використанням
колаборативної фільтрації для вебсистем

Галузь знань 12 – Інформаційні технології
Шифр і назва галузі знань
Спеціальність 122 – Комп'ютерні науки
Шифр і назва спеціальності
Освітня програма Комп'ютерні науки
Назва освітньої програми

Виконав: студент групи КН-20-2  Олександр ПАСІЧНИК
Група виконавця Підпис Ім'я, ПРІЗВИЩЕ
Керівник: к.т.н., доц. каф. КН  Руслан БАГРІЙ
Науковий ступінь, посада Підпис Ім'я, ПРІЗВИЩЕ
Нормоконтроль: к.т.н., доц. каф. КН  Руслан БАГРІЙ
Науковий ступінь, посада Підпис Ім'я, ПРІЗВИЩЕ

До захисту допускаю:

Зав. кафедри КН, д.т.н., професор


Підпис

Олександр БАРМАК

Ім'я, ПРІЗВИЩЕ

17 серпня 2024 р.

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
Факультет інформаційних технологій
Кафедра комп'ютерних наук
Освітній ступінь бакалавр
Галузь знань 12 – Інформаційні технології
Спеціальність 122 – Комп'ютерні науки

ЗАТВЕРДЖУЮ
Завідувач кафедри комп'ютерних наук


(підпис)

д.т.н., професор Олександр БАРМАК
«16» лютого 2024 року

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

1. Тема кваліфікаційної роботи бакалавра: «Рекомендаційна система підбору телесеріалів з використанням колаборативної фільтрації для вебсистем»
2. Завдання видано студенту Олександр ПАСІЧНИКУ
(ім'я, прізвище)
3. Керівник роботи доцент кафедри КН Руслан БАГРІЙ
(посада, ім'я, прізвище)
4. Затверджено наказом університету від «15» лютого 2024 р. № 8
5. Дата видачі завдання студенту: «16» лютого 2024 р.
6. Зміст пояснювальної записки (перелік задач) та вихідні дані:
Мета роботи – покращення сервісу при роботі з сайтами по перегляду телесеріалів шляхом формування персоналізованих рекомендацій щодо перегляду за допомогою колаборативної фільтрації. Розроблювана вебсистема повинна забезпечувати зручний доступ до необхідних даних, аналізувати взаємодію користувачів з вебсистемою та надавати списки рекомендацій.

7. Календарний план виконання кваліфікаційної роботи бакалавра:

| № | Назва етапів (розділів) кваліфікаційної роботи бакалавра | Термін виконання | Примітка |
|---|--|------------------|----------|
| 1 | Вибір напрямку дослідження та узгодження тематики кваліфікаційної роботи бакалавра з керівником, складання календарного графіка виконання роботи | січень 2024 | виконано |
| 2 | Ознайомлення з предметною областю, формулювання мети та задач дослідження, визначення об'єкта та предмета дослідження | січень 2024 | виконано |
| 3 | Робота над розділом 1 – Характеристика предметної області: аналіз моделей, систем та реалізацій | лютий 2024 | виконано |
| 4 | Робота над розділом 2 – Рекомендаційна система на основі колаборативної фільтрації | березень 2024 | виконано |
| 5 | Робота над розділом 3 – Реалізація підбору телесеріалів рекомендаційною системою | квітень 2024 | виконано |
| 6 | Оформлення пояснювальної записки згідно вимог | травень 2024 | виконано |
| 7 | Попередній захист кваліфікаційної роботи бакалавра | травень 2024 | виконано |
| 8 | Захист кваліфікаційної роботи бакалавра | червень 2024 | виконано |

Виконавець: студент групи КН-20-2

Група виконавця

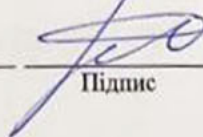

Підпис

Олександр ПАСІЧНИК

Ім'я, ПРІЗВИЩЕ

Керівник: к.т.н., доц. каф. КН

Науковий ступінь, посада


Підпис

Руслан БАГРІЙ

Ім'я, ПРІЗВИЩЕ

Анотація

Тема кваліфікаційної роботи бакалавра: «Рекомендаційна система підбору телесеріалів з використанням колаборативної фільтрації для вебсистем»

Виконавець кваліфікаційної роботи бакалавра: студент групи КН-20-2
Олександр ПАСІЧНИК

Керівник кваліфікаційної роботи бакалавра: к.т.н., доцент кафедри КН
Руслан БАГРІЙ

Кваліфікаційна робота бакалавра містить:

| Пояснювальна записка | | | | Кількість додатків |
|----------------------|----------|---------|-------------------|--------------------|
| Сторінок | Рисунків | Таблиць | Джерел інформації | |
| 67 | 28 | 6 | 34 | 2 |

Мета кваліфікаційної роботи бакалавра – покращення сервісу при роботі з сайтами по перегляду телесеріалів шляхом формування персоналізованих рекомендацій щодо перегляду за допомогою колаборативної фільтрації.

Для реалізації рекомендаційної системи, яка використовує колаборативну фільтрацію було застосовано мову Python та бібліотеки Numpy, Pandas і Surprise. Для реалізації вебсистеми було застосовано мову програмування JavaScript, її розширення TypeScript та фреймворк React.

Практичне використання розробленої рекомендаційної системи дозволяє зекономити час і полегшити пошук цікавого контенту, надаючи персоналізовані рекомендації, що відповідають вподобанням користувача. Цим самим система допомагає відкривати невідомі телесеріали споживачу.

Ключові слова: вебсистема, рекомендаційна система, телесеріали, підбір, колаборативна фільтрація, алгоритм, датасет, схожість, користувач.

Виконавець: студент групи КН-20-2
Група виконавця


Підпис

Олександр ПАСІЧНИК
Ім'я, ПРІЗВИЩЕ

Зміст

| | |
|---|----|
| Перелік скорочень | 3 |
| Вступ..... | 4 |
| Розділ 1 Характеристика предметної області: аналіз моделей, систем та реалізацій..... | 6 |
| 1.1 Аналіз інформаційних моделей..... | 6 |
| 1.2 Огляд систем рекомендацій на основі колаборативної фільтрації..... | 8 |
| 1.3 Аналіз існуючих програмних засобів | 16 |
| 1.4 Мета та завдання кваліфікаційної роботи | 21 |
| Розділ 2 Рекомендаційна система на основі колаборативної фільтрації..... | 22 |
| 2.1 Загальна схема рекомендаційної системи | 22 |
| 2.2 Опис кроків формування рекомендацій у системі | 24 |
| 2.3 Обчислення схожості вподобань користувачів | 27 |
| 2.4 Функціональна структура інформаційної системи для колаборативної фільтрації | 32 |
| 2.5 Проектна архітектура рекомендаційної системи та взаємозв'язок компонентів | 34 |
| 2.6 Інформаційна структура рекомендаційної системи | 37 |
| 2.7 Висновки до розділу 2 | 42 |
| Розділ 3 Програмна реалізація вебсистеми підбору телесеріалів | 44 |
| 3.1 Вебтехнології та засоби розробки рекомендаційної системи | 44 |
| 3.2 Реалізація рекомендаційної системи підбору телесеріалів | 46 |
| 3.2.1 Підготовка набору даних | 46 |
| 3.2.2 Навчання рекомендаційної системи | 49 |
| 3.2.3 Формування рекомендацій..... | 51 |
| 3.4 Тестування рекомендаційної системи..... | 53 |
| 3.5 Висновки до розділу 3 | 61 |
| Загальні висновки..... | 63 |
| Перелік посилань..... | 65 |
| Додатки | |

Перелік скорочень

| Скорочення, термін, позначення | Пояснення |
|---|---|
| БД | База даних |
| ІІ | Штучний інтелект |
| NCF | Neural Collaborative Filtering |
| MLP | Multi-Layer Perceptron |
| VAE-CF | Variational Autoencoder for Collaborative Filtering |
| k-NN | k-Nearest Neighbors |
| SVD | Singular Value Decomposition |
| RMSE | Root Mean Squared Error |
| JS | JavaScript |
| CSS | Cascading Style Sheets |

Вступ

Кваліфікаційна робота бакалавра присвячена розробці рекомендаційної системи підбору телесеріалів з використанням колаборативної фільтрації для вебсистем у вигляді платформи, завдяки якій можна здійснювати пошук телесеріалів за власними вподобаннями.

Актуальність. Через велику кількість надлишкової інформації та контенту, важливо мати інструменти, які допомагають користувачам знаходити те, що їм дійсно цікаво. Персоналізація, на основі штучного інтелекту, може бути ключем до позитивного досвіду взаємодії з технологіями, роблячи його більш приємним та ефективним. Аналізуючи величезні обсяги даних про перегляди, реакції та відгуки глядачів, штучний інтелект пропонує персоналізований контент та прогнозує тенденції.

Об'єкт дослідження – процес формування рекомендацій телесеріалів з використанням колаборативної фільтрації.

Предмет дослідження – методи збору та аналізу інформації, методи колаборативної фільтрації, технології та методи проектування вебсистем перегляду телесеріалів.

Мета кваліфікаційної роботи бакалавра – покращення сервісу при роботі з сайтами по перегляду телесеріалів шляхом формування персоналізованих рекомендацій щодо перегляду за допомогою колаборативної фільтрації.

Завдання кваліфікаційної роботи бакалавра – для досягнення поставленої мети необхідно реалізувати виконання наступних задач:

- провести аналіз систем рекомендацій на основі колаборативної фільтрації;
- розробити рекомендаційну систему, яка використовує колаборативну фільтрацію для формування персоналізованих рекомендацій;
- розробити функціональну та інформаційну структуру вебсистеми перегляду телесеріалів;

- розробити програмну реалізацію вебсистеми підбору телесеріалів з використанням рекомендаційної системи на основі колаборативної фільтрації;
- виконати тестування рекомендаційної системи підбору телесеріалів.

Розділ 1 Характеристика предметної області: аналіз моделей, систем та реалізацій

1.1 Аналіз інформаційних моделей

Телевізійні серіали, що сягають своїм корінням у ранні роки радіомовлення, еволюціонували у динамічну та багатогранну форму розваги, яка захопила увагу глядачів у всьому світі. Від скромних епізодів, об'єднаних однією темою, до епічних саг, що розгортаються протягом кількох сезонів, серіали пропонують безмежний спектр історій, персонажів та світів, які досліджують.

З скромних витоків в епоху радіомовлення телесеріали пройшли шлях еволюції, перетворившись на динамічну та багатогранну форму розваги, що зачаровує глядачів у всьому світі. Від лаконічних епізодів, об'єднаних однією темою, до грандіозних саг, що розгортаються протягом декількох сезонів, серіали пропонують безмежний потік історій, персонажів та захоплюючих світів, які кличуть до дослідження.

Сьогодні телесеріали стали невід'ємною частиною сучасної культури, пропонуючи безмежний спектр жанрів та стилів, що задовольняють будь-які смаки та вподобання. Революційний розвиток інтернету та стрімінгових платформ зробив доступ до масової культури безпрецедентно простим.

Завдяки інноваціям кожен користувач може не лише споживати контент, але й обирати з безлічі альтернатив, ділитися враженнями та думками з усім світом. Це перетворило пасивного глядача на активного учасника культурного процесу, який може впливати на розвиток улюблених історій та формувати нові тренди [1].

Роль телесеріалів у суспільстві зазнала значних змін. З одного боку, вони залишаються важливим медіа-інструментом, що використовується для освіти, розваги, соціальної критики та інших цілей. З іншого боку, їм доводиться конкурувати з безліччю онлайн-платформ, які пропонують різноманітний, персоналізований та інтерактивний контент. Поява телесеріалів в інтернеті та стрімінгових сервісів дала можливість кожному користувачеві стати куратором

власного контенту. Тепер не потрібно чекати на вибір телевізійних каналів, адже можна самостійно обирати, що дивитися, коли це робити, з ким ділитися своїми враженнями та як саме впливати на розвиток улюблених історій [2].

Зміни в технологіях та способах споживання контенту відкрили нові можливості для незалежних творців та акторів. Стрімінгові платформи надають можливість для незалежних авторів та акторів випускати свої роботи на велику аудиторію, що раніше було практично неможливим. Це дозволило зробити телевізійний контент більш різноманітним та інноваційним, а також підтримати талановитих авторів та акторів, які можуть не знайти своє місце на традиційних телевізійних каналах.

Однак, зростання кількості контенту та різноманіття платформ також створює нові виклики для глядачів. Вони змушені витратити більше часу на вибір контенту, а також платити за підписку на декілька платформ, щоб мати доступ до усього бажаного контенту. Це може призвести до перевантаження та втоми від великої кількості доступного контенту.

Також через безмежний доступ до інформації та розваг, користувачі часто стикаються з проблемою надлишку контенту. Вибір того, що подивитися, послухати або прочитати, може перетворитися на справжнє випробування, адже кількість доступних варіантів здатен збити з пантелику навіть найдосвідченішого користувача.

Години, витрачені на перегляд безлічі варіантів, можуть призвести до розчарування, втоми від пошуку та навіть до втрати інтересу до контенту загалом [3]. Саме тому так важливо мати інструменти та рекомендації, які допоможуть користувачам знайти те, що дійсно відповідає їхнім смакам та інтересам.

Сфера персоналізації відкриває нові можливості, змінюючи способи взаємодії з технологіями. Замість того, щоб витратити час на пошук, користувачі можуть зосередитися на тому, що їм дійсно цікаво. Це робить користувацький досвід більш комфортним, персоналізованим та залученим [4].

Технології штучного інтелекту допомагають збирати та аналізувати величезний обсяг даних про перегляди, реакції та відгуки глядачів. Це дає змогу не лише пропонувати персоналізований контент, але і передбачати тенденції та реагувати на них заздалегідь.

Зокрема, аналітика та збір даних допомагають виявити популярні теми, жанри та сюжетні лінії, які варто експлуатувати в майбутніх проектах. Такий підхід не лише сприяє створенню більш збалансованого та привабливого контенту, але й забезпечує більшу ефективність у виробництві та рекламі.

1.2 Огляд систем рекомендацій на основі колаборативної фільтрації

При сучасній доступності та різноманітності телевізійних серіалів виникає потреба в ефективних інформаційних моделях для їх відбору та оцінки. Сучасні глядачі, поринуті в море контенту, шукають способи знайти серіали, які відповідають їхнім інтересам і вподобанням. Інформаційні моделі стають важливим інструментом в цьому процесі, допомагаючи орієнтуватися у великому обсязі доступного контенту. Підбір телесеріалів стає не лише розважальним заняттям, а й складною задачею в умовах інформаційного перенасичення.

Однією з найпоширеніших інформаційних моделей є рекомендаційні системи, які використовуються в стрімінгових платформах та інтернет-ресурсах. Ці системи аналізують інформацію про перегляди, оцінки та поведінку користувачів, щоб запропонувати персоналізовані рекомендації. Ці алгоритми аналізують великі обсяги даних, щоб зрозуміти вподобання кожного конкретного глядача. Такий підхід дозволяє точно серіали, які ймовірно зацікавлять аудиторію, забезпечуючи їм більш особистий та задовільний досвід перегляду.

Однак рекомендаційні системи мають свої обмеження. Вони часто пропонують контент на основі популярності, попередніх переглядів або схожості з іншими фільмами і серіалами, іноді ігноруючи індивідуальні смаки або

новаторські творчі підходи. Також, вони можуть піддаватися ефекту фільтрування та ускладнювати доступ до різноманітних і нестандартних жанрів та концепцій.

Також, важливо враховувати індивідуальні смаки та уподобання користувачів. Це може бути досягнуто шляхом проведення опитувань та анкетування, а також за допомогою інструментів для самоорієнтації глядачів у світі телевізійних серіалів. Наприклад, можна створити інтерактивні тести або ігри, які допоможуть користувачам краще зрозуміти свої смаки та вподобання, а також відкрити для них нові жанри та теми.

Алгоритми штучного інтелекту вдосконалюють взаємодію користувачів з медіа-платформами, адаптуючи рекомендації контенту до їхніх унікальних потреб і вподобань. Вони аналізують історію перегляду кожного користувача, його вподобання, а також враховують різні фактори, такі як жанр, тривалість, рейтинги та рецензії. Це допомагає скоротити час, який користувач витрачає на пошук відповідного контенту, і значно поліпшує його загальний досвід перегляду [5].

Також алгоритми ШІ спроможні враховувати актуальні тенденції у світовому медіа, надаючи користувачам доступ до останніх новин та популярних тенденцій. Це дозволяє користувачам бути на кшталті останніх подій та інформаційно проінформованими, роблячи їхнє медіа-споживання більш збалансованим та насиченим.

Застосування ШІ для персоналізованих рекомендацій відкриває нові можливості пошуку та споживання контенту. Шляхом аналізу даних про поведінку користувачів, їхні вподобання та інтереси, ці алгоритми здатні створювати унікальні підбірки контенту, які точно відповідають індивідуальним потребам кожної особи точково.

Цей підхід не лише економить час користувачів, але й робить їхній досвід споживання медіа значно більш приємним та корисним, зменшуючи зусилля, витрачене на пошук необхідного контенту. Відтепер користувачам не доводиться шукати довго серед безлічі варіантів, адже алгоритми ШІ

пропонують саме те, що вони можуть зацікавити. Це створює більш глибокий та насичений досвід споживання медіа, дозволяючи людям занурюватися у світ цікавих відкриттів та вражень без зайвих зусиль [6].

Колаборативна фільтрація представляє з себе один з найефективніших методів побудови рекомендаційних систем, який широко застосовується в сучасних платформах. Він заснований на аналізі схожості уподобань користувачів та їхньої попередньої взаємодії з елементами контенту з метою передбачення їхньої майбутньої взаємодії [7].

Суть полягає в тому, щоб на основі досвіду користувачів у минулому, таких як їхні вибори, покупки або перегляди, і подібних рішень інших користувачів, створювати модель рекомендацій. При здійсненні користувачами схожих дії у минулому, такі як вибір однакового телесеріалу, існує велика ймовірність, що вони зроблять подібний вибір і в майбутньому.

Таким чином, система рекомендацій може розуміти, які користувачі мають схожі смаки і інтереси у контенті, і рекомендувати їм елементи, які були популярні або сподобалися іншим користувачам з подібними уподобаннями.

Існують різноманітні методи колаборативної фільтрації.

1. Модель Neural Collaborative Filtering представляє собою нейронну мережу, яка використовується для колаборативного фільтрування на основі взаємодії користувача та елемента. У цій моделі використовується факторизація матриць з точки зору нелінійності (Рисунок 1.1). Для реалізації цієї моделі у TensorFlow приймається послідовність пар ідентифікаторів користувача та ідентифікаторів елемента на вході. Потім ці дані окремо подаються на факторизацію матриць, де виконується множення вкладень, а також у мережу багатошарових перцептронів (MLP).

Виходи факторизації матриць та мережі MLP потім комбінуються та подаються на вхід у щільний шар, який передбачає, чи ймовірно взаємодіяти вхідному користувачу з вхідним елементом.

Цей підхід дозволяє моделі враховувати складні залежності між користувачами та елементами, використовуючи нейронні мережі для здійснення прогнозів взаємодії.

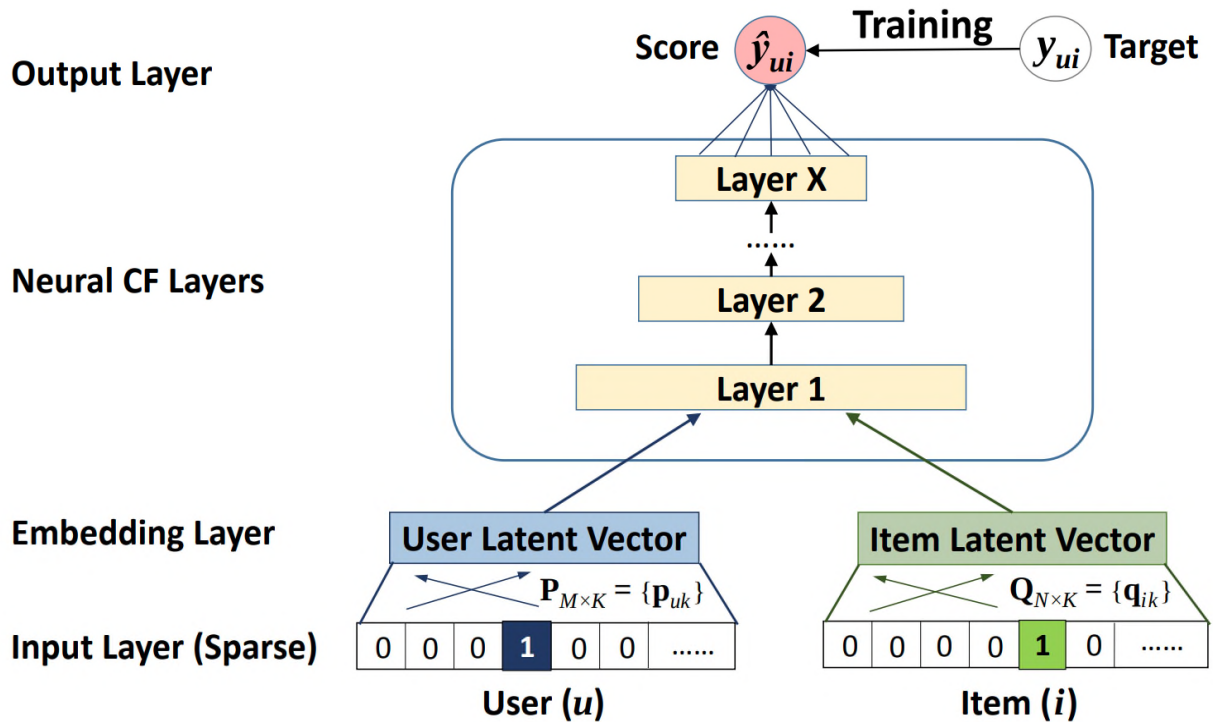


Рисунок 1.1 – Модель NCF [8]

2. У контексті колаборативного фільтрування, нейронна мережа автоенкодера реконструює вхідний шар на виході, використовуючи представлення, отримане у прихованому шарі. У цьому випадку, автоенкодер навчається створювати нелінійне представлення матриці користувач-елемент та відтворює її, визначаючи відсутні значення.

Варіаційний автоенкодер для колаборативного фільтрування на базі прискореної графічної обробки NVIDIA представляє оптимізовану реалізацію архітектури. Модель VAE-CF це нейронна мережа, що забезпечує колаборативне фільтрування на основі взаємодії користувача та елемента. Навчальні дані для цієї моделі складаються з пар ідентифікаторів користувача-елемента для кожної взаємодії.

Модель складається із енкодера та декодера. Енкодер перетворює вхідний вектор, що містить взаємодії для певного користувача, у n-вимірний варіаційний розподіл. Це латентне представлення подається на вхід до декодера, який також є прямопрохідною мережею зі схожою структурою до енкодера. В результаті отримується вектор ймовірностей взаємодії з елементами для певного користувача.

Модель включає в себе дві основні компоненти, такі як енкодер та декодер (Рисунок 1.2). Енкодер відповідає за перетворення вхідного вектора, який містить взаємодії для конкретного користувача, у n-вимірний варіаційний розподіл, який можна сприймати як латентне представлення. Отримане латентне представлення подається на вхід до декодера.

Декодер також є прямо прохідною мережею, що має подібну структуру до енкодера. Після проходження через декодер отримується вектор ймовірностей взаємодії з елементами для конкретного користувача, що є результатом роботи моделі.

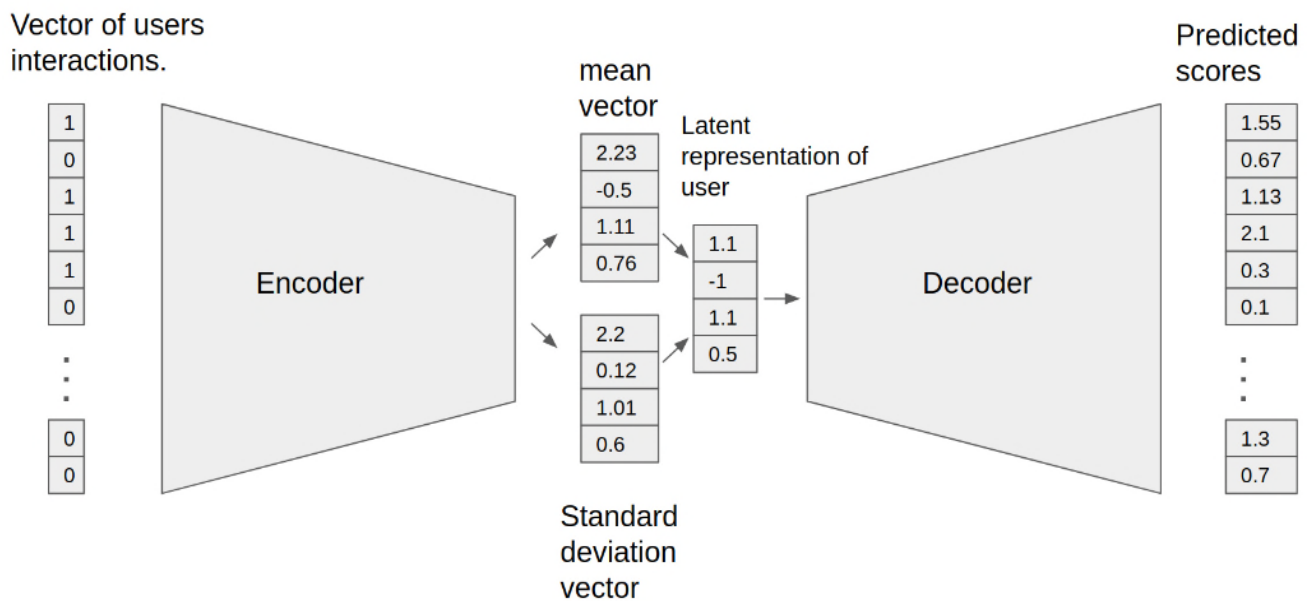


Рисунок 1.2 – Модель VAE-CF [9]

Такий підхід дозволяє ефективно враховувати складності взаємодії між користувачами та елементами, створюючи зручну та інформативну модель для аналізу та передбачення взаємодій.

3. Алгоритм колаборативного фільтрування на основі підходу k-найближчих сусідів (k-NN) виявляється одним із найпоширеніших підходів у системах рекомендацій (Рисунок 1.3). Основна ідея колаборативного фільтрування полягає в передбаченні вподобань користувача шляхом ідентифікації інших користувачів з подібними вподобаннями та використання їхніх оцінок для рекомендацій.

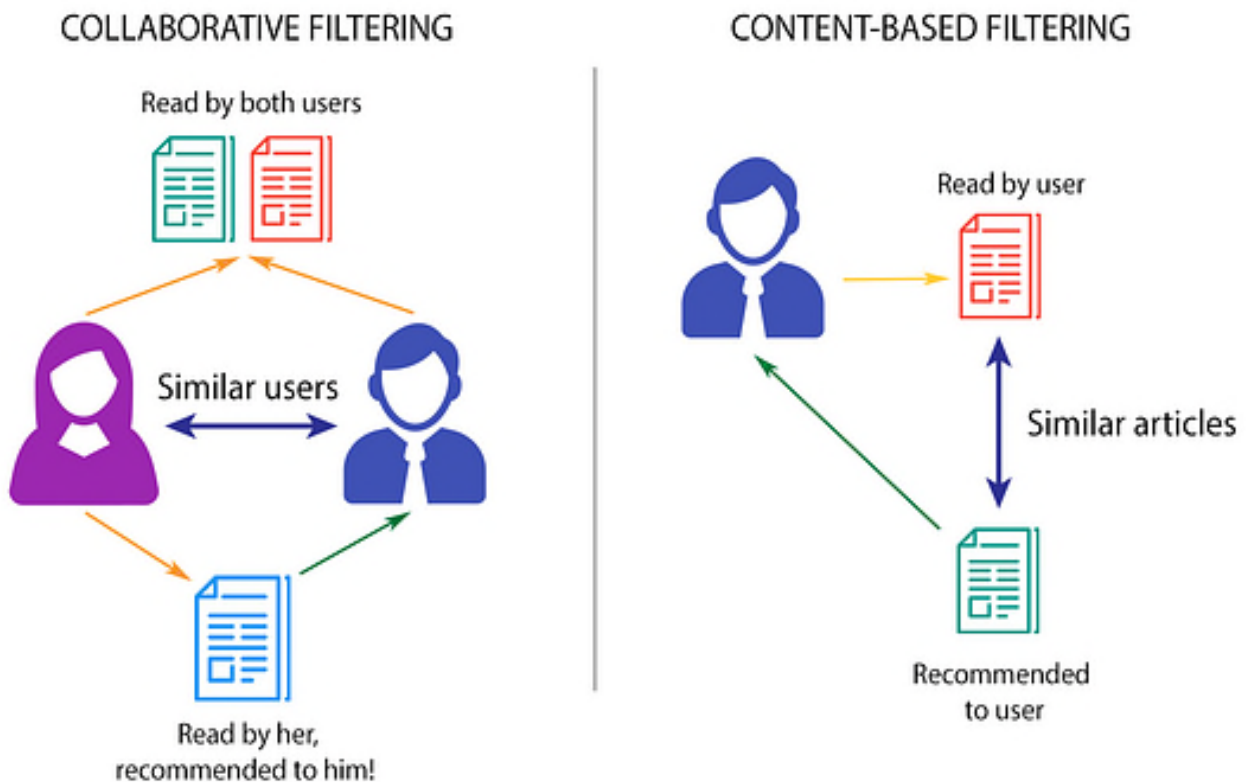


Рисунок 1.3 – Приклад Алгоритм спільної фільтрації на основі k-NN [10]

Алгоритм k-NN вимірює схожість між користувачами за допомогою метрики відстані і потім ідентифікує k користувачів, найбільш схожих на цільового користувача. За допомогою оцінок цих схожих користувачів обчислюється зважений середній рейтинг для кожного елемента. Ваги цього

середнього рейтингу визначаються за допомогою показників схожості між цільовим користувачем і кожним з k сусідів. Прогнозований рейтинг для елемента представляє собою зважений середній рейтинг k сусідів [11].

Крім того, важливо враховувати особливості даних та контексту застосування перед вибором параметра k . Оскільки, у випадку збалансованих даних та стабільного контексту використання, може бути доцільним обрати менше значення k для забезпечення швидкої роботи алгоритму, як у той же час, у складних або змінюваних середовищах, де рівновага між користувачами та елементами може змінюватися, варто розглядати більші значення k для досягнення стійкості та надійності рекомендацій.

4. Алгоритм Matrix Factorization з використанням підходу градієнтного спуску (Gradient Descent) є одним з ключових підходів у колаборативної фільтрації, який використовується для виявлення неявних факторів, що впливають на взаємодію між користувачами та елементами. Підхід Matrix Factorization включає в себе розкладання матриці взаємодій користувач-елемент на дві менші матриці, які представляють користувачів і елементи відповідно [12].

Ідея матричної факторизації полягає в тому, щоб побудувати низько рангову матрицю, наближену до матриці оцінок користувачів, яка може вплинути на рекомендації, надані користувачам. Цей процес дозволяє виявити суттєві зв'язки та закономірності в даних, які можуть бути використані для покращення якості рекомендацій.

Існують різні підходи, доступні для факторизації матриці.

Декомпозиція сингулярного значення розкладає матрицю на три інші матриці, які представляють лінійно незалежні вектори, що утворюють базис для даних. Вона широко використовується в колаборативній фільтрації через його ефективність та здатність виявляти структуру даних.

Імовірнісна матрична факторизація базується на імовірнісному підході до факторизації матриці, де параметри моделі оцінюються з використанням методів

статистичного навчання. Це дозволяє враховувати невизначеність у даних та покращує точність прогнозів.

Факторизація невід'ємної матриці вимагає, щоб всі елементи матриць факторизації були невід'ємними. Це може допомогти виявити більш локальні та частинні закономірності в даних, оскільки не допускається компенсація позитивних та негативних вкладів.

Факторизація Байєсівської імовірнісної матриці (поєднує імовірнісний підхід з байєсівською статистикою для оцінки параметрів моделі. Це дозволяє краще обробляти дані з високим рівнем шуму та невизначеності.

Тензорна матрична факторизація розширює концепцію матричної факторизації на багатовимірні тензори, що дозволяє враховувати додаткові виміри в дані, такі як час або різні аспекти взаємодії користувачів та елементів.

Ці матриці можуть бути використані для передбачення невідомих взаємодій, що є ключовим аспектом колаборативної фільтрації. Кожен з цих реалізацій має свої переваги та недоліки, і вибір найбільш підходящого залежить від конкретних характеристик та потреб даних.

У процесі навчання, алгоритм використовує підхід градієнтного спуску для мінімізації похибки передбачень. Цей процес включає регулярне оновлення параметрів матриць факторизації, щоб зменшити розбіжність між передбаченнями та фактичними даними. Цей підхід дозволяє ефективно здійснювати персоналізовані рекомендації, враховуючи схожість у вподобаннях між користувачами та елементами.

Цей алгоритм Matrix Factorization з використанням підходу градієнтного спуску (Gradient Descent) є надзвичайно важливим доповненням до класичних реалізацій колаборативної фільтрації. Його значення полягає в тому, що він забезпечує велику гнучкість у налаштуванні параметрів, що дозволяє адаптувати модель до конкретних потреб і особливостей даних. Це особливо важливо в умовах, коли дані можуть бути дуже різноманітними та нелінійними.

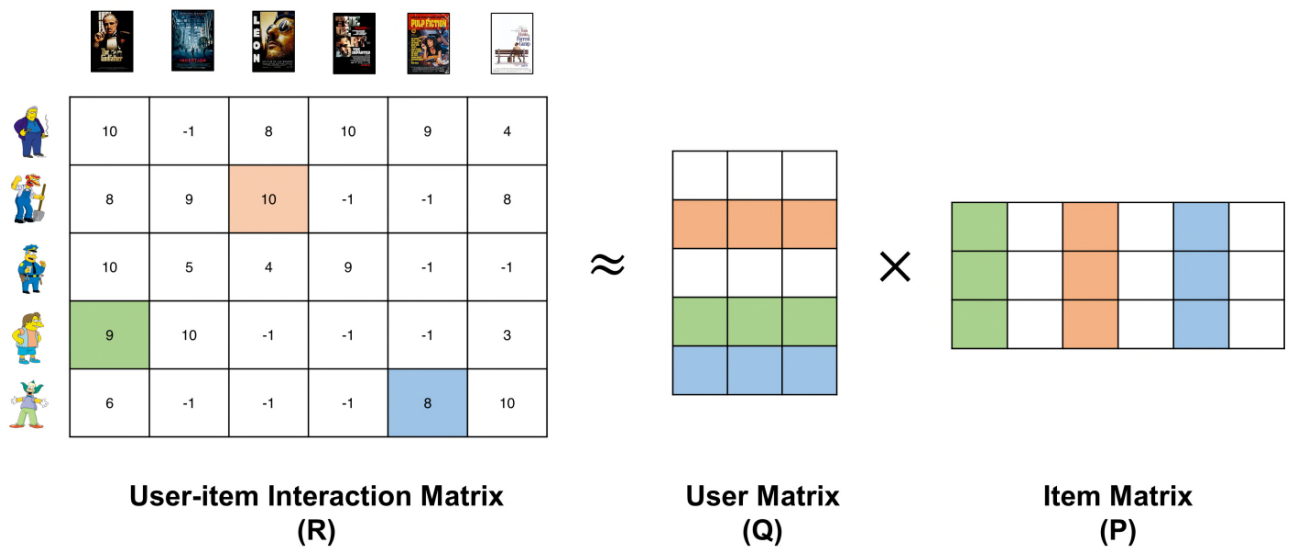


Рисунок 1.4 – Модель Matrix Factorization з використанням Gradient Descent [13]

Крім того, алгоритм Matrix Factorization легко інтегрується з іншими реалізаціями машинного навчання, такими як нейронні мережі або автоенкодері. Таке поєднання дозволяє створювати більш комплексні та точні моделі, які можуть враховувати не тільки лінійні, але й нелінійні залежності у даних. Це призводить до значного покращення точності рекомендацій, що є критично важливим для ефективності систем персоналізованого контент-підбору.

1.3 Аналіз існуючих вебсистем з колаборативною фільтрацією

Всесвітньо відомі стрімінгові платформи здебільшого покладаються на колаборативну фільтрацію для забезпечення користувачам персоналізованих рекомендацій телесеріалів. Для цього аналізується історія переглядів та рейтинги користувачів, а також інші дані, щоб точно визначити, які серіали можуть бути до вподоби, або до аналізу також додається інформація про покупки, демографічні дані та інші фактори.

Враховуючи успішність використання колаборативної фільтрації платформами світових компаній-гігантів, важливо проаналізувати та оцінити існуючі проекти з використанням реалізацій у вебсистемах.

Netflix представляє з себе сервіс потокового медіа, який надає доступ до широкої бібліотеки фільмів, серіалів, телесеріалів, документальних фільмів та спеціальних програм [14]. Замість того, щоб купувати або орендувати контент окремо, користувач платитиме щомісячну абонентську плату, яка надає необмежений доступ до всього контенту (Рисунок 1.4).

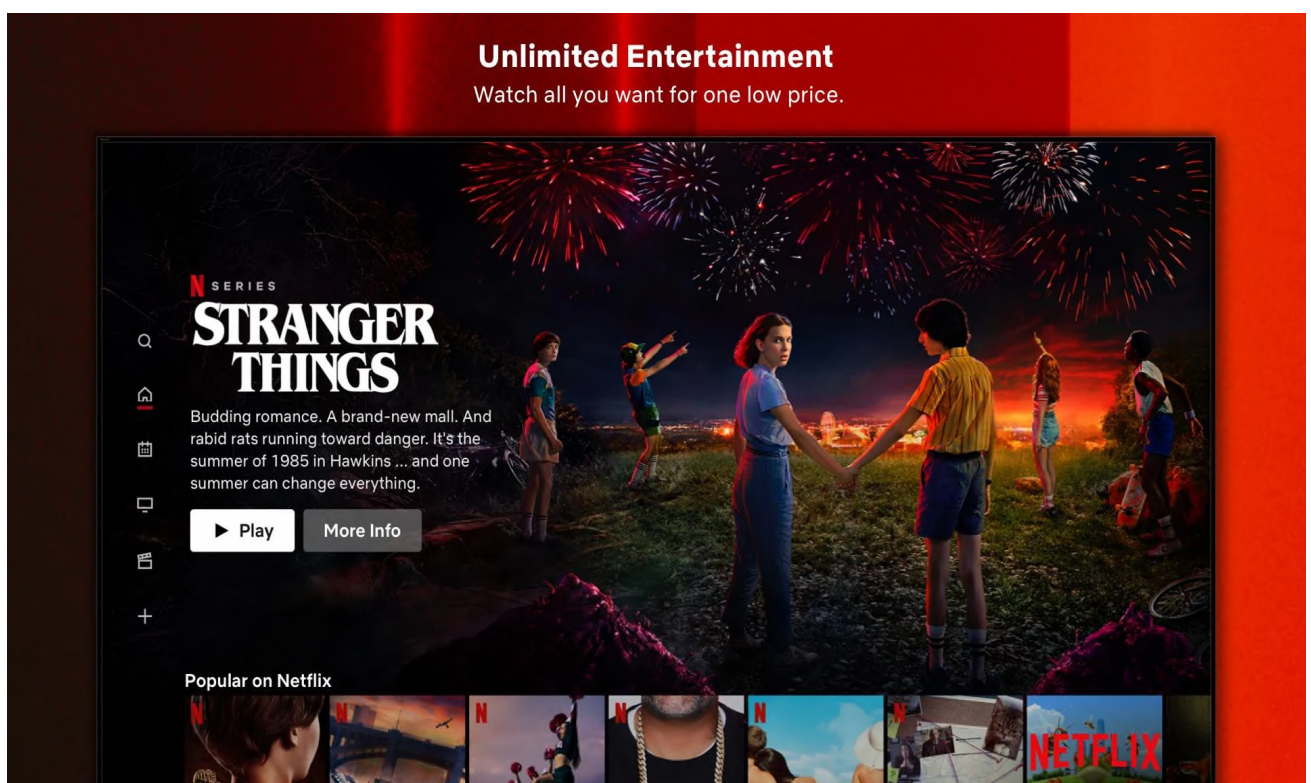


Рисунок 1.4 – Netflix сервіс потокового відео [15]

Завдяки колаборативній фільтрації, Netflix значно покращив задоволеність користувачів своїми рекомендаціями телесеріалів. Колаборативна фільтрація допомагає Netflix утримувати понад 75% своїх підписників. Система аналізує історію переглядів користувачів, їхні рейтинги та інші дані, щоб визначити, які серіали їм можуть сподобатися.

До переваг Netflix можна виділити те, що він володіє одним із найбільших у світі наборів даних про поведінку користувачів, що дає

можливість навчати та вдосконалювати свої алгоритми колаборативної фільтрації з більшою точністю. Також використовується додатково інші підходи, такі як контентний аналіз та машинне навчання, щоб пропонувати користувачам максимально персоналізовані рекомендації. До цього ж варто зазначити, що рекомендації Netflix здебільшого точні та релевантні, що призводить до високого рівня задоволеності користувачів.

Серед недоліків є те, що алгоритми можуть бути упередженими, якщо вони навчені на даних, які не є репрезентативними для всієї аудиторії користувачів, а також те, що рекомендації не завжди є прозорими для користувачів, що може призвести до недовіри [16].

Amazon Prime Video є сервісом потокового відео від відомого гіганта онлайн-торгівлі – Amazon. Сервіс пропонує широкий спектр фільмів, телесеріалів, документальних фільмів, а також спеціальних програм. На відміну від Netflix, доступ до Prime Video може бути частиною більш широкого преміум пакета підписки Amazon Prime (Рисунок 1.5).

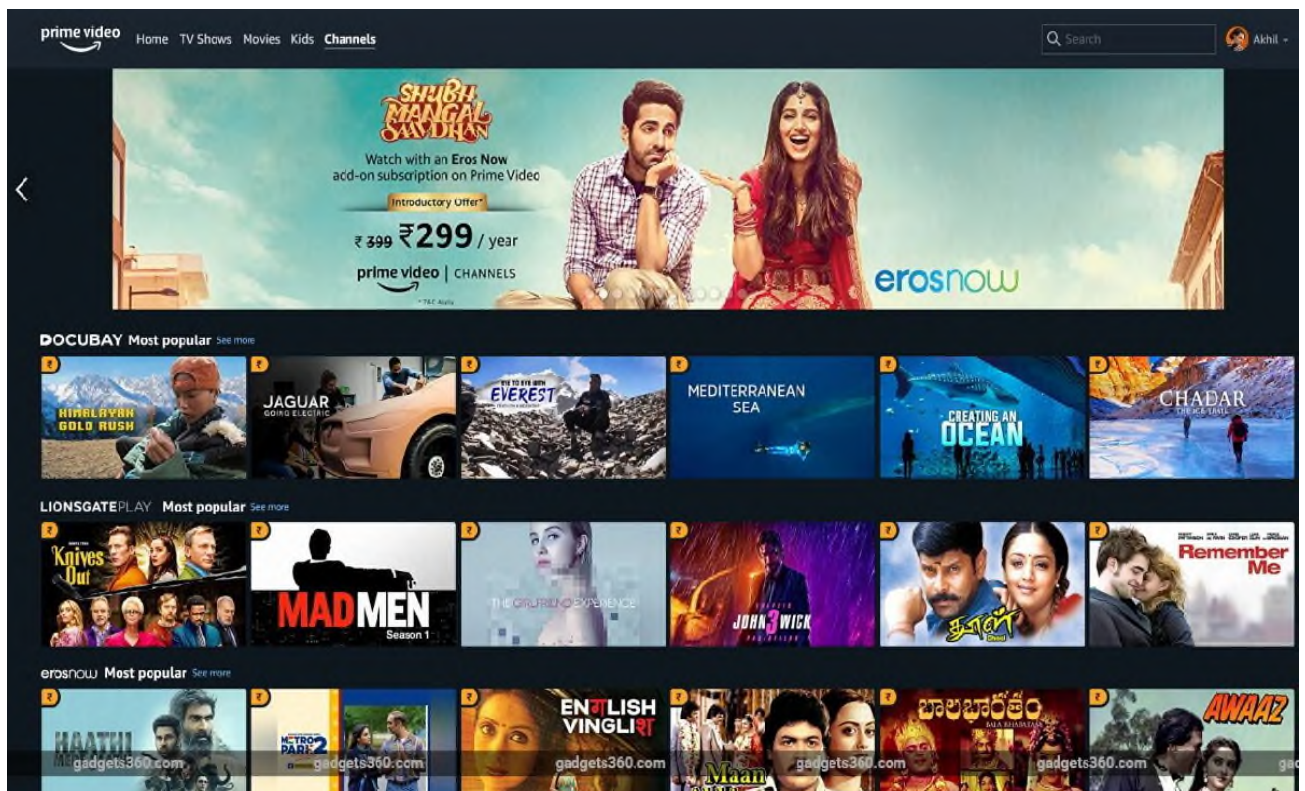


Рисунок 1.5 – Amazon Prime Video сервіс потокового відео [17]

У Amazon Prime Video колаборативна фільтрація використовується для рекомендації контенту, який з великою ймовірністю сподобається користувачам. Це призвело до значного зростання кількості годин перегляду контенту на платформі. Система враховує не лише історію переглядів користувачів, але й їхні покупки, демографічні дані та інші фактори.

До переваг даного сервісу можна віднести оригінальний контент, який є ексклюзивом для даної платформи, так і сторонній контент. Це дає можливість використовувати колаборативну фільтрацію для рекомендацій більш різноманітного контенту. Також Amazon може використовувати дані про покупки та інші дії користувачів на своїх платформах, щоб покращити рекомендації Prime Video. Крім цього, завдяки колаборації з іншими сервісами Amazon, сервіс має доступ до більшої кількості даних про користувачів, що значно покращує якість рекомендацій.

До недоліків можна додати, що алгоритми колаборативної фільтрації надмірно фокусуються на рекомендаціях контенту, який користувачі можуть купити, а не на тому, що їм дійсно цікаво [18].

Hulu позиціонується як сервіс потокового відео, що належить Walt Disney Company [19]. Він відомий тим, що пропонує широкий вибір телевізійних шоу, включаючи як найновіші епізоди популярних серіалів, так і класичні телевізійні програми (Рисунок 1.6). Окрім телебачення, Hulu також пропонує фільми та власні оригінальні програми.

Hulu використовує колаборативну фільтрацію у поєднанні з іншими реалізаціями, щоб пропонувати користувачам більш персоналізовані та релевантні рекомендації телесеріалів. Це допомогло сервісу збільшити свою частку ринку та конкурувати з такими гігантами, як Netflix.

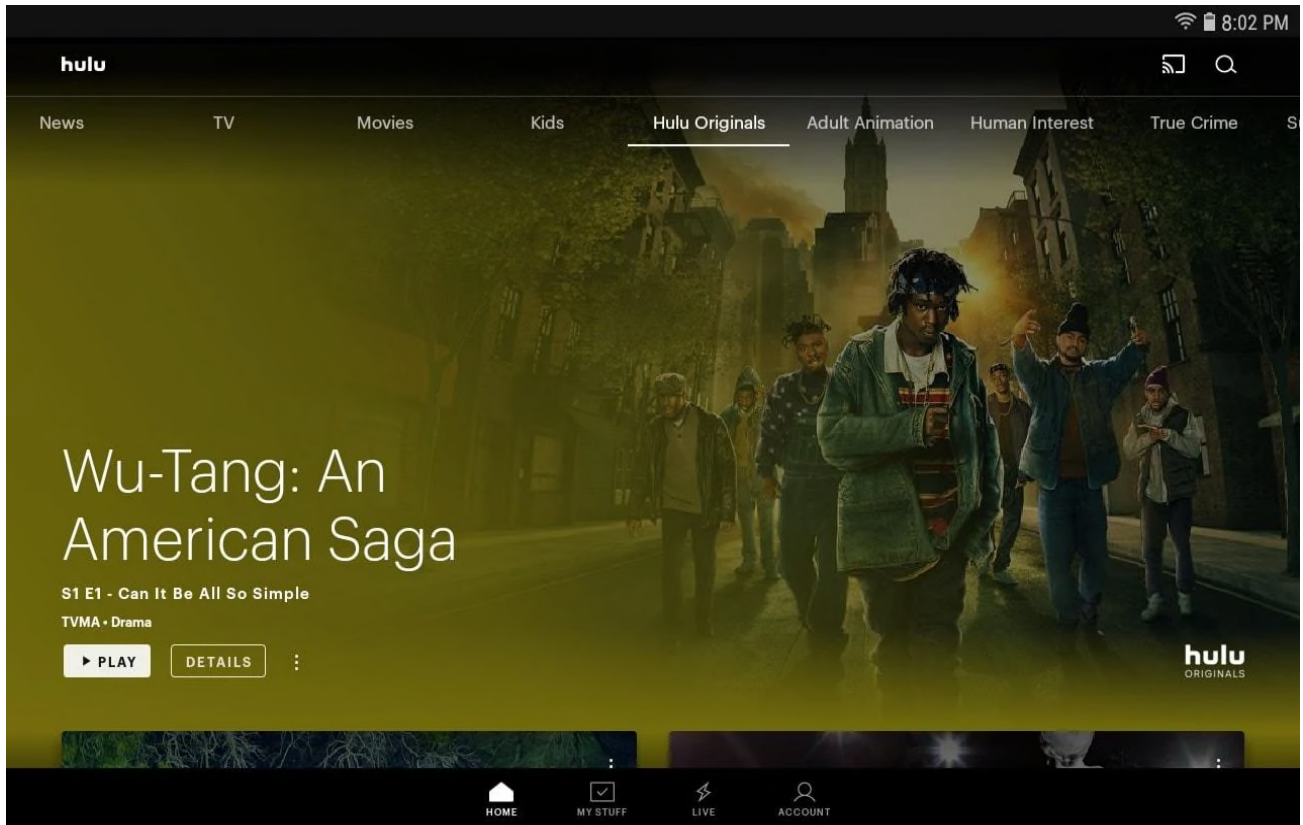


Рисунок 1.6 – Hulu сервіс потокового відео [20]

Перевагами Hulu можна вважати інтеграцію з традиційним телебаченням, що дає їм доступ до даних про те, які телепередачі дивляться користувачі, а також поєднувати із своїми оригінальними телесеріалами. Це дає їм можливість використовувати для рекомендацій контенту, який справді цікавий їхній аудиторії. Hulu також пропонує різні плани підписки.

Серед недоліків сервісу є такі, як значно менший набір даних, ніж у Netflix або Prime Video, що може обмежити точність та релевантність їхніх рекомендацій. Крім цього відсутня можливість для налаштування своїх рекомендацій, що може ускладнити отримання бажаних рекомендацій, а також через обмеженість малої кількості жанрів, рекомендації можуть бути викривленими [21].

Загалом, після огляду декількох систем підбору телесеріалів, де використана колаборативна фільтрація, можна зрозуміти, що у кожній з реалізацій фільтрації для рекомендацій є свої недоліки. З упевненістю можна

стверджувати, що для заданих цілей колаборативна фільтрація є достатньо ефективним вирішенням проблеми.

1.4 Мета та завдання кваліфікаційної роботи

Мета розробки рекомендаційної системи полягає у покращенні сервісу сайтів для перегляду телесеріалів шляхом формування персоналізованих рекомендацій щодо перегляду за допомогою колаборативної фільтрації. Ця система спрямована на полегшення пошуку цікавого контенту для користувачів та підвищення їх загального задоволення від використання вебсистеми.

Для досягнення поставленої мети необхідно реалізувати виконання наступних задач:

- провести аналіз систем рекомендацій на основі колаборативної фільтрації;
- розробити рекомендаційну систему, яка використовує колаборативну фільтрацію для формування персоналізованих рекомендацій;
- розробити функціональну та інформаційну структуру вебсистеми перегляду телесеріалів;
- розробити програмну реалізацію вебсистеми підбору телесеріалів з використанням рекомендаційної системи на основі колаборативної фільтрації;
- виконати тестування рекомендаційної системи підбору телесеріалів.

Виконання цих етапів дозволить вирішити поставлену мету та забезпечить зручність для користувачів у отриманні рекомендацій щодо перегляду телесеріалів.

Розділ 2 Рекомендаційна система на основі колаборативної фільтрації

2.1 Загальна схема рекомендаційної системи

При врахуванні всіх визначених цілей та завдань, пов'язаних із створенням рекомендаційної системи підбору телесеріалів з використанням методу колаборативної фільтрації для вебсистем, яка на даний момент розробляється, важливо ретельно прорахувати основні принципи функціонування цієї системи.

Однією з ключових ідей є концепція того, що користувачі з подібними смаками мають тенденцію поділяти схожі вподобання щодо телесеріалів. Таким чином, розглядаючи дані про те, які серіали вже були переглянуті та оцінені користувачами, система може вести прогноз щодо того, які телесеріали можуть зацікавити їх у майбутньому.

Розробка такої системи вимагає вивчення широкого спектру факторів, які впливають на вибір телесеріалів. Вона повинна враховувати не лише історію переглядів та оцінок користувачів, а й їхні особисті уподобання та інші параметри, щоб надати максимально релевантні рекомендації для кожного індивідуального користувача.

Однак, ефективність рекомендаційної системи також залежить від якості зібраних даних та використаних алгоритмів. Важливо врахувати навіть найдрібніші деталі, такі як структура бази даних, методи збору та обробки інформації, щоб система працювала швидко та ефективно [22].

На рисунку 2.1 зображена загальна схема роботи системи рекомендації телесеріалів на основі методу колаборативної фільтрації. На ній описано покроковий процес, за допомогою якого система генерує персоналізовані рекомендації.

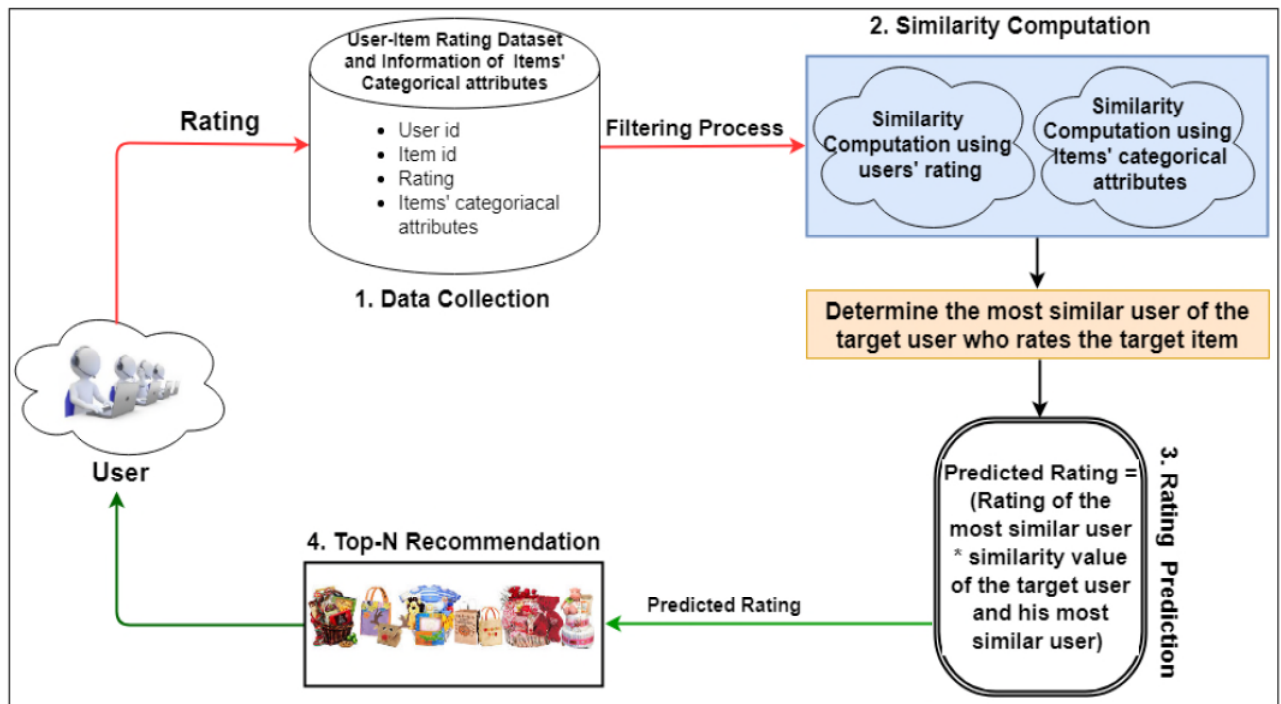


Рисунок 2.1 – Загальна схема рекомендаційної системи [23]

Рекомендаційна система на основі колаборативної фільтрації складається з наступних кроків:

1) Збір даних включає в себе збір інформації про перегляди та рейтинги телесеріалів, а також інших його параметрів, таких як жанри, назви, ключові слова тощо.

2) Обчислення подібності є етапом, де розраховуються коефіцієнти схожості між користувачами на основі їх переглядів, а також схожість телесеріалів і його атрибутів. Існує кілька підходів до розрахунку подібності, таких як косинусна схожість, Евклідова відстань та Пірсоновський коефіцієнт.

3) Прогнозування рейтингу є етапом, де модель обчислює, яким чином користувачі собі будуть обирати до перегляду телесеріали, якщо вони ще не були переглянуті. Для цього використовуються обчислені раніше коефіцієнти схожості та історія переглядів користувачів.

4) На базі прогнозованих рейтингів телесеріалів до перегляду формується список рекомендацій, де найкращі рекомендації вказують на те, які телесеріали набрали найбільшу кількість пріоритетних рекомендацій, що означає, які з них користувачам слід розглянути для перегляду.

Хоча метод колаборативної фільтрації є надійним та ефективним методом рекомендацій для різних сфер, таких як фільми, музика, новини та електронна комерція, вона не може визначити кореляції в дуже розріджених наборах даних. Ця розрідженість знижує точність рекомендаційних систем на основі колаборативної фільтрації через те, що не вистачає спільно оцінених елементів для ефективного обчислення схожості.

Для вирішення цієї проблеми можна використовувати міру схожості SeM на основі коефіцієнта Батташар'ї. Однак він не є ідеальним, коли вектори рейтингів користувачів не мають перетину, що обмежує його ефективність.

Алгоритми колаборативної фільтрації можуть бути упередженими, якщо вони навчені на даних, які не є репрезентативними для всієї аудиторії користувачів. Це може призвести до того, що певним групам користувачів рекомендуватимуться лише певні типи контенту, що може обмежити їхнє знайомство з іншими жанрами та ідеями.

У процесі роботи вебсистеми буде використана модель машинного навчання, яка на основі визначених параметрів забезпечуватиме ідентифікацію найбільш відповідних рекомендацій для користувача.

2.2 Опис кроків формування рекомендацій у системі

Рекомендаційна система підбору телесеріалів поділяється на етап ініціалізації та обробки даних, а також на етап, коли вона може надавати рекомендації.

При першій ініціалізації вебсистеми виділяється певний період на збирання даних про користувачів та їхні уподобання. Під час цього етапу рекомендаційна система збирає інформацію про переглянуті телесеріали, оцінки, які користувачі надають телесеріалам, а також інші релевантні дані, необхідні для колаборативної фільтрації. Ці дані включають інформацію про жанри, акторський склад, ключові слова, тривалість епізодів, час виходу та інші характеристики телесеріалів.

Система також використовує алгоритми для розподілу користувачів на групи на основі їхніх уподобань, що дозволяє зменшити обсяг обрахунків та підвищити ефективність рекомендацій. Цей процес може включати кластеризацію користувачів за допомогою різноманітня алгоритмів, або використовувати методи аналізу схожості, такі як косинусна схожість або Евклідова відстань.

Після збору достатньої кількості даних про користувачів та їхні уподобання, вебсистема починає надавати персоналізовані рекомендації телесеріалам. На цьому етапі система використовує модель прогнозування, яка враховує історію перегляду користувача та історію перегляду схожих користувачів. Модель може базуватися на алгоритмах колаборативної фільтрації, таких як Matrix Factorization, Neighborhood-based methods або Hybrid methods, які об'єднують колаборативні та контент-базовані методи.

Для надання рекомендацій телесеріалів користувачам, система використовує алгоритм знаходження схожості між користувачами. Цей алгоритм дозволяє визначити, наскільки схожі два користувачі за їхніми уподобаннями та інтересами, і на основі цієї схожості надавати рекомендації телесеріалів.

Процес рекомендацій відбувається у формі циклу, що дозволяє системі постійно вдосконалюватись. Це означає, що після кожної рекомендації система аналізує відгук користувача на цю рекомендацію. Наприклад, вона визначає, чи переглянув користувач рекомендований телесеріал, і який рейтинг він надав цьому телесеріалу.

На основі отриманих відгуків система коригує свою модель для покращення майбутніх рекомендацій. Це дозволяє системі стати все більш точною та ефективною у наданні рекомендацій, які відповідають уподобанням користувачів.

Схема формування рекомендацій телесеріалів на основі колаборативної фільтрації зображена на рисунку 2.2.

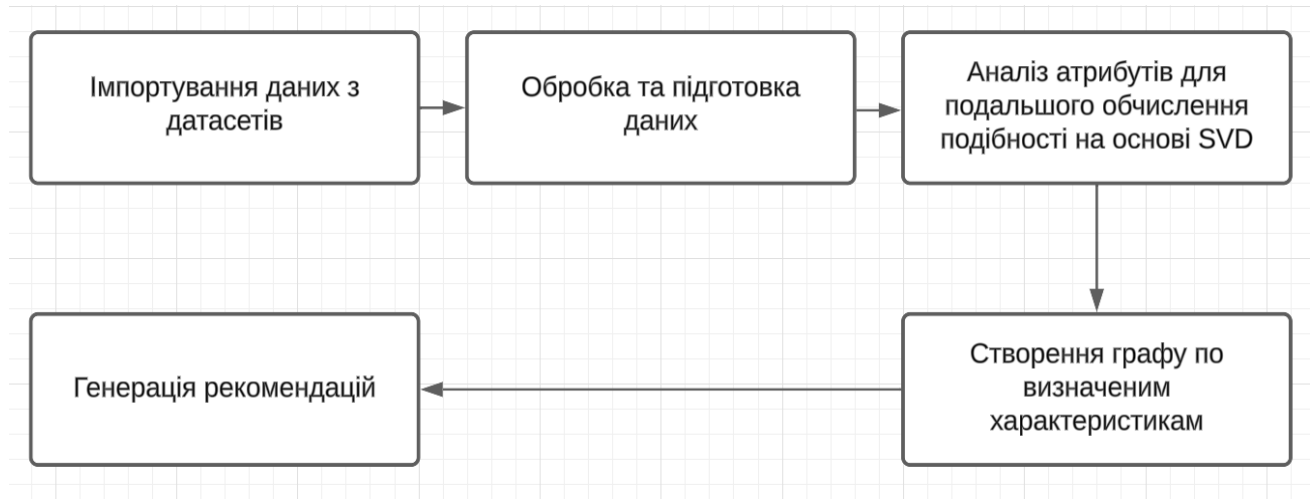


Рисунок 2.2 – Схема формування рекомендацій телесеріалів на основі колаборативної фільтрації

1. Атрибути даних у наборі даних використовуються для системи рекомендацій. Набір даних, який містить інформацію про користувачів та їх взаємодію з контентом, завантажується в базу даних. Ці дані включають ідентифікатори користувачів, ідентифікатори фільмів та телесеріалів, рейтинги, дати переглядів, а також метадані про фільми та телесеріали, такі як жанри, ключові слова, акторський склад та інші релевантні характеристики.

2. Попередня обробка атрибутів даних. На цьому етапі дані очищуються та готуються для подальшого аналізу. Це включає в себе видалення прогалів, нормалізацію даних, заповнення пропущених значень, а також перетворення текстових даних у структурований формат. Також може бути виконана трансформація даних для покращення їхньої придатності для машинного навчання, наприклад, за допомогою кодування частот ключових слів або одночасного кодування.

3. Виконується аналіз жанрів і ключових слів. Цей етап включає в себе вивчення жанрів і ключових слів, пов'язаних з кожним телесеріалом. Для визначення їх значимості виконується факторизація матриці оцінок за допомогою Singular Value Decomposition (SVD). За допомогою цього розраховуються ваги для жанрів і ключових слів, які відображають їхній вплив на вподобання користувачів. Ваги можуть бути розраховані за допомогою різних

методів, таких як Term Frequency-Inverse Document Frequency або з використанням інших статистичних методик.

4. Створення графу. На цьому етапі створюється граф, де вузлами є телесеріали та ключові слова, а ребра відображають зв'язки між ними з відповідними вагами. Граф дозволяє візуалізувати та аналізувати взаємозв'язки між телесеріалами та їх характеристиками. Ваги ребр можуть відображати силу зв'язку між телесеріалами та ключовими словами, що допомагає виявити схожість між телесеріалами та прогнозувати вподобання користувачів.

5. Застосування колаборативного фільтрування до значень даних. На цьому етапі застосовується підхід колаборативного фільтрування до значень даних, який включає в себе аналіз взаємодії користувачів з телесеріалами та виявлення схожих користувачів або телесеріалів. Це дозволяє генерувати персоналізовані рекомендації для кожного користувача на основі їхніх уподобань та поведінки. На останньому етапі генеруються очікувані рекомендації, які відображаються користувачеві у вигляді списку телесеріалів, які вони можуть цікавитися.

Ці етапи дозволяють створити ефективну систему рекомендацій, яка враховує різноманітні аспекти даних та взаємозв'язки між користувачами та контентом.

2.3 Обчислення схожості вподобань користувачів

Система рекомендацій на основі графів пропонується як новий підхід методу колаборативної фільтрації. Він зберігає оцінені дані вмісту користувача в структурі графа в поєднанні з алгоритмами графів і різними методами рекомендацій. Порівняно з існуючою системою рекомендацій, система рекомендацій на основі графів має дві основні переваги, а саме масштабованість і різноманітність моделювання зв'язків [24].

Відмінність між системою рекомендацій на основі графів та існуючою моделлю рекомендацій полягає в ефективності зберігання даних. Графова

система рекомендацій не будує розріджену матрицю для передбачення схожості технік колаборативного фільтрування. Натомість інформація про вподобання користувача та значення обчислення схожості накопичуються та зберігаються безпосередньо в структурі графа. Використовуючи алгоритм k -найближчих сусідів (k -NN), можна створити граф, використовуючи лише певну кількість користувачів у порядку найбільшої схожості. Передбачення вподобань здійснюються з меншими обсягами даних та з меншою обчислювальною потужністю (Рисунок 2.3).

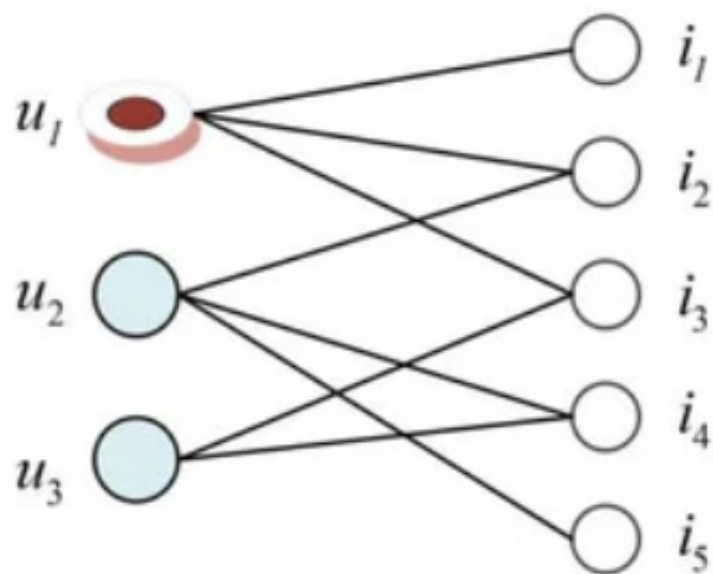


Рисунок 2.3 – Графік взаємодії користувача з елементом [25]

Завдяки цій особливості немає потреби виконувати обчислення схожості під час реального часу рекомендацій, оскільки час обчислення не збільшується пропорційно до кількості користувачів або елементів. Крім того, на відміну від класичної реалізації рекомендацій, система рекомендацій на основі графів може використовувати розріджені дані про вподобання користувачів для знаходження зв'язку між елементами, тим самим вирішуючи проблеми масштабованості та недостатності даних.

Система рекомендацій на основі графів також має перевагу в моделюванні різних відносин між даними. Сучасна система рекомендацій

поступово розвивається в персоналізовану систему рекомендацій для індивідуальних користувачів.

Необхідно візуалізувати дані таким чином, щоб модель могла навчатися від кожного користувача і диверсифікувати логіку або політику рекомендацій. Для того щоб диверсифікувати моделі за допомогою традиційного підходу, необхідно створювати моделі для кожного складного набору даних та системи.

Проте база даних на основі графів ефективно керує логікою і реляційною моделлю з легкістю при рекомендації контенту для кожного користувача, одночасно встановлюючи зв'язки між моделями контенту. Наприклад, моделюючи відносини між контентом, такими як телесеріал – жанр або телесеріал – актор і режисер, можна побудувати граф однієї категорії телесеріалів та надавати користувачам різні рекомендаційні сервіси (Рисунок 2.4).

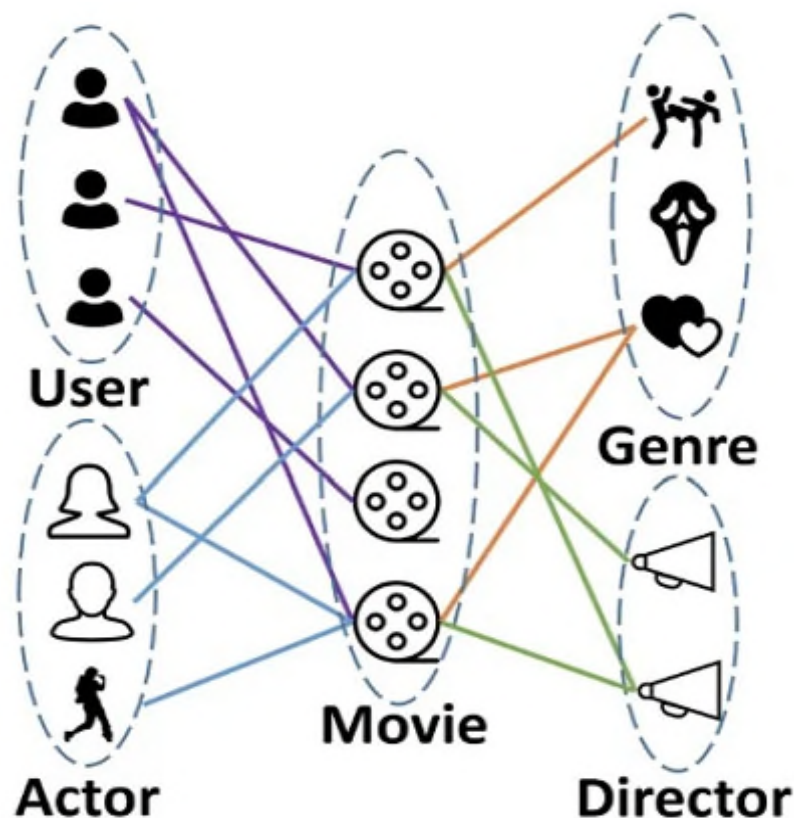


Рисунок 2.4 – Граф із вузлами, що представляють користувача, жанр, телесеріал, актора та режисера [26]

Всього є чотири основні процеси, які складають графові системи рекомендацій.

На першому етапі створюється структура графа, де вузли відповідають користувачам або елементам контенту, а ребра вказують на зв'язки між ними. Вага ребер може відображати силу зв'язку між вузлами, наприклад, рейтинги, перегляди або інші форми взаємодії користувачів з контентом. Це забезпечує базову архітектуру, яка відображає всі відношення та взаємодії в системі.

За допомогою отриманих даних про структуру графа та взаємозв'язки між вузлами, система може пропонувати рекомендації користувачам на основі їхніх уподобань та схожості з іншими користувачами або елементами контенту. Наприклад, якщо два користувачі мають схожі уподобання, і один з них переглянув новий телесеріал, система може рекомендувати цей телесеріал іншому користувачеві. Це дозволяє користувачам швидше знаходити новий та цікавий для них контент, що значно покращує їхній досвід користування платформою.

Крім того, система може враховувати не лише прямі взаємозв'язки між користувачами, але й складніші схеми взаємодії в мережі. Наприклад, якщо користувачі А і В обидва взаємодіють з користувачем С, система може вважати, що між користувачами А і В також є потенційний інтерес до подібного контенту, навіть якщо вони безпосередньо не взаємодіяли. Це дозволяє створювати більш точні та персоналізовані рекомендації, які відповідають глибинним зв'язкам у структурі графа.

Для оцінки ефективності колаборативної фільтрації на основі графів використовуються різні метрики, одними з яких є точність, повнота, середнє зважене відхилення або середнє відхилення. Ці метрики допомагають визначити, наскільки добре система рекомендацій працює та виявити можливі напрямки покращення.

Точність вимірює, наскільки вірогідно система правильно передбачає вподобання користувача [27]. Це відношення кількості правильно передбачених

позитивних рекомендацій до загальної кількості наданих рекомендацій. Формула для обчислення точності:

$$\text{Precision} = \frac{\text{Relevant retrieved instances}}{\text{All retrieved instances}} \quad (2.1)$$

Relevant retrived instances означає кількість правильно передбачених позитивних рекомендацій, all retrived instances – кількість всіх передбачених рекомендацій.

Повнота оцінює, яку частку з усіх можливих позитивних рекомендацій система змогла запропонувати. Це відношення кількості правильно передбачених позитивних рекомендацій до загальної кількості фактичних позитивних випадків. Формула для обчислення повноти:

$$\text{Recall} = \frac{\text{Relevant retrieved instances}}{\text{All relevant instances}} \quad (2.2)$$

Корінь середньоквадратичної помилки (RMSE) вказує на різницю між рекомендованими системою та дійсно вподобаними користувачем елементами контенту. Обчислюється як корінь середнього значення квадрату різниць між передбаченими та фактичними значеннями:

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2} \quad (2.3)$$

Значення n вказує на загальну кількість передбачень, Y_i – фактичне значення для i -го елемента, \hat{Y}_i – передбачене значення для i -го елемента.

Додатково, для підвищення ефективності системи, можуть бути використані й інші підходи, такі як врахування часу взаємодії користувачів з контентом, аналіз текстової інформації з відгуків або коментарів, а також інтеграція з соціальними мережами, де користувачі діляться своїми вподобаннями. Врахування часу, протягом якого користувач взаємодіє з контентом, дозволяє системі рекомендацій бути більш точною у передбаченні вподобань користувача.

Використання текстової інформації з відгуків та коментарів допомагає зрозуміти вподобання користувачів на більш глибокому рівні. Техніки обробки природної мови дозволяють виділяти ключові слова та фрази, які характеризують вподобання користувачів. Інтеграція з соціальними мережами дозволяє враховувати взаємодії та вподобання користувачів у соціальному контексті. Це допомагає системі рекомендацій бути більш адаптивною та враховувати широкий спектр факторів.

Система рекомендацій на основі графів підходить для використання як база знань для системи, яка працює з великою кількістю користувачів і контенту. Однак, під час зберігання даних через з'єднання різного контенту, необхідно зберігати унікальні значення різних відкритих даних з ретельним розглядом. Рекомендується зберігати дані великої ємності в розподіленому сховищі разом із графовим моделюванням.

2.4 Функціональна структура інформаційної системи для колаборативної фільтрації

Наступним кроком є представлення діаграми всіх варіантів використання, яка охоплює загальні напрямки застосування вебсистеми з інтегрованою колаборативною фільтрацією (Рисунок 2.6). Ця діаграма надає розуміння того, як користувачі можуть взаємодіяти із рекомендаційною системою та як вона може відповідати їхнім потребам у підборі та рекомендаціях телесеріалів.

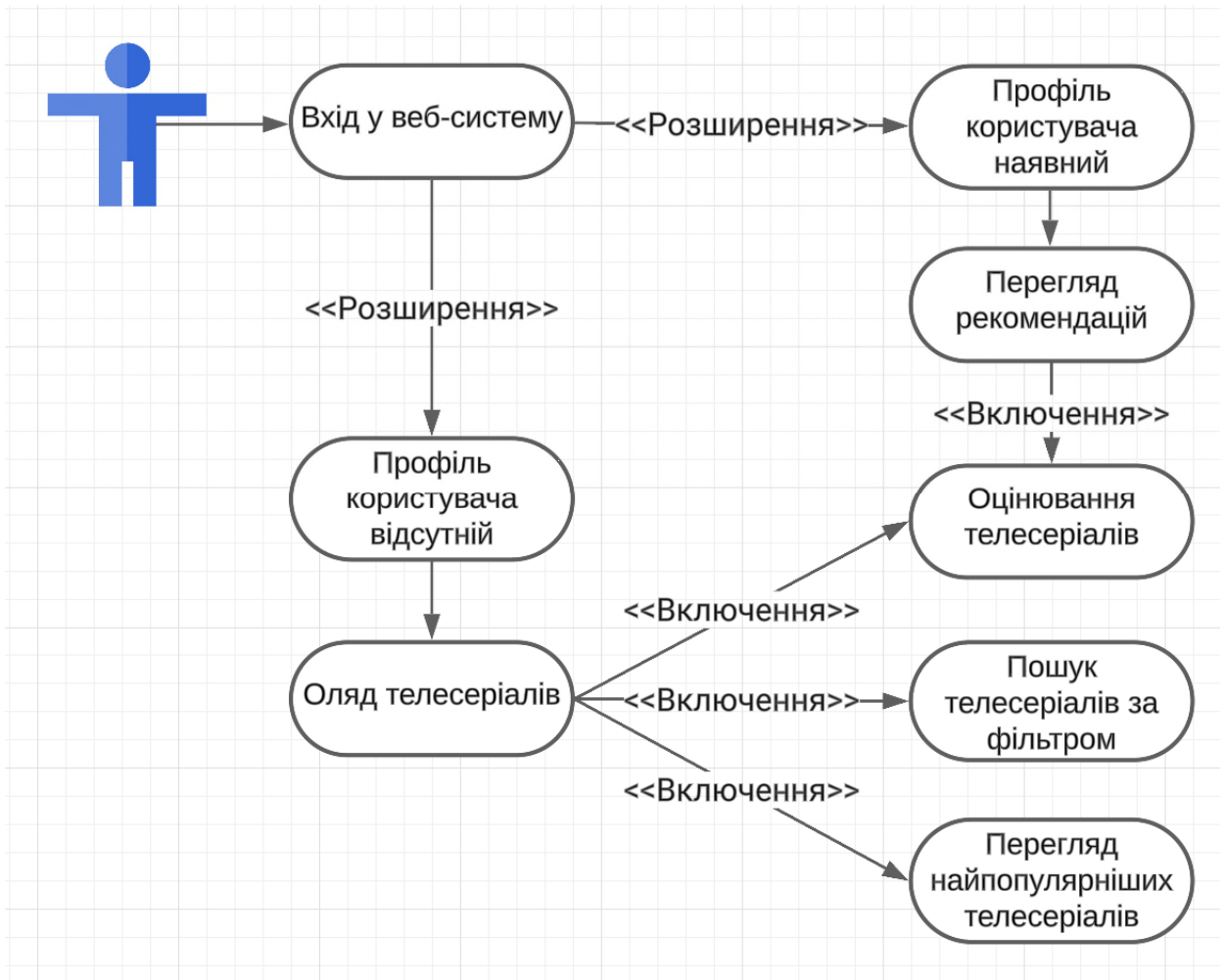


Рисунок 2.6 – Діаграма варіантів використання

Далі наведено сценарії прецедентів використання вебсистеми:

Назва першого прецеденту: Пошук нового телесеріалу.

Діюча особа: Користувач.

Мета: Знайти новий телесеріал, який відповідає смаку користувача та зацікавить його.

Передумови: Користувач має уявлення про те, які жанри та теми телесеріалів його цікавлять, а також він новий у вебсистемі.

Головна послідовність:

1. Вхід у вебсистему.
2. Перегляд списку доступних телесеріалів. Оскільки користувач новий в системі, йому спочатку пропонуються всі популярні телесеріали.

3. Використання фільтрів пошуку, відсортувавши за жанром, рейтингом та популярністю, щоб звужити список телесеріалів, які відповідають інтересам користувача.

4. Відбір телесеріалів, які зацікавили користувача, та їх оцінювання.

Назва другого прецеденту: Перегляд персоналізованих рекомендацій телесеріалів

Діюча особа: Користувач

Мета: Знайти нові телесеріали, які, ймовірно, сподобаються користувачеві, на основі історії переглядів.

Передумови: Вебсистемі відомі вподобання інших користувачів, які перетинаються із вподобаннями діючого користувача, у якого є профіль користувача.

Головна послідовність:

1. Вхід у вебсистему.

2. Перегляд персоналізованих рекомендацій телесеріалів, які видала вебсистема, котрі ґрунтуються на його вподобаннях. Оскільки користувач не новий в системі, йому одразу пропонується список рекомендацій.

3. Користувач читає описи телесеріалів, переглядає їхню інформацію та оцінює запропоновані.

Після огляду розроблених схем, діаграм станів та сценаріїв використання, можна чітко зрозуміти, як функціонує система рекомендацій в контексті вебсистеми.

2.5 Проектна архітектура рекомендаційної системи та взаємозв'язок компонентів

Згідно з рекомендаційною системою підбору телесеріалів з використанням колаборативної фільтрації для вебсистем, була розроблена відповідна структура інформаційної системи (Рисунок 2.7). Ця система включає

в себе кілька ключових модулів, кожен з яких виконує свою специфічну функцію для забезпечення ефективності та точності рекомендацій.

Зокрема, система складається з таких модулів: модуль збору даних, модуль обробки та зберігання даних, модуль аналізу даних, модуль генерації рекомендацій та модуль візуалізації та інтерфейсу користувача.

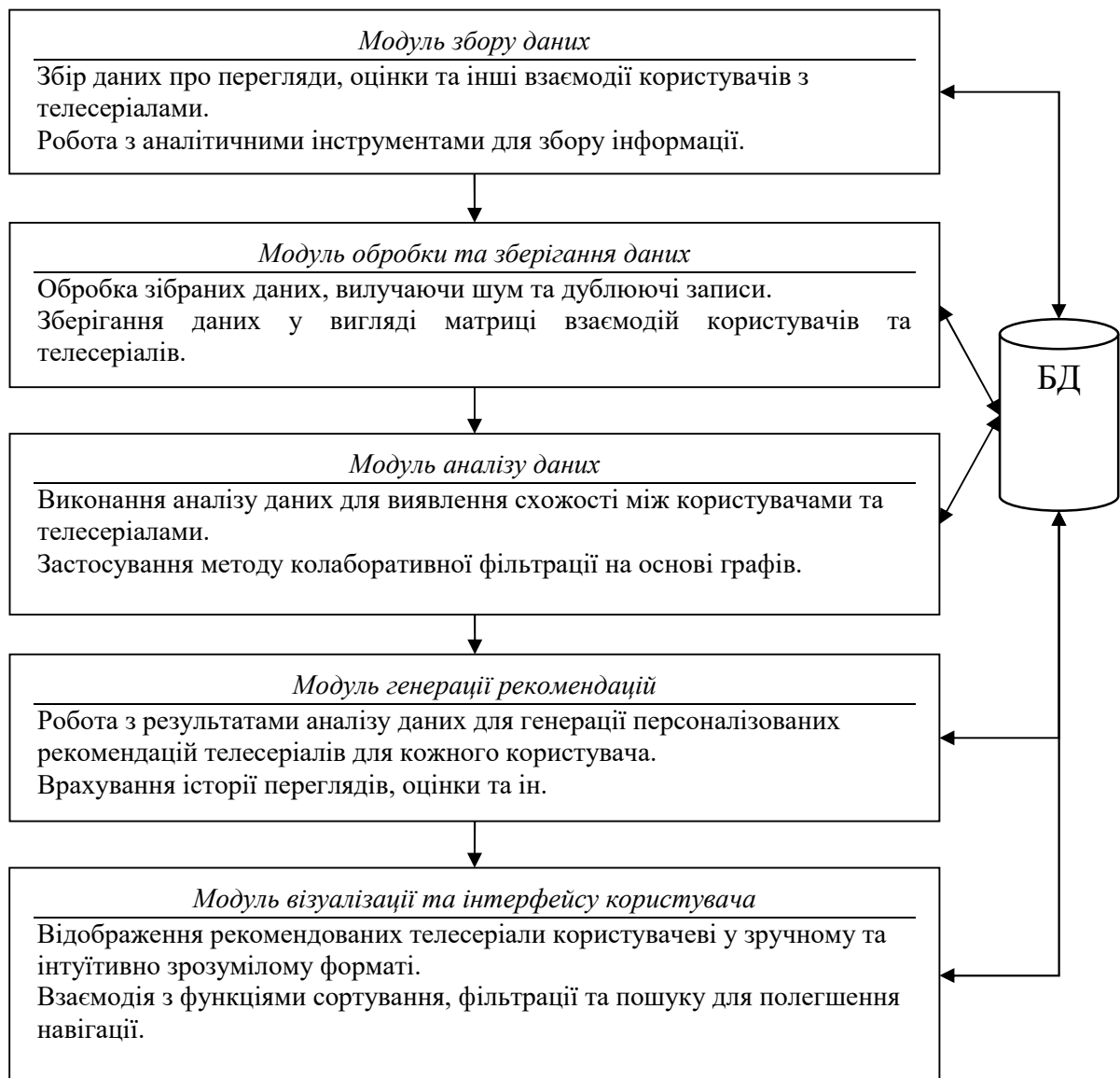


Рисунок 2.7 – Схема інформаційної системи вебсистеми з колаборативною фільтрацією

Модуль обробки та зберігання даних виконує очищення та структурування зібраних даних перед їх зберіганням. Це включає видалення шуму, дублюючих записів та інших неточностей, які можуть вплинути на якість

рекомендацій. Очищені дані зберігаються у вигляді матриці взаємодій між користувачами та телесеріалами, що дозволяє легко виконувати подальші аналітичні операції.

Модуль аналізу даних використовує передові алгоритми для виявлення схожості між користувачами та телесеріалами. Він застосовує такі методи, як колаборативна фільтрація на основі графів та матричне розкладання, для генерації точних рекомендацій. Цей модуль аналізує велику кількість даних, щоб виявити приховані патерни та зв'язки між різними елементами, забезпечуючи високу релевантність результатів.

Модуль генерації рекомендацій використовує результати аналізу даних для створення персоналізованих рекомендацій телесеріалів для кожного користувача. Враховуючи історію переглядів, оцінки та інші фактори, цей модуль пропонує користувачеві найбільш підходящі серіали, які відповідають його вподобанням. Цей процес включає не лише пряму рекомендацію, але й можливість врахування трендових телесеріалів та новинок, які можуть бути цікавими для користувача.

Модуль візуалізації та інтерфейсу користувача відповідає за представлення рекомендацій користувачам у зручному та зрозумілому форматі. Він надає інтерфейс для взаємодії з користувачем, дозволяючи шукати, сортувати та фільтрувати телесеріали для полегшення навігації. Цей модуль забезпечує інтуїтивно зрозумілий дизайн та високий рівень юзабіліті, що сприяє позитивному користувацькому досвіду.

Отже, розроблена інформаційна система з використанням колаборативної фільтрації для підбору телесеріалів є комплексним рішенням, яке інтегрує збір, обробку, аналіз та візуалізацію даних, щоб надавати користувачам персоналізовані та релевантні рекомендації. Це сприяє покращенню користувацького досвіду та підвищує залученість аудиторії на платформі.

2.6 Інформаційна структура рекомендаційної системи

Для розробки рекомендаційної системи підбору телесеріалів з використанням колаборативної фільтрації для вебсистем, необхідно обрати відповідний датасет, що характеризує дану сферу. Це необхідно для того, щоб навчити модель нейронної мережі, яка на основі цього навчання зможе пропонувати точні та релевантні рекомендації.

Для колаборативної фільтрації найкраще використовувати датасет, який містить інформацію про користувачів та їхні перегляди телесеріалів. Такий датасет дозволить моделі вивчити зв'язки між користувачами та їхніми переглядами, що забезпечить можливість пропонувати рекомендації на основі схожості між користувачами. Наприклад, якщо двоє користувачів мають схожі історії переглядів, система зможе запропонувати одному з них ті телесеріали, які вже були переглянуті іншим.

Одним з популярних датасетів для цього завдання є MovieLens, який містить інформацію про оцінки користувачів для фільмів [28]. Проте для телесеріалів, які можуть мати різну структуру даних, можуть використовуватися інші датасети, такі як «The Movies Dataset», який містить інформацію про метадані, рейтинги, ключові слова та інші атрибути телесеріалів. Цей датасет доступний на Kaggle [29].

«The Movies Dataset» на Kaggle містить інформацію про фільми та телесеріали, зібрані з бази даних IMDb та The Movie DB. Він складається з ряду файлів, що охоплюють різні аспекти про фільми та телесеріали, такі як метадані, рейтинги, ключові слова та посилання.

Також наведено гістограми для кращого розуміння розподілу інформації, що міститься в датасеті.

Перша гістограма демонструє розподіл рейтингів фільмів, які містяться в датасеті «The Movies Dataset» на Kaggle. Ця гістограма має горизонтальну вісь, на якій відображаються рейтинги фільмів у діапазоні від 0 до 10. Вертикальна вісь гістограми показує відсоткову кількість фільмів, які мають відповідний

рейтинг (Рисунок 2.8). Додатково до гістограми, на цьому графіку також представлена крива щільності розподілу, яка базується на підході kernel density estimate. Ця крива допомагає краще зрозуміти загальну форму розподілу рейтингів фільмів у датасеті.

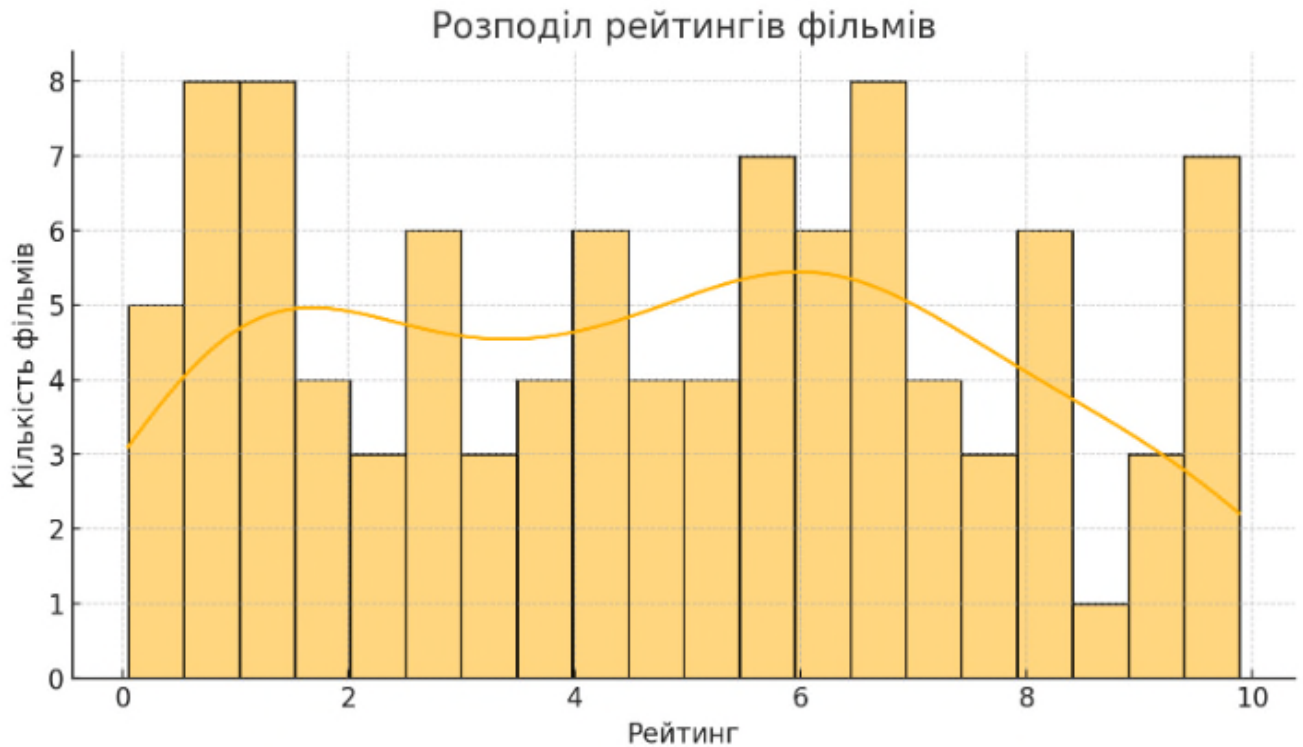


Рисунок 2.8 – Гістограма розподілу рейтингів

Наступний графік відображає кількість фільмів у кожному з представлених жанрів. На цьому графіку представлено по горизонтальній осі жанри, а по вертикальній осі – кількість фільмів, що належать до кожного жанру (Рисунок 2.9). Слід зазначити, що один фільм може належати до кількох жанрів, тому загальна кількість фільмів у жанрах може перевищувати кількість фільмів у датасеті.

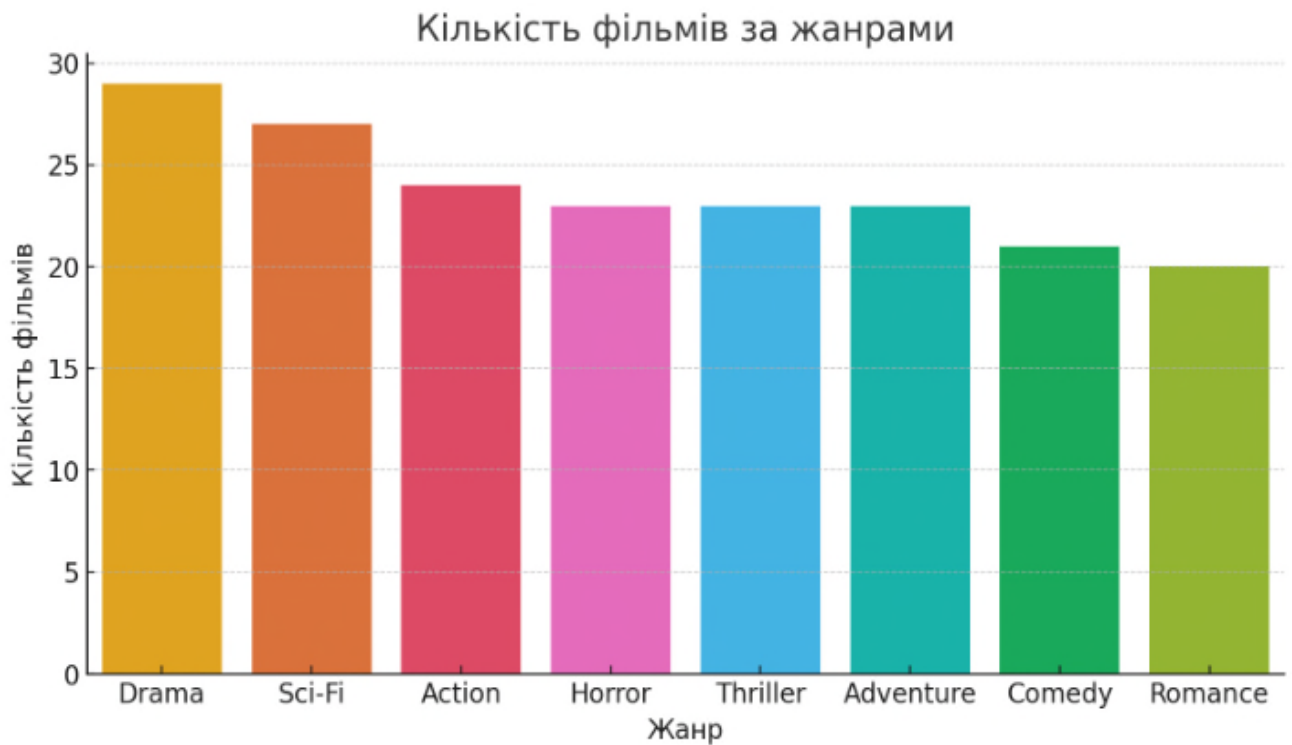


Рисунок 2.9 – Гістограма кількості фільмів за жанрами

Для розробки рекомендаційної системи цей датасет можна адаптувати під телесеріали, виділивши відповідні дані та налаштувавши модель для роботи з ними. Це дозволить створити ефективну рекомендаційну систему, яка враховуватиме вподобання користувачів та забезпечить персоналізовані рекомендації.

Далі наведено файли з датасету, у вигляді таблиць, які будуть використовуватись для рекомендацій.

Таблиця «movies_metadata» містить метадані про фільми, а також про телесеріали, такі як назва, рік випуску, тривалість, рейтинг IMDb, кількість голосів, бюджет, дохід, мова оригіналу, країна, компанія-продюсер, жанри та ключові слова (Таблиця 2.1). Ці дані дозволяють аналізувати популярні фільми за рік, аналіз бюджетів та доходів, аналіз ключових слів, аналіз акторської спільноти та інші статистичні дані.

Таблиця 2.1 – Атрибути таблиці «movies_metadata»

| № п/п | Назва | Тип даних | Опис |
|-------|-------|-----------|------|
|-------|-------|-----------|------|

| | | | |
|-----|-----------------------|---------|--|
| 1. | adult | boolean | Вказує, чи фільм для дорослих. |
| 2. | belongs_to_collection | json | Інформація про колекцію, до якої фільм належить. |
| 3. | budget | bigint | Бюджет фільму. |
| 4. | genres | json | Жанри фільму. |
| 5. | homepage | varchar | Домашня сторінка фільму. |
| 6. | id | bigint | Унікальний ідентифікатор фільму. |
| 7. | imdb_id | varchar | Унікальний ідентифікатор на IMDb. |
| 8. | original_language | varchar | Оригінальна мова фільму. |
| 9. | original_title | varchar | Оригінальна назва фільму. |
| 10. | overview | text | Короткий опис фільму. |
| 11. | popularity | float | Популярність фільму. |
| 12. | poster_path | varchar | Шлях до постера фільму. |
| 13. | production_companies | json | Компанії-продюсер фільму. |
| 14. | production_countries | json | Країни-продюсер фільму. |
| 15. | release_date | date | Дата виходу фільму. |
| 16. | revenue | bigint | Дохід фільму. |
| 17. | runtime | float | Тривалість фільму. |
| 18. | spoken_languages | json | Мови, які використовуються в фільмі. |
| 19. | status | varchar | Статус фільму. |
| 20. | tagline | text | Слоган фільму. |
| 21. | title | varchar | Назва фільму. |
| 22. | video | boolean | Вказує, чи є відео для фільму. |
| 23. | vote_average | float | Середня оцінка фільму. |
| 24. | vote_count | bigint | Кількість голосів для фільму. |

Таблиця «ratings» містить рейтинги користувачів для всіх фільмів. Ці дані дозволяють аналізувати популярність фільмів, які мають високі оцінки, а також визначати популярність фільмів за часом (Таблиця 2.2).

Таблиця 2.2 – Атрибути таблиці «ratings»

| № п/п | Назва | Тип даних | Опис |
|-------|-----------|-----------|---|
| 1. | userId | bigint | Унікальний ідентифікатор користувача. |
| 2. | movieId | bigint | Унікальний ідентифікатор фільму. |
| 3. | rating | float | Оцінка, яку користувач поставив фільму. |
| 4. | timestamp | bigint | Час, коли користувач поставив оцінку. |

Таблиця «keywords» містить ключові слова для кожного фільму. Ці дані дозволяють аналізувати популярні слова, які використовуються в фільмах, а також визначати популярні фільми за ключовими словами (Таблиця 2.3).

Таблиця 2.3 – Атрибути таблиці «keywords»

| № п/п | Назва | Тип даних | Опис |
|-------|----------|-----------|-------------------------------------|
| 1. | Id | bigint | Унікальний ідентифікатор фільму. |
| 2. | keywords | json | Ключові слова, пов'язані з фільмом. |

Таблиця «links_small» містить посилання на невеликі фотографії для кожного фільму та їхні оцінки. Ці дані дозволяють аналізувати популярні фільми за оцінками користувачів (Таблиця 2.4).

Таблиця 2.4 – Атрибути таблиці «links_small»

| № п/п | Назва | Тип даних | Опис |
|-------|---------|-----------|-----------------------------------|
| 1. | movieId | bigint | Унікальний ідентифікатор фільму. |
| 2. | imdbId | bigint | Унікальний ідентифікатор на IMDb. |
| 3. | tmdbId | bigint | Унікальний ідентифікатор на TMDb. |

Датасет «The Movies Dataset» містить інформацію про близько 45 тисяч фільмів та телесеріалів, з якими можна провести комплексний аналіз, включаючи метадані, рейтинги, ключові слова, посилання та акторську спільноту. Найстаріший фільм у датасеті був випущений в 1896 році. Середній рейтинг фільмів становить 6.44/10, середня тривалість близько 1 години, середній бюджет близько 15 мільйонів доларів, середній дохід близько 11 мільйонів доларів. У датасеті представлено близько 20 жанрів фільмів, фільми зібрано з 65 різних країн, у датасеті представлено близько 2000 компаній-продюсерів та близько 7000 акторів.

Також датасет містить інформацію про 26 мільйонів оцінок, зроблених 270 тисяч користувачів, середня кількість ключових слів в одному фільмі становить 5, датасет містить посилання на фотографії для близько 9000 фільмів, середня кількість акторів в одному фільмі становить 10, середня кількість сценаріїв, композиторів, керівників та директорів в одному фільмі становить 3, а середня кількість фільмів з однаковими акторами становить 7.

Ці статистики дають глибоке розуміння про структуру та зміст датасету «The Movies Dataset», який дозволяє здійснювати комплексні аналітичні дослідження.

2.7 Висновки до розділу 2

У цьому розділі було розглянуто загальну схему роботи системи підбору телесеріалів за допомогою колаборативної фільтрації. Також було розкрито метод формування рекомендацій телесеріалів та продемонстровано діаграми варіантів використання цієї системи.

Під час аналізу різних підходів до створення методу рекомендацій, було обрано систему рекомендацій на основі графів. Крім цього, було визначено взаємозв'язок модулів вебсистеми

Також розглянуто та обрано датасет «The Movies Dataset» із вебресурсу Kaggle. Цей датасет містить інформацію про фільми та телесеріали, зібрану з бази даних IMDb та The Movie DB, що дозволяє використовувати його для створення рекомендацій телесеріалів на основі колаборативної фільтрації. Для кращого розуміння інформації даного датасету також було наведено гістограми розподілу контенту по рейтингам і жанрам.

Розділ 3 Програмна реалізація вебсистеми підбору телесеріалів

3.1 Вебтехнології та засоби розробки рекомендаційної системи

Згідно із поставленими задачами та вимогами, які були зазначені вище, обрано наступні технології розробки для реалізації рекомендаційної системи підбору телесеріалів з використанням колаборативної фільтрації для вебсистем.

Для реалізації вебсистеми ми використовували WebStorm як основне середовище розробки. Даний інструмент обрано через його підтримку JS, а також TypeScript, додатково наявна інтеграція з системами контролю версій, зручний дебагінг та автоматичне завершення коду, що значно полегшує процес розробки. Зважаючи на важливість типізації для зниження кількості помилок та покращення структури проекту, було прийнято рішення використовувати TypeScript. Така надбудова над JS додає статичну типізацію, яка дозволяє виявляти помилки ще на етапі компіляції [30].

Для забезпечення швидкої збірки проекту обрано Vite. Цей інструмент забезпечує миттєвий запуск серверу розробки, гаряче перезавантаження модулів та підтримку сучасних ES-модулів, що робить процес розробки більш ефективним та зручним. Однією з головних переваг Vite є його здатність миттєво запускати сервер розробки, що дозволяє розробникам відразу бачити зміни в коді без затримок. Це особливо корисно на ранніх етапах розробки, коли швидкість ітерацій є критично важливою. Крім того, гаряче перезавантаження модулів дозволяє оновлювати тільки ті частини коду, які змінилися, без необхідності повного перезавантаження сторінки, що значно прискорює процес розробки [31].

Щоб стилізувати інтерфейс вебсистеми, був використаний SCSS, який є надбудовою над CSS. Це розширення забезпечує підтримку додаткових можливостей, які значно спрощують написання стилів та покращують структуру CSS-коду [32].

Він дозволяє використовувати змінні для зберігання кольорів, шрифтів, розмірів та інших властивостей, які часто повторюються в проекті. Це дозволяє

легко управляти дизайном, оскільки зміна значення змінної в одному місці автоматично відбивається на всіх місцях, де ця змінна використовується. Також підтримує вкладеність селекторів, що дозволяє писати CSS-код в контексті батьківських елементів. Це спрощує структуру коду та зменшує кількість повторюваних селекторів.

Для реалізації колаборативної фільтрації обрано кілька ключових бібліотек для Python, які дозволяють ефективно працювати з даними та виконувати складні обчислення.

Основним інструментом для роботи з числовими даними є NumPy, який забезпечує високу продуктивність у виконанні числових обчислень та має велику кількість функцій для обробки масивів даних. NumPy підтримує багатовимірні масиви та матриці, а також набір функцій для виконання операцій над цими структурами даних [33]. Ця бібліотека є основою для багатьох інших наукових бібліотек і часто використовується для створення та маніпуляції даними на низькому рівні.

Для роботи з даними використано Pandas, оскільки ця бібліотека дозволяє ефективно аналізувати та обробляти дані, маніпулювати таблицями даних, а також має простий синтаксис для виконання складних операцій [34]. Pandas надає структури даних та операції для маніпуляції з часовими рядами та табличними даними. Вона включає два основних типи структур даних: одновимірний масив та двовимірну таблицю даних, які є дуже гнучкими та можуть зберігати дані різних типів, включаючи числа, рядки та об'єкти.

Pandas інтегрується з іншими науковими бібліотеками, що робить її універсальним інструментом для аналізу даних. Вона також має функції для читання та запису даних у різних форматах, таких як CSV, Excel, SQL бази даних та HDF5 формати. Це дозволяє легко імпортувати та експортувати дані з різних джерел та зберігати результати аналізу.

Для того щоб зберегти результати роботи програми між її запусками, було необхідно використати спеціальну бібліотеку під назвою Pickle. Ця бібліотека є важливою інструкцією в Python, оскільки вона забезпечує процес

серіалізації та десеріалізації різних об'єктів, створених у середовищі Python. Завдяки функціоналу Pickle, можна легко зберігати та завантажувати різноманітні об'єкти, такі як моделі машинного навчання, конфігураційні файли, дані експериментів та інші важливі дані, що дозволяє ефективніше управляти ресурсами та зменшувати час, необхідний для повторного обчислення або налаштування програми.

Кожна з цих технологій була обрана з огляду на специфіку завдань, зокрема розробки рекомендаційної системи для підбору телесеріалів з використанням колаборативної фільтрації. Це забезпечує високу продуктивність, зручність розробки та якість кінцевого продукту, цей підхід є достатньо ефективним та сучасним.

3.2 Реалізація рекомендаційної системи підбору телесеріалів

3.2.1 Підготовка набору даних

Нижче детально розглянемо кожен з етапів реалізації системи, включаючи підготовку даних, навчання моделі, та інтеграцію різних технологій.

Дані завантажуються з CSV-файлів за допомогою Pandas. Датасети містять інформацію про рейтинги користувачів та метадані телесеріалів. Цей етап включає завантаження даних, обробку відсутніх значень, очищення та трансформацію даних, а також об'єднання кількох джерел даних в єдиний набір.

Спершу відбувається завантаження даних з CSV-файлів, використовуючи бібліотеку Pandas. Основні набори даних включають метадані телесеріалів, рейтинги користувачів, ключові слова, пов'язані з телесеріалами, та зв'язки між різними ідентифікаторами телесеріалів.

Метадані телесеріалів містять основну інформацію, таку як назва, жанри та ідентифікатори телесеріалів. Рейтинги користувачів містять інформацію про те, які користувачі оцінили які телесеріали і які оцінки вони поставили. Ключові слова допомагають зрозуміти тематику та основні елементи телесеріалів. Зв'язки між ID дозволяють з'єднати різні бази даних, такі як TMDB та IMDB.

Далі інформація обробляється, а саме відбувається фільтрація по телесеріалам, видалення дублікатів та рядків з відсутніми значеннями, а також перетворення типів даних для подальшої обробки. Наприклад, здійснюється перетворення ідентифікаторів телесеріалів з типу object у тип float, а також зберігається інформація про ключові слова та жанри телесеріалів у вигляді рядків, які можна легко зберегти або обробити далі (Рисунок 3.4), що дозволяє уникнути проблем з типами даних у подальших обчисленнях.

```
def clean_data(rows):
    if len(rows) > 0:
        return [item['name'].lower().strip() for item in rows]
    return rows

movies.keywords = movies.keywords.apply(lambda x: clean_data(ast.literal_eval(x)))
movies.genres = movies.genres.apply(lambda x: clean_data(ast.literal_eval(x)))
```

Рисунок 3.4 – Код відповідний за один із етапів обробки даних

Наступним кроком є об'єднання таблиць, де поєднуються метадані телесеріалів, ключові слова та зв'язки між ID телесеріалів, що дозволяє створити єдиний набір даних з усією необхідною інформацією про телесеріали. Це здійснюється за допомогою функції merge бібліотеки Pandas.

Після цього обробляються текстові поля, такі як жанри і ключові слова. Використовуючи бібліотеку ast, перетворюються рядки в списки, що дозволяє працювати з ними більш ефективно.

Потім об'єднується набір даних з рейтингами користувачів і зберігаємо його для подальшого використання, що дає змогу легше зрозуміти, які телесеріали були оцінені користувачами, і які оцінки вони поставили.

Дані про взаємодії користувачів із телесеріалами зберігаються у таблиці movie_ratings, де кожен рядок містить ідентифікатор користувача userId, ідентифікатор телесеріалу movieId, рейтинг rating та мітку часу timestamp.

Таблиця movie_ratings об'єднана з таблицею телесеріалів, що включає додаткову інформацію про них, такі як жанри та ключові слова.

Далі застосовується темпоральний поділ – сортування даних за міткою часу (timestamp). Це гарантує, що записи будуть упорядковані в хронологічному порядку, починаючи з найстаріших і закінчуючи найновішими.

Після сортування даних за часом, вони розділяються на дві частини: тренувальний набір і тестовий набір. Для цього використовується співвідношення 80:20, де 80% даних йдуть до тренувального набору, а решта 20% – до тестового.

Темпоральний поділ забезпечує збереження природного порядку записів, що важливо для моделювання динамічних процесів. Наприклад, у випадку рекомендаційних систем, користувацькі вподобання можуть змінюватися з часом, і важливо враховувати ці зміни.

У реальних умовах, коли нові дані надходять після тренування моделі, вони завжди з'являються у хронологічному порядку. Темпоральний поділ дозволяє моделі бути більш підготовленою до таких реальних сценаріїв, оскільки тестовий набір імітує майбутні дані.

Модель SVD тренується на тренувальному наборі даних за допомогою бібліотеки Surprise. Після тренування модель зберігається для подальшого використання.

Також варто зазначити, що для обробки та перетворення текстових даних у числові формати застосовується модуль `sklearn.feature_extraction.text` з бібліотеки Scikit-learn. Цей модуль надає потужні інструменти для текстового представлення даних, що є необхідним для аналізу та побудови моделей машинного навчання.

Для ефективної роботи з великими датасетами використовується модуль `tensorflow.python.data.Dataset` з TensorFlow. Цей інструмент забезпечує зручну та ефективну обробку великих обсягів даних, підтримує потокову обробку даних, що є критично важливим для тренування моделей машинного навчання.

3.2.2 Навчання рекомендаційної системи

Розглянемо реалізацію тренування моделі рекомендаційної системи, в якому використовується алгоритм SVD++ та з використанням графів.

Використовуючи бібліотеку `pandas`, дані аналізуються для визначення важливості жанрів та ключових слів. Ваги обчислюються на основі їх частоти відгуків, що дозволяє визначити, які жанри та ключові слова мають більш великий вплив на вподобання користувачів.

Також дані аналізуються для визначення частоти повторень жанрів та ключових слів у відгуках користувачів. Це дозволяє встановити важливість кожної одиниці даних для прогнозування вподобань користувачів.

Після обчислення вагів, важливість жанрів та ключових слів визначається за допомогою цих ваг. Жанри та ключові слова, які мають високі ваги, мають більший вплив на вподобання користувачів.

Далі, після аналізу даних та обрахування ваг, для тренування моделі застосовується об'єкт `Reader` з бібліотеки `surprise`, щоб можна було зчитувати та перетворювати дані у формат, який можна використовувати для тренування моделі. Він визначає, як будуть представлені дані.

Після чого створюється об'єкт, для створення набору тренувальних даних, який використовує створений раніше `Reader` для перетворення даних у формат, який можна використовувати для тренування моделі. Після цього використовується метод `build_full_trainset`, щоб отримати повний набір тренувальних даних.

Останнім кроком є тренування моделі SVD++ за допомогою методу `fit`. Цей метод надає можливість навчання моделі на наданих даних. Після тренування, модель має змогу прогнозувати рейтинги для користувачів та телесеріалів на основі навчальних даних.

Після тренування моделі SVD++ проводиться тестування на тестовому наборі даних для оцінки її точності та ефективності. Це допомагає зрозуміти, наскільки добре рекомендаційна система працює на нових даних.

Для оцінки точності моделі на тестових даних використовується метрика RMSE (Рисунок 3.1).

```
Train RMSE: 0.6973894260142327  
Test RMSE: 1.0171675467576995
```

Рисунок 3.1 – Корінь середньоквадратичної помилки між реальними та прогнозованими оцінками

Також варто розглянути застосування колаборативної фільтрації на основі графів для аналізу взаємозв'язків між телесеріалами та їх ключовими словами з метою покращення рекомендаційної системи. Далі наведено детальний опис реалізації.

Для створення графу використовується бібліотека `networkx`. Кожен телесеріал є вузлом графу, а також його ключові слова. Ребра графу відображають зв'язки між телесеріалами та їх ключовими словами (Рисунок 3.2).

```
Number of nodes: 18923  
Number of edges: 70868
```

Рисунок 3.2 – Кількість вузлів та ребр у графі

Для кожного телесеріалу визначаються його ключові слова, після чого створюються ребра між телесеріалом та кожним з його ключових слів. Вага кожного ребра відповідає вазі ключового слова для даного телесеріалу, яка обчислюється на основі аналізу даних (Рисунок 3.3).

```
{(1, 'jealousy'): 4.354647016876634,
 (1, 'toy'): 7.532894736842105,
 (1, 'boy'): 3.4039390561129688,
 (1, 'friendship'): 2.7213309566250743,
 (1, 'friends'): 2.05680925115078,
 (1, 'rivalry'): 6.14352783366868,
 (1, 'boy next door'): 9.734325185972368,
 (1, 'new toy'): 9.734325185972368,
 (1, 'toy comes to life'): 8.799231508165226,
 (1, 'animation'): 49.513513513513516,
 (1, 'comedy'): 5.290974729241878,
 (1, 'family'): 20.70056497175141}
```

Рисунок 3.3 – Вага ребр між вузлами

Після створення графу він зберігається у файлі за допомогою бібліотеки `pickle`. Це дозволяє перевикористовувати граф без необхідності його повторного створення при кожному запуску скрипту.

Останнім кроком є інтеграція графової бази даних з моделлю SVD++. Це означає, що під час прогнозування рейтингу для користувача, модель SVD++ буде використовувати графовий базу даних для покращення прогнозування.

3.2.3 Формування рекомендацій

Далі описана генерація рекомендацій на основі методу колаборативної фільтрації.

Для кожного користувача визначаються телесеріали, які він ще не бачив, і для них прогноуються оцінки за допомогою тренованої моделі. Телесеріали, які користувач вже переглянув, виключаються з можливих рекомендацій. Потім для кожного невідомого телесеріалу обчислюється передбачуваний рейтинг за допомогою натренованої моделі SVD++.

Для генерації рекомендацій також використовується підхід на основі графів. Створюється граф, де вузли представляють телесеріали та їх ключові слова або жанри, а ребра відображають ваги взаємодій між ними.

Для пошуку схожих телесеріалів використовуються алгоритми випадкового блукання, такі як BiasedRandomWalk з бібліотеки stellargraph. Ця реалізація дозволяє здійснювати випадкові прогулянки графом, враховуючи ваги ребер, що дозволяє знаходити найбільш релевантні вузли для рекомендацій (Рисунок 3.5).

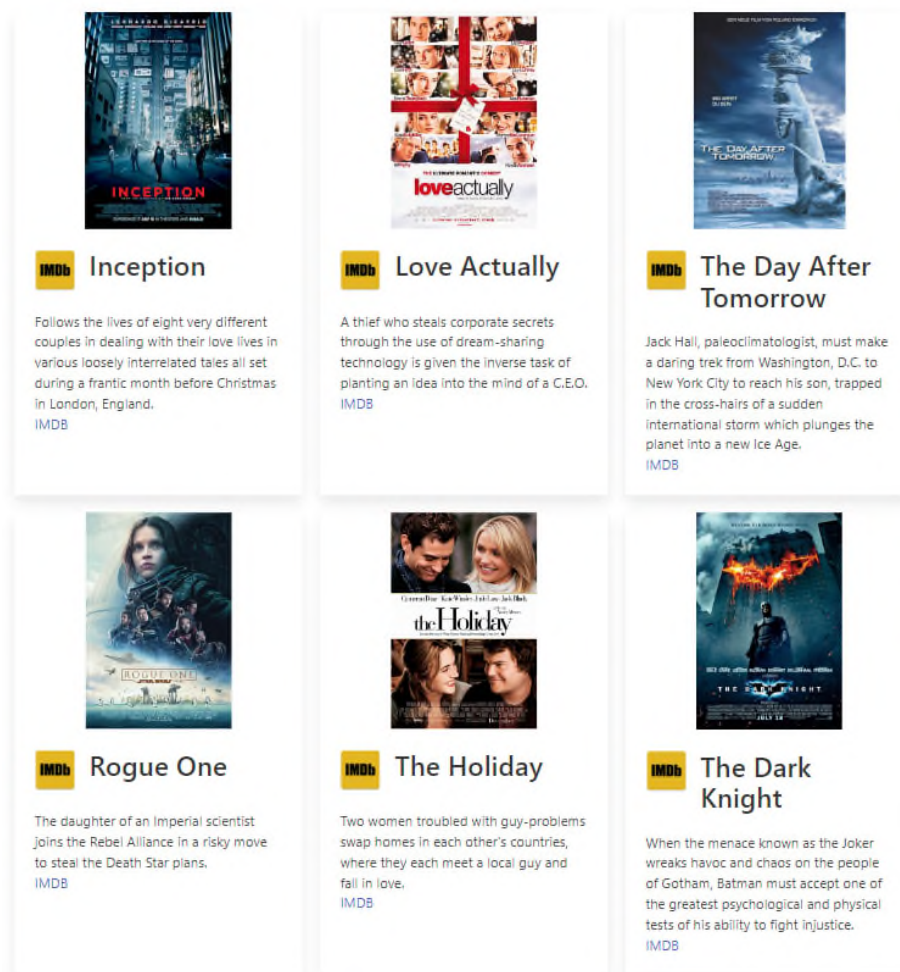


Рисунок 3.5 – Вигляд рекомендованих телесеріалів

Поєднання цих реалізацій та технологій дозволило створити ефективну та інтуїтивно зрозумілу рекомендаційну систему для підбору телесеріалів. Кожна реалізація модуля забезпечує певний аспект функціональності, що в сукупності дозволяє створити потужний інструмент для користувачів.

3.4 Тестування рекомендаційної системи

Наступним кроком є тестування рекомендаційної системи, щоб забезпечити її максимальну ефективність та точність. Для підтвердження коректної роботи, використовується модульне тестування компонентів, відповідальних за обробку даних певних таблиць.

Цей тест перевіряє коректність завантаження даних із датасету. Тест описується у вигляді юніт-тесту, де використовуються методи із бібліотек `pd.read_csv` (Рисунок 3.6).

```
movies : (45466, 24)
ratings : (100004, 4)
keywords : (46419, 2)
links : (9125, 3)
```

Рисунок 3.6 – Результати роботи тестування правильного завантаження даних із датасету

В результаті тестування показало, що дані завантажуються та обробляються коректно. Перевірка пройшла успішно, що підтверджує правильність виконання цього етапу. Це забезпечує надійну основу для подальшої роботи з обробки даних.

Наступний юніт-тест спрямований на перевірку правильності перетворення та об'єднання даних з різних таблиць, а саме перевірка на коректність зміни типу даних та об'єднання таблиць. Це все є важливим для забезпечення цілісності даних, які використовуватимуться в рекомендаційній системі.

В процесі перевіряється коректність перетворення типу даних стовпця `id` в таблиці `movies` з `object` на `float`. Це необхідно для об'єднання з таблицею `links`, де `tmdbId` має числовий формат.

Після цього об'єднуємо таблиці `movies` та `links` за стовпцем `tmdbId`, додаючи інформацію з `links` до `movies`, а після видаляються непотрібні стовпці

imdbId та id з movies для зменшення обсягу даних і уникнення дублювання (Рисунок 3.7).

| | title | genres | movieId | tmdbId | keywords | userId | rating | timestamp |
|---|---------------------|----------------------------------|---------|---------|--|--------|--------|-----------|
| 0 | Se7en | ['crime', 'mystery', 'thriller'] | 47 | 807.0 | ['self-fulfilling prophecy', 'detective...'] | 383 | 5.0 | 789652009 |
| 1 | A Fish Called Wanda | ['comedy', 'crime'] | 1079 | 623.0 | ['robbery', 'infidelity', 'cheating', '...'] | 383 | 3.0 | 789652009 |
| 2 | Get Shorty | ['comedy', 'thriller', 'crime'] | 21 | 8012.0 | ['gambling', 'miami', 'based on novel', '...'] | 383 | 3.0 | 789652009 |
| 3 | Shanghai Triad | ['drama', 'crime'] | 30 | 37557.0 | ['servant', 'cabaret', 'shanghai', 'div...'] | 409 | 5.0 | 828212412 |
| 4 | Carrington | ['history', 'drama', 'romance'] | 35 | 47018.0 | ['england', 'painter', 'conscientious o...'] | 409 | 4.0 | 828212412 |

Рисунок 3.7 – Результати роботи тестування перетворення даних

Для перевірки коректності переглядаються перші рядки таблиці movies та аналізується її структура. Тестування показало, що таблиця має правильні розміри – 8 стовпців, це означає, що об'єднання пройшло успішно.

Наступне тестування полягає в тому, щоб перевірити ефективність та точність рекомендаційної системи після її тренування на основі передбачених вподобання користувача.

Після проведення тренування моделі використовуються наступні кроки для тестування правильності рекомендацій:

Для вибору користувача для тестування були використані дані з датасету для оцінки точності рекомендацій. Для обраного користувача використовуються всі телесеріали, яким він дав оцінки. Це дозволяє знати фактичні уподобання користувача, які будуть використовуватися для порівняння з передбаченими рекомендаціями.

При тестуванні рекомендаційної системи важливо розуміти, як дані розподіляються між тренувальним та тестовим наборами. Тому проведено аналіз цього розподілу, що надає ключові показники, які допомагають оцінити, наскільки добре дані тренування відображають дані тестування.

Спочатку аналізуються телесеріали, які присутні в тренувальному та тестовому наборах даних на рисунку 3.8:

```
Movies in train set: 7328
Movies in test set: 4732
Movies common in train and test sets: 3035
Movies present in train set but not in test set: 4293
Movies present in test set but not in train set: 1697
Percentage of movies present in test but not in train set of all the movies: 18.803324099722992%
```

Рисунок 3.8 – Статистика телесеріалів тестовому наборі

Ці результати показують, що тренувальний набір містить більше телесеріалів, ніж тестовий, і деяка частка телесеріалів є унікальною для кожного набору.

Далі аналізуються користувачі, присутні в тренувальному та тестовому наборах даних на рисунку 3.9:

```
Users in train set: 546
Users in test set: 148
Users common in train and test sets: 23
Users present in train set but not in test set: 523
Users present in test set but not in train set: 125
Percentage of users present in test but not in train set of all the users: 18.628912071535023%
```

Рисунок 3.9 – Статистика користувачів тестовому наборі

Такі результати вказують на те, що в обох наборах даних є як спільні, так і унікальні елементи. Високий відсоток унікальних телесеріалів і користувачів в тестовому наборі свідчить про те, що система рекомендацій повинна бути достатньо гнучкою, щоб добре працювати з новими даними. Це також підкреслює важливість різноманітності та повноти тренувального набору для покращення точності рекомендацій.

Далі, використовуючи треновану модель, юніт-тест спрямований на перевірку генерації списку телесеріалів, які рекомендуються для користувача. Частина вхідних даних для формування рекомендацій зображені на рисунку 3.10:

| userId | movieId | rating |
|--------|---------|--------|
| 108 | 274 | 4.0 |
| 108 | 320 | 4.5 |
| 108 | 1119 | 3.0 |
| 108 | 2361 | 5.0 |
| 108 | 7627 | 4.5 |

Рисунок 3.10 – Вивід даних про рейтинги користувача «108»

Список рекомендацій формується на основі вказаних даних, згідно з якими movieId включає телесеріали, серед яких є:

- 1) «Buffy the Vampire Slayer», що відноситься до жанрів бойовик, фентезі, драма;
- 2) «The Lord of the Rings», який об'єднує фентезі, пригоди, драма, війна;
- 3) «Forrest Gump», що належить до жанру комедія, драма, романс;
- 4) «Game of Thrones», який поєднує фентезі, драму та пригоди;
- 5) «Fight Club», що включає жанри драма, трилер, кримінал.

Рекомендовані телесеріали порівнюються з реальними уподобаннями користувача. Оцінка точності проводиться за допомогою таких метрик, як precision та recall, які визначають, наскільки добре система вгадала уподобання користувача.

Під час тестування для обраного користувача були отримані наступні результати, які зображені на рисунку 3.11:

| | title | genres |
|------|--------------------------|--|
| 273 | Three Colors: Red | ['drama', 'mystery', 'romance'] |
| 320 | Forrest Gump | ['comedy', 'drama', 'romance'] |
| 618 | A Close Shave | ['family', 'animation', 'comedy'] |
| 779 | Night of the Living Dead | ['horror'] |
| 780 | The African Queen | ['adventure', 'war', 'romance'] |
| 946 | The Princess Bride | ['adventure', 'family', 'fantasy', 'com...'] |
| 1514 | Rain Man | ['drama'] |
| 2361 | Fight Club | ['drama'] |
| 3337 | Memento | ['mystery', 'thriller'] |
| 6257 | The Departed | ['drama', 'thriller', 'crime'] |

Рисунок 3.11 – Результати роботи при тестуванні формування рекомендацій

Зі списку рекомендованих телесеріалів, три телесеріали: «Forrest Gump», «Fight Club», «Memento» – відповідають реальним уподобанням користувача. Це свідчить про високу точність рекомендаційної системи.

У даному випадку метрика precision вказує на відношення кількості правильних рекомендацій до загальної кількості рекомендацій, а саме $3/10 = 0.3$ або 30%.

Метрика recall вказує на відношення кількості правильних рекомендацій до загальної кількості реальних уподобань користувача, яка показує $3/3 = 1$ або 100%.

Також проведено додатковий юніт-тест, який також спрямований на перевірку генерації списку телесеріалів, які рекомендуються для користувача. Вхідні дані для формування рекомендацій зображені на рисунку 3.12:

| userId | movieId | rating |
|--------|---------|--------|
| 52 | 372 | 4.5 |
| 52 | 723 | 4.5 |
| 52 | 779 | 4.5 |
| 52 | 946 | 5.0 |
| 52 | 1119 | 5.0 |
| 52 | 2491 | 4.0 |

Рисунок 3.12 – Вивід даних про рейтинги користувача «52»

Список рекомендацій формується на основі даних, згідно з якими movieId включає телесеріали, серед яких є:

- 1) «What We Do in the Shadows», що відноситься до жанрів комедія та жахи;
- 2) «Sleepy Hollow», який об'єднує фантастику, містику та трилер;
- 3) «Night of the Living Dead», що належить до жанру жахів;
- 4) «The Princess Bride», який поєднує пригоди, сімейні, фантастичні та комедійні елементи;
- 5) «Game of Thrones», що включає фантастику, драму та пригоди;
- 6) «The Haunting of Hill House», який поєднує жахи та драму.

Тренована модель колаборативної фільтрації створює список рекомендацій щодо телесеріалів. Під час тестування для обраного користувача було отримано результати, що відображені на рисунку 3.13.

| | title | genres |
|------|---------------------------|--|
| 185 | Stranger Things | ['horror', 'mystery', 'science fiction'] |
| 423 | Supernatural | ['fantasy', 'horror', 'drama'] |
| 779 | Night of the Living Dead | ['horror'] |
| 946 | The Princess Bride | ['adventure', 'family', 'fantasy', 'com...'] |
| 1119 | Game of Thrones | ['adventure', 'drama', 'fantasy'] |
| 1451 | Shadow and Bone | ['adventure', 'drama', 'fantasy'] |
| 1514 | The Thing | ['science fiction', 'horror'] |
| 2021 | Hannibal | ['horror', 'thriller'] |
| 2737 | The Haunting of Bly Manor | ['horror', 'romance'] |
| 4152 | Pulp Fiction | ['horror', 'romance', 'crime'] |

Рисунок 3.13 – Результати роботи при тестуванні формування рекомендацій

Три телесеріали зі списку рекомендованих – «Night of the Living Dead», «The Princess Bride», «Game of Thrones» – відповідають реальним уподобанням користувача, що свідчить про високу ефективність рекомендаційної системи. У даному контексті метрика precision становить 3 з 10, що дорівнює 0.3 або 30%. А метрика recall становить 3 з 3, що дорівнює 1 або 100%.

Такі результати тестування показують, що рекомендаційна система ефективно вгадує уподобання користувачів та надає релевантні рекомендації. Високі значення точності та повноти підтверджують, що система здатна правильно передбачати телесеріали, які можуть сподобатися користувачам.

Далі було проведено тестування, яке включало використання тест-кейсів для перевірки функціональності та ефективності системи.

Перший тест-кейс, представлений у таблиці 3.1, в якому описані кроки для перевірки працездатності вебсистеми. Тест-кейс призначений для забезпечення того, що система функціонує відповідно до встановлених вимог та стандартів.

Таблиця 3.1 – Тест-кейс RC00001

| Тест-кейс ID: RC00001 | Пріоритет: 1 | Створено: 23.05.2024, Пасічник О.О. |
|---|---|--|
| Назва: Перевірка коректності та зручності взаємодії рекомендаційної системи з вебсистемою | | |
| Вхідні дані: Запущена вебсистема для рекомендацій телесеріалів. | | |
| Кроки | Очікуваний результат | |
| <p>1. Відкрити браузер та перейти за посиланням на вебсистему.</p> <p>2. Перевірити наявність елементів у вебінтерфейсі: кнопок, посилань.</p> <p>3. Переміщуватись по вебсистемі, використовуючи основні функції: отримання рекомендацій.</p> <p>4. Перевірити швидкодію вебсистеми під час переходу між сторінками та завантаженням контенту.</p> <p>5. Перевірити адаптивність вебсистеми на різних пристроях та розмірах екранів.</p> <p>6. Завершити сеанс роботи з вебсистемою та перевірити правильність закриття сесії.</p> | <p>Вебсистема запускається без помилок.</p> <p>Елементи інтерфейсу відображаються коректно на різних пристроях.</p> <p>Користувач може безпроблемно отримувати рекомендації.</p> <p>Сторінки вебсистеми завантажуються швидко та без зайвої затримки.</p> | |
| Результат виконання тест-кейсу: пройдено успішно | | |

Під час проведення тестування вебсистеми не було виявлено непрацюючих або несправних функцій. Це однозначно підтверджує коректну роботу та зручність використання рекомендаційної системи в поєднанні із вебсистемою.

Далі проведено тест-кейс, який представлений у таблиці 3.2, на перевірку тренування рекомендаційної системи підбору.

Таблиця 3.2 – Тест-кейс RC00002

| Тест-кейс ID: RC00002 | Пріоритет: 1 | Створено: 23.05.2024, Пасічник О.О. |
|--|---|--|
| Назва: Перевірка тренування рекомендаційної системи | | |
| Вхідні дані: CSV-файли з даними про телесеріали та рейтинги користувачів. | | |
| Кроки | Очікуваний результат | |
| 1. Запустити скрипт тренування моделі. | Скрипт успішно запускається без помилок. | |
| 2. Перевірити, що дані з CSV-файлів movies_metadata.csv, ratings_small.csv коректно завантажені. | Дані зчитані коректно і відповідають очікуваним розмірам та структурі. Відсутні значення оброблені або вилучені; дублікати вилучені з даних. | |
| 3. Впевнитись в тому, що відсутні значення та дублікати в даних оброблені коректно. | Рейтинги користувачів та дані про телесеріали об'єднані в єдиний датасет. | |
| 4. Перевірити, що рейтинги користувачів та дані про телесеріали зібрані разом в один датасет. | Модель тренується без помилок і зберігається для подальшого. | |
| 5. Запустити тренування рекомендаційної моделі на зібраних даних. | | |
| 6. Перевірити, що модель готова для використання після тренування. | | |
| 7. Перевірити якість тренування моделі за допомогою метрик | | |

| | |
|---|--|
| <p>якості.</p> <p>8. Зберегти навчену модель для подальшого використання в рекомендаційній системі.</p> | |
| <p>Результат виконання тест-кейсу: пройдено успішно</p> | |

Після завершення тренування модель готова до використання для генерації рекомендацій користувачам. Метрики якості показують задовільні результати, що свідчить про те, що модель ефективно тренується на наданих даних та може надавати рекомендації з високою точністю.

Навчена модель успішно збережена та готова для використання системі, що дозволяє використовувати її безпосередньо для генерації персоналізованих рекомендацій для користувачів.

Отже, коректна робота вебсистеми та отримані результати виконання усіх тестів підтверджують, що розроблена рекомендаційна система відповідає вимогам та потребам користувачів для ефективного отримання персоналізованих рекомендацій телесеріалів. Кожен тест показав, що рекомендаційна система колаборативної фільтрації працює стабільно, без помилок та зручно для користувачів.

3.5 Висновки до розділу 3

У цьому розділі детально розглянуто різноманітні вебтехнології, а також засоби, які використовувалися для розробки рекомендаційної системи створеної для підбору телесеріалів з використанням колаборативної фільтрації, що дозволило реалізувати її.

Також описано принципи роботи модулів навчання SVD++ та їхньої взаємодії з реалізаційним підходом на основі графів, а також з іншими модулями системи. Кожен модуль забезпечує певний аспект функціональності, що в сукупності дає змогу реалізувати потужний інструмент для користувачів.

На завершальному етапі проведено тестування рекомендаційної системи з отриманням результатів, які підтверджують її ефективність та продуктивність. Кожен тест показав, що рекомендаційна система працює стабільно та зручно для користувачів.

Таким чином, створено рекомендаційну систему для підбору телесеріалів, яка задовольнить потреби користувачів у пошуку нових телесеріалів, які можуть зацікавити їх на основі їхніх уподобань та інших факторів.

Загальні висновки

Кваліфікаційна робота бакалавра спрямована на реалізацію системи персоналізованих рекомендацій для сайтів з перегляду телесеріалів, яка використовує колаборативну фільтрацію для покращення сервісу та ефективного підбору контенту для кожного користувача.

В результаті виконання кваліфікаційної роботи бакалавра було розроблено рекомендаційну систему колаборативної фільтрації, яка виконує функції персоналізованого підбору телесеріалів на основі аналізу великих обсягів даних про перегляди та відгуки користувачів. Відповідно до мети були розроблені наступні функції для рекомендаційної системи колаборативної фільтрації:

- проведено аналіз систем рекомендацій на основі колаборативної фільтрації;
- розробити рекомендаційну систему, яка використовує колаборативну фільтрацію для формування персоналізованих рекомендацій;
- розроблено функціональну та інформаційну структуру вебсистеми перегляду телесеріалів;
- розроблено програмну реалізацію вебсистеми підбору телесеріалів з використанням рекомендаційної системи на основі колаборативної фільтрації;
- виконано тестування рекомендаційної системи підбору телесеріалів.

Дана реалізація відповідає актуальній потребі в інструментах, які допомагають користувачам ефективно знаходити контент, що відповідає їхнім вподобанням у середовищі надлишкової інформації.

Дослідження ефективності системи під час тестування показало, що рекомендації, засновані на колаборативній фільтрації, мають високий рівень відповідності очікуванням користувачів, що свідчить про високу ефективність реалізації системи.

Рекомендаційна система ефективно формує список рекомендованих телесеріалів, які відповідають реальним уподобанням користувача. Зокрема, для

одного з користувачів система рекомендувала телесеріали, які відповідали його уподобанням. Метрики precision та recall для цього випадку становили 0.3 та 1 відповідно, що підтвердили правильність рекомендацій. У другому тестовому випадку система також показала високу ефективність, рекомендуючи телесеріали, які відповідали уподобанням користувача. Метрики precision та recall для цього випадку також становили 0.3 та 1 відповідно.

Рекомендаційна система успішно пройшла тестування, що дозволило уникнути потенційних помилок, які могли б завадити адекватному використанню.

Розроблена рекомендаційна система має значні перспективи для впровадження, оскільки вона може бути широко використана у вебсервісах, які займаються розподілом контенту.

Можливості та шляхи до вдосконалення програмного продукту включають інтеграцію з іншими типами рекомендаційних систем, а також розширення бази даних про телесеріали для підвищення точності рекомендацій. Також планується впровадження інтерактивних функцій, які дозволять користувачам більш активно взаємодіяти з системою, що в свою чергу підвищить її ефективність та привабливість для користувачів.

Перелік посилань

1. Cambridge Scholars. Television Series as Mirrors of Contemporary Life. URL: <https://www.cambridgescholars.com/resources/pdfs/978-1-5275-3399-8-sample.pdf>.
2. Medium. The Rise of Streaming Platforms. URL: <https://medium.com/@kasturichatterjee1108/the-rise-of-streaming-platforms-a-revolution-in-entertainment-3553b094d799>.
3. Wikipedia. Information overload. URL: https://en.wikipedia.org/wiki/Information_overload.
4. LinkedIn. Unveiling the Future: The Rise of AI-Powered Personalization in Tech. URL: <https://www.linkedin.com/pulse/unveiling-future-rise-ai-powered-personalization-tech-vrushti-shah-fmejf/>.
5. AIContentfy. The role of AI in content personalization. URL: <https://aicontentfy.com/en/blog/role-of-ai-in-content-personalization>.
6. Nvidia. Types of Recommendation Systems. URL: [https://www.nvidia.com/en-us/glossary/recommendation-system/#:~:text=A%20recommendation%20system%20\(or%20recommender,exponentially%20growing%20number%20of%20options](https://www.nvidia.com/en-us/glossary/recommendation-system/#:~:text=A%20recommendation%20system%20(or%20recommender,exponentially%20growing%20number%20of%20options).
7. Wikipedia. Collaborative filtering. URL: https://en.wikipedia.org/wiki/Collaborative_filtering.
8. Github. Neural Collaborative Filtering (NCF) for TensorFlow. URL: <https://github.com/NVIDIA/DeepLearningExamples/tree/master/TensorFlow/Recommendation/NCF>.
9. Github. Variational Autoencoder for Collaborative Filtering for TensorFlow. URL: <https://github.com/NVIDIA/DeepLearningExamples/tree/master/TensorFlow/Recommendation/VAE-CF>.
10. Medium. Prototyping a Recommender System Step by Step Part 1: KNN Item-Based Collaborative Filtering. URL: <https://towardsdatascience.com/prototyping->

a-recommender-system-step-by-step-part-1-knn-item-based-collaborative-filtering-637969614ea.

11. MDPI. Adaptive k-NN-Based Extended Collaborative Filtering Recommendation Services. URL: <https://www.mdpi.com/2504-2289/7/2/106#:~:text=The%20KNN%2Dbased%20collaborative%20filtering%20algorithm%20is%20a%20type%20of,users%20to%20the%20target%20user.>

12. Quuxlabs. Matrix Factorization: A Simple Tutorial and Implementation in Python. URL: <http://www.quuxlabs.com/blog/2010/09/matrix-factorization-a-simple-tutorial-and-implementation-in-python/>.

13. LinkedIn. Fundamental of Matrix Factorization For Recommender System. URL: <https://www.linkedin.com/pulse/fundamental-matrix-factorization-recommender-system-saurav-kumar/>.

14. Wikipedia. Netflix. URL: <https://en.wikipedia.org/wiki/Netflix>.

15. Googleplay. Netflix URL: https://play.google.com/store/apps/details?id=com.netflix.ninja&hl=en_US.

16. Netflix. How Netflix's Recommendations System Works. URL: <https://help.netflix.com/en/node/100639>.

17. Gadgets360. Prime Video Channels Bundling Service Launched by Amazon for Video Streaming Apps in India. URL: <https://www.gadgets360.com/entertainment/news/prime-video-channels-amazon-bundling-service-video-streaming-subscribe-single-interface-2551847>.

18. Baeldung. Collaborative Filtering. URL: <https://www.baeldung.com/cs/amazon-recommendation-system#:~:text=Collaborative%20Filtering,Their%20browsing%20behavior%20and%20ratings.>

19. Wikipedia. Hulu. URL: <https://en.wikipedia.org/wiki/Hulu>.

20. Googleplay. Hulu for Android TV URL: <https://play.google.com/store/apps/details?id=com.hulu.livingroomplus&hl=uk>.

21. Hulu. Personalization Features on Hulu. URL: <https://help.hulu.com/article/hulu-personalized-recommendations>.

22. Mdpi. Utilizing Alike Neighbor Influenced Similarity Metric for Efficient Prediction in Collaborative Filter-Approach-Based Recommendation System. URL: <https://www.mdpi.com/2076-3417/12/22/11686>.

23. Researchgate. Utilizing Alike Neighbor Influenced Similarity Metric for Efficient Prediction in Collaborative Filter-Approach-Based Recommendation System. URL: https://www.researchgate.net/publication/365498582_Utilizing_Alike_Neighbor_Influenced_Similarity_Metric_for_Efficient_Prediction_in_Collaborative_Filter-Approach-Based_Recommendation_System.

24. Bitnine. What's special about a graph-based recommendation system? URL: <https://bitnine.net/blog-graph-database/graph-based-recommendation-system/?ckattempt=1>.

25. Mdpi. A Graph Neural Network Social Recommendation Algorithm Integrating the Multi-Head Attention Mechanism. URL: <https://www.mdpi.com/2079-9292/12/6/1477>.

26. Springerlink. Recommending on graphs: a comprehensive review from a data perspective. URL: <https://link.springer.com/article/10.1007/s11257-023-09359-w>

27. Wikipedia. Precision and recall. URL: https://en.wikipedia.org/wiki/Precision_and_recall.

28. Movielens. Non-commercial, personalized movie recommendations. URL: <https://movielens.org/>.

29. Kaggle. The movies dataset. URL: <https://www.kaggle.com/datasets/rounakbanik/the-movies-dataset>.

30. TypeScript. The TypeScript Handbook. URL: <https://www.typescriptlang.org/docs/handbook/intro.html>.

31. Vite. Getting Started. URL: <https://vitejs.dev/guide/>.

32. Sass. Sass is a stylesheet language that's compiled to CSS. URL: <https://sass-lang.com/documentation/>.

33. Numpy. NumPy documentation. URL: <https://numpy.org/doc/stable/index.html>.

34. Pandas. Pandas documentation. URL: <https://pandas.pydata.org/docs/>.

ДОДАТКИ

Додаток А

Програмні коди

Лістинг rec-system.py:

```
import streamlit as st
import numpy as np
import pandas as pd
import surprise
import pickle
import networkx as nx
from stellargraph import StellarGraph
from stellargraph.data import BiasedRandomWalk
from sklearn.feature_extraction.text import CountVectorizer
import requests

st.set_page_config(
    page_title="Movie App",
    page_icon=":film_projector:",
    layout="centered",
    initial_sidebar_state="expanded"
)

def get_movie_details(tmdb_id):
    response = requests.get(
        f'https://api.themoviedb.org/3/movie/{tmdb_id}?api_key=d8ee6f66bcbb558bffdca1a35af3f725d&language=en-US')
    return response.json()

base_path = 'C:/Users/user/Desktop/dyplom/'
base_image_url = 'https://image.tmdb.org/t/p/w500'
X_train = pd.read_csv(base_path + 'xtrain.csv')
train_movies = pd.read_csv(base_path + 'train_movies.csv')

user_id = np.random.choice(range(671))

popular_movies = X_train.groupby('movieId').count()['rating']
popular_movie_ids = popular_movies[popular_movies > 100].index
popular_movies_data = X_train[X_train.movieId.isin(popular_movie_ids)]
top20_movie_ids =
popular_movies_data.groupby('movieId').rating.mean().sort_values(ascending=False)[:20].index

svdpp_algo = surprise.dump.load(base_path + 'svdpp_algo.pkl')[1]
train_ratings = X_train[['userId', 'movieId', 'rating']]

def recommend_collaborative(user_id):
    watched_movies = set(train_ratings[train_ratings.userId == user_id].movieId.values)
    if not watched_movies:
        return [], []

    unwatched_movies = set(train_movies.movieId) - watched_movies
    predictions = [svdpp_algo.predict(user_id, mid) for mid in unwatched_movies]
    sorted_predictions = pd.DataFrame([(pred.iid, pred.est) for pred in predictions],
                                      columns=['movieId',
'rating']).sort_values('rating', ascending=False)
```

```

    top10_movie_ids = sorted_predictions.movieId[:10]
    liked_movie_ids = train_ratings[train_ratings.userId ==
user_id].sort_values('rating',
ascending=False).movieId.values[:5]
    return top10_movie_ids, liked_movie_ids

top10_movie_ids, liked_movie_ids = recommend_collaborative(user_id)

B = pickle.load(open(base_path + 'graph.pkl', 'rb'))

def recommend_graph_based(query_movie_id):
    random_walk = BiasedRandomWalk(StellarGraph(B))
    walk = random_walk.run(nodes=[query_movie_id], n=1, length=10000, p=0.01, q=100,
weighted=True, seed=42)
    walk = [str(node) for node in walk[0] if isinstance(node, int)]
    walk_str = ' '.join(walk)

    movie_vocab = {str(movie): idx for idx, movie in
enumerate(train_movies.movieId.sort_values().unique())}
    vectorizer = CountVectorizer(vocabulary=movie_vocab)
    walk_vector = vectorizer.fit_transform([walk_str])

    reverse_vocab = {idx: int(movie) for movie, idx in movie_vocab.items()}
    walk_vector = walk_vector.toarray()[0]

    top5_recommended_ids = [reverse_vocab[idx] for idx in walk_vector.argsort()[::-1] if
idx in reverse_vocab and reverse_vocab[idx] !=
query_movie_id][:5]
    return top5_recommended_ids

def display_recommendations(movie_ids, steps):
    movies = []
    for movie_id in movie_ids:
        tmdb_id = train_movies[train_movies.movieId == movie_id].tmdbId.values[0]
        details = get_movie_details(tmdb_id)
        title = details['title']
        image_url = base_image_url + details['poster_path']
        movies.append((title, image_url))

    cols = st.columns(5)
    for col, (start, end) in zip(cols, zip([0] + steps[:-1], steps)):
        with col:
            for title, image_url in movies[start:end]:
                st.image(image_url)
                st.text(title)

def search_movie():
    selected_movie = st.selectbox('What are you looking for...?',
train_movies.title.values)
    movie_id, tmdb_id = train_movies[train_movies.title == selected_movie][['movieId',
'tmdbId']].values[0]
    details = get_movie_details(tmdb_id)

    if st.button('Search'):
        col1, col2 = st.columns(2)

```

```

with col1:
    st.image(base_image_url + details['poster_path'])
with col2:
    st.header(details['title'])
    st.caption(details['tagline'])
    st.write(details['overview'])
    st.markdown(f"***Released in {details['release_date']}**")
    st.write(f'Runtime: {details['runtime']} mins')
    st.write(f'Avg. Rating: {details['
vote_average
']} :star:  Votes: {details['
vote_count
']} :thumbsup:')

    st.header("More like this...")
    recommended_ids = recommend_graph_based(movie_id)
    display_recommendations(recommended_ids, [1, 2, 3, 4, 5])

    if not top10_movie_ids:
        st.header('Hello stranger!!')
        search_movie()
        st.title('Most Popular movies on the platform')
        display_recommendations(top20_movie_ids, [4, 8, 12, 16, 20])
    else:
        st.header(f'Welcome user {user_id}')
        search_movie()
        st.title('Your Favourites...')
        display_recommendations(liked_movie_ids, [1, 2, 3, 4, 5])
        st.title('Based on your taste...')
        display_recommendations(top10_movie_ids, [2, 4, 6, 8, 10])

```

Лістинг rec-system_train.py:

```

import pandas as pd
import numpy as np
import warnings
import ast
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
import seaborn as sns
import os
from surprise import Reader, Dataset, SVDpp
import surprise
import pickle
import networkx as nx
warnings.filterwarnings('ignore')

base_path = 'C:/Users/user/Desktop/dyplom/'

movies = pd.read_csv(base_path + 'movies_metadata.csv')
ratings = pd.read_csv(base_path + 'ratings_small.csv')
keywords = pd.read_csv(base_path + 'keywords.csv')
links = pd.read_csv(base_path + 'links_small.csv')

movies.drop_duplicates(inplace=True)
keywords.drop_duplicates(inplace=True)
movies.drop(movies[movies.title.isnull()].index, inplace=True)

movies = movies[['id', 'title', 'genres']]
movies.id = movies.id.astype(float)
movies = movies.merge(links, left_on='id', right_on='tmdbId')
movies.drop(['imdbId', 'id'], axis=1, inplace=True)

```

```

movies.drop_duplicates(inplace=True)
movies = movies.merge(keywords, left_on='tmdbId', right_on='id')

movies.drop(['id'], axis=1, inplace=True)

def clean_data(row):
    return str([(i['name'].lower().strip()) for i in row]) if row else str(row)

movies.keywords = movies.keywords.apply(lambda x: clean_data(ast.literal_eval(x)))
movies.genres = movies.genres.apply(lambda x: clean_data(ast.literal_eval(x)))

if not os.path.isfile(base_path + 'movie_ratings.csv'):
    movie_ratings = movies.merge(ratings, on='movieId')
    movie_ratings.sort_values('timestamp', inplace=True)
    movie_ratings.to_csv(base_path + 'movie_ratings.csv', index=False)
movie_ratings = pd.read_csv(base_path + 'movie_ratings.csv')

if not (os.path.isfile(base_path + 'xtrain.csv') and os.path.isfile(base_path +
'xtest.csv')):
    movie_ratings[:int(len(movie_ratings) * 0.8)].to_csv(base_path + 'xtrain.csv',
index=False)
    movie_ratings[int(len(movie_ratings) * 0.8):].to_csv(base_path + 'xtest.csv',
index=False)

X_train = pd.read_csv(base_path + 'xtrain.csv')
X_test = pd.read_csv(base_path + 'xtest.csv')

train_movies = movies[movies.movieId.isin(set(X_train.movieId.values))]
test_movies = movies[movies.movieId.isin(set(X_test.movieId.values))]

genres = set()
for genre_list in train_movies.genres:
    genres.update(ast.literal_eval(genre_list))
genres = list(genres)

genre_dist = [(len(train_movies[train_movies.genres.str.match('.*' + genre + '.*'])) /
len(train_movies) * 100, genre) for genre in genres]
genre_dist = pd.DataFrame(genre_dist).sort_values(0)
genre_dist[0] = genre_dist[0].apply(lambda x: x**(-1))
genre_weights = dict(zip(list(genre_dist[1].values), list(genre_dist[0].values)))
keywords_set = set()
for keyword_list in train_movies.keywords:
    keywords_set.update(ast.literal_eval(keyword_list))
keywords_list = list(keywords_set)

keyword_dist = [(len(train_movies[train_movies.keywords.str.match('.*' + keyword +
'.*'])) / len(train_movies) * 100, keyword) for keyword in keywords_list if keyword]
keyword_dist = pd.DataFrame(keyword_dist).sort_values(0)
keyword_dist[0] = keyword_dist[0].apply(lambda x: (x + 1) ** (-1))
keyword_weights = dict(zip(list(keyword_dist[1].values), list(keyword_dist[0].values)))

train_ratings = X_train[['userId', 'movieId', 'rating']]
test_ratings = X_test[['userId', 'movieId', 'rating']]

reader = Reader(rating_scale=(1, 5))
data = Dataset.load_from_df(train_ratings, reader)
trainset = data.build_full_trainset()
testset = test_ratings.values

```

```

if not os.path.isfile(base_path + 'svdpp_algo.pkl'):
    svdpp_algo = SVDpp(random_state=24)
    svdpp_algo.fit(trainset)
    surprise.dump.dump(base_path + 'svdpp_algo.pkl', algo=svdpp_algo)

edges = []
for _, row in train_movies.iterrows():
    mid = row.movieId
    edges.extend([(mid, k, keyword_weights.get(k, 0) * 10) for k in
ast.literal_eval(row.keywords)])
    edges.extend([(mid, g, genre_weights[g] * 200) for g in
ast.literal_eval(row.genres)])

edges_data = pd.DataFrame(edges, columns=['movieId', 'keyword', 'weight'])

if not os.path.isfile(base_path + 'graph.pkl'):
    B = nx.Graph()
    B.add_nodes_from(edges_data.movieId.unique(), bipartite=0, label='movie')
    B.add_nodes_from(edges_data.keyword.unique(), bipartite=1, label='keyword')
    B.add_weighted_edges_from(edges)
    with open(base_path + 'graph.pkl', 'wb') as graph_file:
        pickle.dump(B, graph_file)

```

Додаток Б

Презентаційний матеріал



Кваліфікаційна робота бакалавра

Рекомендаційна система підбору телесеріалів
з використанням колаборативної фільтрації
для вебсистем



Виконав: студент 4-го курсу, група КН-20-2
Керівник: к.т.н., доц. каф. КН

Пасічник Олександр Олександрович
Руслан Багрій Олександрович

АКТУАЛЬНІСТЬ

Через велику кількість надлишкової інформації та контенту, важливо мати інструменти, які допомагають користувачам знаходити те, що їм дійсно цікаво. Персоналізація, на основі штучного інтелекту, може бути ключем до позитивного досвіду взаємодії з технологіями, роблячи його більш приємним та ефективним. Аналізуючи величезні обсяги даних про перегляди, реакції та відгуки глядачів, штучний інтелект пропонує персоналізований контент та прогнозує тенденції.

МЕТА

Мета кваліфікаційної роботи бакалавра – покращення сервісу при роботі з сайтами по перегляду телесеріалів шляхом формування персоналізованих рекомендацій щодо перегляду за допомогою колаборативної фільтрації

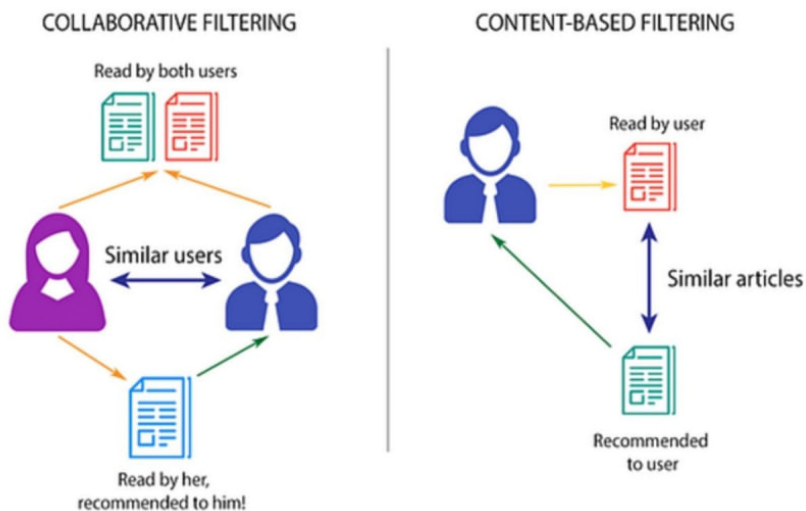
В результаті роботи, система підвищуватиме задоволеність користувачів від перегляду телесеріалів та покращуватиме сервіс сайтів з перегляду телесеріалів.

ЗАДАЧІ ДЛЯ

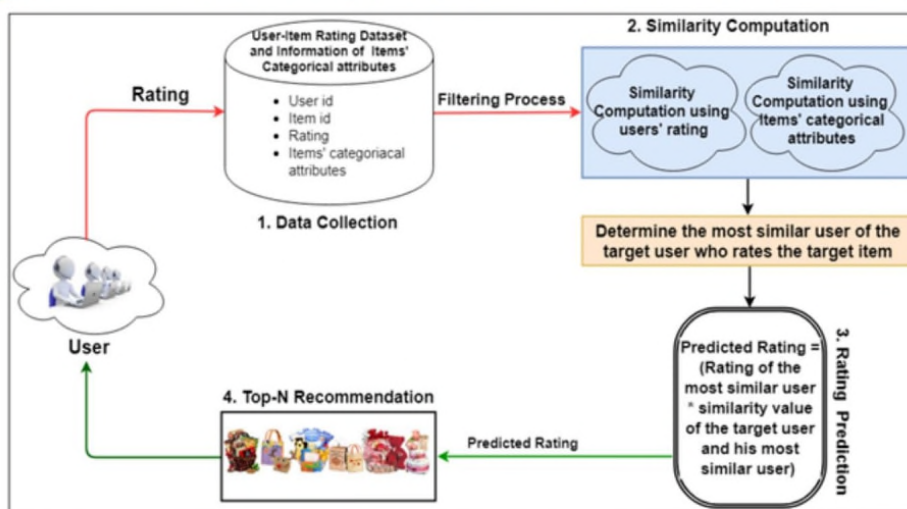
ДОСЯГНЕННЯ МЕТИ

- Провести аналіз систем рекомендацій на основі колаборативної фільтрації;
- розробити рекомендаційну систему, яка використовує колаборативну фільтрацію для формування персоналізованих рекомендацій;
- розробити функціональну та інформаційну структуру вебсистеми перегляду телесеріалів;
- розробити програмну реалізацію вебсистеми підбору телесеріалів з використанням рекомендаційної системи на основі колаборативної фільтрації;
- виконати тестування рекомендаційної системи підбору телесеріалів.

АНАЛІЗ СИСТЕМ РЕКОМЕНДАЦІЙ

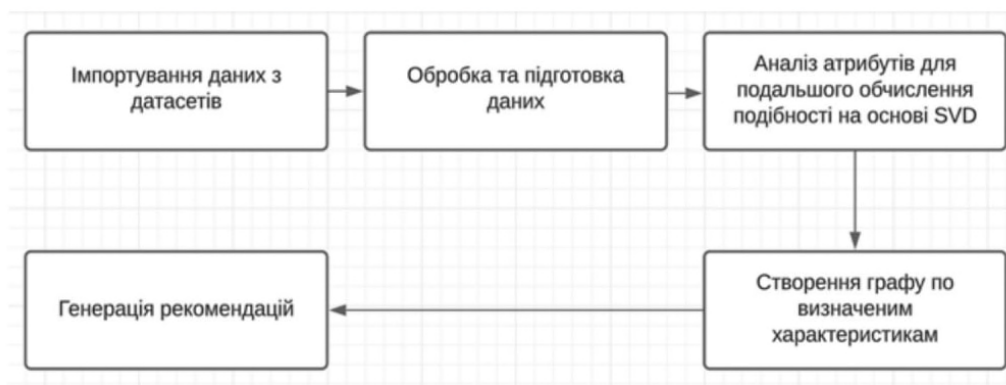


ЗАГАЛЬНА СХЕМА РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ



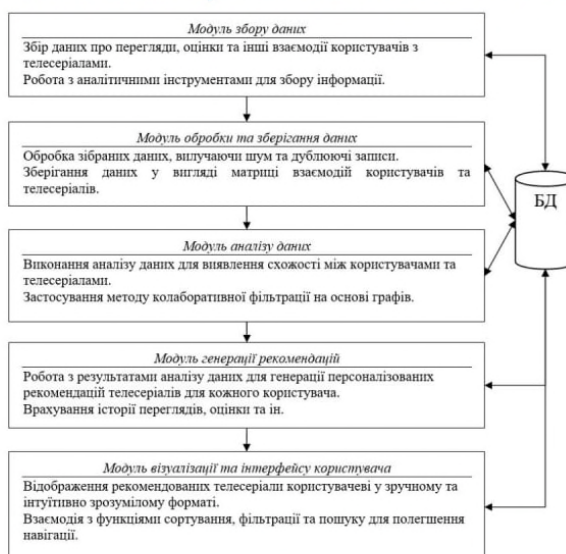
СХЕМА

ФОРМУВАННЯ РЕКОМЕНДАЦІЙ



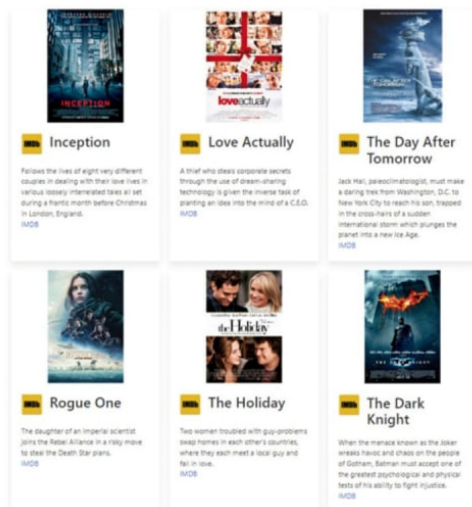
СХЕМА

ІНФОРМАЦІЙНОЇ СИСТЕМИ



ПРОГРАМНА

РЕАЛІЗАЦІЯ



ТЕСТУВАННЯ

Тестування відбувалось для декількох користувачів, для одного з яких система рекомендувала телесеріали, які відповідали уподобанням користувача, серед яких були телесеріали «Buffy the Vampire Slayer», «The Lord of the Rings», «Forrest Gump», «Game of Thrones», «Fight Club».

Метрики precision та recall для цього випадку становили 0.3 та 1 відповідно, що підтвердили правильність рекомендацій. Результат тестування наведено у тесті 1.

ТЕСТУВАННЯ 1

```

Three Colors: Red
Forrest Gump
A Close Shave
Night of the Living Dead
The African Queen
The Princess Bride
Rain Man
Fight Club
Memento
The Departed

```

ТЕСТУВАННЯ

У другому тестовому випадку система також показала високу ефективність, рекомендуючи телесеріали, які відповідали уподобанням користувача, які включали в себе телесеріали «What We Do in the Shadows», «Sleepy Hollow», «Night of the Living Dead», «The Princess Bride», «Game of Thrones», «The Haunting of Hill House».

Метрики precision та recall для цього випадку також становили 0.3 та 1 відповідно. Результат тестування наведено у тесті 2.

Рекомендаційна система успішно пройшла тестування, що дозволило уникнути потенційних помилок, які могли б завадити адекватному використанню.

ТЕСТУВАННЯ 2

```

Stranger Things
Supernatural
Night of the Living Dead
The Princess Bride
Game of Thrones
Shadow and Bone
The Thing
Hannibal
The Haunting of Bly Manor
Pulp Fiction

```

ВИСНОВКИ

В результаті виконання кваліфікаційної роботи бакалавра було розроблено рекомендаційну систему колаборативної фільтрації, яка персоналізовано підбирає телесеріали, аналізуючи великі обсяги даних про перегляди та відгуки користувачів. Робота включала аналіз систем рекомендацій на основі колаборативної фільтрації, розробку власної рекомендаційної системи з використанням колаборативної фільтрації для формування персоналізованих рекомендацій, створення функціональної та інформаційної структури вебсистеми перегляду телесеріалів, програмну реалізацію вебсистеми підбору телесеріалів з використанням рекомендаційної системи на основі колаборативної фільтрації, а також тестування рекомендаційної системи підбору телесеріалів.



ДЯКУЮ ЗА УВАГУ

Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 2.0%

Словники перевірки: en_US, ru_RU, ua_UA. Помилки в документах: 12%

| | | | | |
|---|----------|---------|-----------------------------|---------|
| ID: 130346 Назва: КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА на тему Рекомендаційна система підбору телесеріалів з використанням колаборативної фільтрації для вебсистем Додано в БД: 2024-06-13 Автора: Олександр ПАСІЧНИК Керівники: Руслан БАГРІЙ Консультанти: Опоненти: | Документ | | Сумарний збіг по Базі Даних | |
| | Символи | Лексеми | Символи | Лексеми |
| | 77734 | 1188 | 2538 (3%) | 38 (3%) |

Джерело плагіату

| | | | |
|----|------|--------------------------------|---------|
| ID | Опис | Наявність плагіату в документі | |
| | | Символи | Лексеми |

Ім'я користувача:
Кафедра КН

ID перевірки:
1016358198

Дата перевірки:
13.06.2024 21:33:22 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
13.06.2024 21:51:28 EEST

ID користувача:
100005671

Назва документа: КН-20-2 Пасічник_ЗАПИСКА

Кількість сторінок: 70 Кількість слів: 11975 Кількість символів: 97298 Розмір файлу: 2.16 MB ID файлу: 1016162677

6.76% Схожість

Найбільша схожість: 3.03% з джерелом з Бібліотеки (ID файлу: 1016146452)

4.86% Джерела з Інтернету

592

Сторінка 72

4.12% Джерела з Бібліотеки

160

Сторінка 76

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0% Вилучень

Немає вилучених джерел

**РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ КАФЕДРИ КОМП'ЮТЕРНИХ НАУК
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ**

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Рекомендаційна система підбору телесеріалів з використанням колаборативної фільтрації для вебсистем

Автор: студент гр. КН-20-2 Олександр Пасічник

Спеціальність: 122 – Комп'ютерні науки

Освітня програма: освітньо-професійна

Науковий керівник: к.т.н., доц. Руслан Багрій

Після аналізу звіту подібності зроблено такий висновок:

| № | Висновок | Позначка про відповідність |
|---|---|----------------------------|
| 1 | Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту. | відповідає |
| 2 | Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи | |
| 3 | Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат. | |
| 4 | Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту. | |

Підтвердження:

Обсяг запозичень, визначений системами виявлення збігів/ ідентичності/схожості, складає:

- 1) за системою Anti-Plagiarism виявлені 2% є фрагментарними – містять поширені конструкції, загальновідомі терміни, скорочення та визначення.
- 2) За системою UNICHECK виявлені 6,76%, що є запозиченнями, які розміщені в розділах аналізу існуючих технологій та прототипів, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ ідентичності/схожості, складає 2% і 6.76% відповідно, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи

Гарант ОП

Завідувач кафедри КН

Руслан БАГРІЙ

Олександр МАЗУРЕЦЬ

Олександр БАРМАК



**ВІДГУК НАУКОВОГО КЕРІВНИКА
на кваліфікаційну роботу бакалавра**

студента гр. КН-20-2 Пасічника Олександра Олександровича

за темою Рекомендаційна система підбору телесеріалів з використанням колаборативної фільтрації для вебсистем

1. Актуальність теми

Актуальність теми достатньо обґрунтована, оскільки сучасний користувач стикається з величезною кількістю контенту, і саме рекомендаційні системи допомагають швидко та ефективно знаходити релевантні телесеріали, що відповідають його вподобанням. Колаборативна фільтрація є одним із найбільш ефективних підходів для розв'язання цієї задачі, адже вона враховує не лише вподобання конкретного користувача, а й смаки інших користувачів з подібними уподобаннями, що дозволяє підвищити точність та персоналізацію рекомендацій.

2. Відповідність роботи предметній області Стандарту спеціальності 122 Комп'ютерні науки

Теми кваліфікаційної роботи "Рекомендаційна система підбору телесеріалів з використанням колаборативної фільтрації для вебсистем" відповідає предметній області спеціальності 122 Комп'ютерні науки та вимогам до кваліфікаційної роботи бакалавра, оскільки результатом роботи є рекомендаційна система, що використовує метод колаборативної фільтрації для аналізу користувацьких вподобань та надання персоналізованих рекомендацій. Система базується на алгоритмах, які дозволяють визначати схожість між користувачами, використовуючи дані про попередні оцінки та перегляди. При вирішенні поставленої задачі використано методи збору та аналізу інформації, метод колаборативної фільтрації, технології та методи проектування вебсистем перегляду телесеріалів.

3. Професійні та особистісні якості бакалавра

Пасічник О.О. під час роботи над кваліфікаційною роботою бакалавра продемонстрував розуміння теорії та практичних аспектів використання методів колаборативної фільтрації для створення рекомендаційних систем. Також він успішно реалізував вебсистему, яка інтегрує систему рекомендацій для ефективного підбору

телесеріалів, показуючи вміння працювати з сучасними технологіями програмування та базами даних.

4. Ступінь самостійності під час виконання кваліфікаційної роботи

Робота виконана самостійно, академічного плагіату не виявлено, стосовно всіх запозичень наведено відповідні посилання на джерела.

5. Ступінь оволодіння методами дослідження

При реалізації кваліфікаційної роботи показала високий рівень компетентностей та володіння необхідними інструментами та обладнанням, методами, методиками та технологіями предметної області комп'ютерних наук.

6. Повнота та якість розкриття теми роботи

Тема роботи повністю розкрита, проведено аналіз актуальності та відомих досліджень в межах обраної теми, виконані усі поставлені задачі та розроблено програмну реалізація для підтвердження запропонованого методу.

7. Логічність, послідовність, аргументованість, літературна грамотність викладення матеріалу

Викладення матеріалу логічне, послідовне та аргументоване. Мова і стиль викладення кваліфікаційної роботи відповідають стандартам, що забезпечують доступність сприймання матеріалу і відповідає вимогам до сучасних наукових робіт.

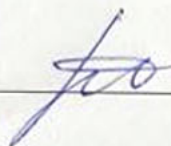
8. Можливість практичного застосування кваліфікаційної роботи бакалавра, окремих її частин

Розроблена у роботі вебсистема з рекомендаціями телесеріалів має значний потенціал для практичного застосування як цілісного продукту, так і окремих її компонентів. Вона може бути інтегрована у стрімінгові платформи для покращення користувацького досвіду за рахунок персоналізованих рекомендацій, що базуються на колаборативній фільтрації.

9. Висновок про можливість допуску кваліфікаційної роботи бакалавра до захисту, на яку оцінку заслуговує робота

Враховуючи високий рівень виконання та забезпечення усіх необхідних вимог, робота може бути допущена до захисту. Рекомендована оцінка «відмінно».

Керівник _____



к.т.н., доц. Руслан БАГРІЙ



РЕЦЕНЗІЯ

на кваліфікаційну роботу бакалавра

студента гр. КН-20-2 Пасічник Олександр Олександрович

за темою: Рекомендаційна система підбору телесеріалів з використанням колаборативної фільтрації для вебсистем

1. Актуальність обраної теми

В умовах сучасного інформаційного перевантаження, коли користувачі мають доступ до надмірної кількості контенту, важливо мати зручні інструменти для пошуку потрібної інформації. Рекомендаційна система підбору телесеріалів з використанням колаборативної фільтрації допомагає покращити користувацький досвід, роблячи взаємодію з технологіями більш зручною та ефективною. Аналізуючи дані про перегляди, реакції та відгуки глядачів, штучний інтелект може пропонувати персоналізований контент та прогнозувати майбутні тенденції.

2. Повнота розкриття мети та завдань роботи

Під час виконання кваліфікаційної роботи бакалавра було реалізовано рекомендаційну систему підбору телесеріалів з використанням колаборативної фільтрації для вебсистем, що відповідає меті та завданням кваліфікаційної роботи і розкриває їх повною мірою.

3. Зміст кожного розділу роботи

Записка кваліфікаційної роботи складається з трьох розділів. У першому розділі проводиться аналіз предметної області, включаючи аналіз інформаційних моделей, огляд систем та визначення мети і завдань. Другий розділ присвячений дослідженню методу формування рекомендацій на основі колаборативної фільтрації, з оглядом датасету. Третій розділ акцентує увагу на виборі засобів розробки, описі програмної реалізації рекомендаційної системи підбору телесеріалів та проведенні тестування.

4. Оцінка розробленої інформаційної системи, її практична цінність

Розроблена рекомендаційна система підбору телесеріалів надає інструменти, які допомагають користувачам ефективно знаходити контент, що відповідає їхнім вподобанням у середовищі надлишкової інформації.

5. Якість оформлення кваліфікаційної роботи бакалавра

Записка якісно оформлена, відповідно до встановлених вимог. Вірно та зрозуміло написана, з виразною структурою і логічною послідовністю представлення матеріалу.

6. Недоліки кваліфікаційної роботи бакалавра

Рекомендовано розглянути можливість інтеграції рекомендаційної системи підбору телесеріалів на окремому хостингу та забезпечення її взаємодії з сервісами перегляду.

7. Загальний висновок (допускається чи не допускається до захисту), та оцінка на яку заслуговує кваліфікаційна робота.

Враховуючи рівень виконання та забезпечення усіх необхідних вимог, робота може бути допущена до захисту. Рекомендована оцінка відмінно.

Рецензент

Марчишин В.В.

д.т.н., проф. зав.каф. АКИТ та Р.