МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Харківський національний університет радіоелектроніки
Факультет Комп'ютерних наук
Кафедра Програмної інженерії

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

другий (магістерський)
(рівень вищої освіти)

Дослідження типів колаборативної фільтрації для побудови прогнозів у рекомендаційних системах

Виконав:
Студент(ка) 2 курсу, групи ІПЗм-20-1
Газнозій Д.Ю.
(прізвище, ініціали)

Спеціальність	121 – Інженерія програмного забезпечення
Тип програми	освітньо – наукова
Керівник	доц. Голян В.В.

Допускається до захисту
Зав. Кафедри _____ З.В. Дудар

2022 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук

Кафедра Програмної інженерії

Рівень вищої освіти другий (магістерський)

Спеціальність 121 – Інженерія програмного забезпечення
(код і повна назва)

Тип програми освітньо-наукова програма

Освітня програма Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

«__»_____202_ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студента Газнозія Дмитра Юрійовича
(прізвище ім'я по батькові студента)

1. Тема роботи «Дослідження типів колаборативної фільтрації для побудови прогнозів у рекомендаційних системах»

затверджена наказом університету від «__»_____202_ р. №___

2. Термін подання студентом роботи до екзаменаційної комісії «15» травня 2022 р.

3. Вихідні дані до роботи колаборативна фільтрація, система рекомендацій, типи колаборативної фільтрації.

4. Перелік питань, що потрібно опрацювати в роботі мета роботи, аналіз предметної галузі та постановка задачі, конкуренти, опис ідеї і алгоритму, проектування і розробка програмного модуля, проектування і проведення експериментів, аналіз результатів.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Пошук інформації та аналогічних рішень	25.09.2021	виконано
2	Розробка плану дослідження	18.10.2021	виконано
3	Вибір датасету для дослідження	1.11.2021	виконано
4	Вибір моделі для дослідження	6.11.2021	виконано
5	Аналіз предметної галузі та постановка задачі	20.11.2021	виконано
6	Опис ідеї та алгоритму	22.12.2021	виконано
7	Проектування програмного модуля	14.01.2022	виконано
8	Розробка програмної системи	25.01.2022	виконано
9	Проектування і проведення експериментів	22.02.2022	виконано
10	Аналіз отриманих результатів	25.02.2022	виконано
11	Підготовка пояснювальної записки	20.04.2022	виконано
12	Захист роботи	20.05.2022	

Дата видачі завдання _____ 202 р.

Студент _____
(підпис)

Керівник роботи _____ доц. Голян В. В.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ / ABSTRACT

Пояснювальна записка містить: 53 стор., 28 рис., 9 джерел.

ВПОДОБАННЯ, КОЛАБОРАТИВНА ФІЛЬТРАЦІЯ, РЕКОМЕНДАЦІЙНА СИСТЕМА.

Об’єктом дослідження є алгоритми та типи колаборативної фільтрації, що спрямовані на надання персоніфікованих рекомендацій.

Метою дослідження є аналіз типів колаборативної фільтрації для побудови прогнозів у рекомендаційних системах, їхній детальний огляд та порівняння ефективності роботи.

Дослідження, що будуть проведені, будуть базуватися на результатах роботи вже існуючих рекомендаційних систем, а також на теорії побудови рекомендаційних систем за допомогою методів колаборативної фільтрації.

Методи розробки базуються на мові програмування Python та різноманітних бібліотеках, а також на відкритому датасеті оцінок кінофільмів.

У результаті роботи реалізовані різні типи колаборативної фільтрації, досліджено залежність між вибором функції подібності та результатом точності рекомендаційного алгоритму, реалізовано базову рекомендаційну систему кінофільмів.

LIKES, COLLABORATIVE FILTRATION, RECOMMENDATION SYSTEM.

The object of research is algorithms and types of collaborative filtering, aimed at providing personalized recommendations.

The aim of the study is to analyze the types of collaborative filtering to build forecasts in recommendation systems, their detailed review and comparison of performance.

The research that will be conducted will be based on the results of the work of existing recommendation systems, as well as on the theory of construction of recommendation systems using the methods of collaborative filtering.

Development methods are based on the Python programming language and various libraries, as well as an open data set of movie ratings.

As a result of work different types of collaborative filtering are realized, the dependence between the choice of similarity function and the result of accuracy of the recommendation algorithm is investigated, the basic recommendation system of films is implemented.

Я, Газнозій Дмитро Юрійович
(прізвище, ім'я, по-батькові)
студент(ка) групи ІПЗм-20-1 здобувач вищої освіти на другому (магістерському) рівні

кафедра програмної інженерії,
(повна назва кафедри)

заявляю: моя кваліфікаційна робота на тему
Дослідження типів колаборативної фільтрації для побудови прогнозів у рекомендаційних системах,

що буде представлена до ЕК для публічного захисту, виконана самостійно, в ній не містяться елементи плагіату і вона може бути опублікована в електронному архіві відкритого доступу ElArKhNURE. Всі запозичення з друкованих та електронних джерел мають відповідні посилання.

Я ознайомлений(а) з діючим положенням «Про протидію академічному плагіату в ХНУРЕ», згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування дисциплінарних заходів.

ЗМІСТ

ВСТУП

Неможливо уявити сучасний світ без перегляду кінофільмів, серіалів, різноманітних шоу, прослуховування музики, читання різних книг тощо. Люди все частіше користуються популярними сервісами для надання таких послуг: Netflix, Youtube, Spotify. Але дедалі людина поглинає все більше контенту, тим прискіпливіша вона становиться до пошуку нових вражень, адже знайти те, що їй дійсно сподобається непроста задача.

Таким чином, постає питання, як знаходити ту інформацію або той контент, який може заінтересувати користувача, адже у дуже швидкому русі сучасного світу не завжди можна знайти час на пошук того що тобі подобається. До цієї ж думки прийшли розробники популярних платформ надання контенту, як, наприклад, стрімінговий сервіс Netflix [1].

Їхній найуспішніший алгоритм, Netflix Recommendation Engine (NRE), складається з алгоритмів, які фільтрують вміст на основі кожного окремого профілю користувача. Система фільтрує понад 3000 назв одночасно, використовуючи 1300 рекомендаційних кластерів на основі уподобань користувача. Це настільки точно, що 80% активності глядачів Netflix обумовлено персоналізованими рекомендаціями системи.

Netflix — не єдина компанія, яка використовує механізм рекомендацій. Amazon, LinkedIn, Spotify, Instagram, Youtube та багато інших веб-платформ використовують механізми рекомендацій, щоб передбачити вподобання своїх користувачів і підвищити їх бізнес. Одним із методів побудови рекомендаційних систем є метод колаборативної фільтрації.

Розвиток Інтернету значно ускладнив ефективне вилучення корисної інформації з усієї доступної інформації в Інтернеті. Переважна кількість даних потребує механізмів ефективної фільтрації інформації. Колаборативна фільтрація є одним із методів, які використовуються для вирішення цієї проблеми.

Колаборативна фільтрація — це метод автоматичного прогнозування (фільтрації) інтересів користувача шляхом збору вподобань або інформації про

смаки від багатьох користувачів (спільна робота). Колаборативна фільтрація має декілька підходів до реалізації. Саме дослідження різних підходів цього методу і є метою даної дослідницької роботи.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Аналіз предметної галузі

Аж до появи Інтернету покупці, які шукають потенційні товари для придбання, або слідували своїм вродженим уподобанням і смакам, або слідували за смаками людини, якій вони особливо довіряли. Можна стверджувати, що всі ми маємо природну рису, яка змушує нас класифікувати речі: це нам подобається; цього ми не робимо. І, в принципі, можна, хоча й складно, переглянути весь товар у магазині чи торговому центрі і вибрати той, який дійсно варто купити.

Парадигма «відбір шляхом перевірки» [2] творить чудеса, оскільки ставить високу планку між речами, які ми хотіли б розглянути далі, і тими, які ми можемо негайно відкинути. Однак існує фундаментальне припущення, що стоїть за успіхом цієї поведінкової моделі, яку варто пам’ятати: вона передбачає, що кількість елементів, які ми будемо сортувати, є керованою. Але якщо Інтернет чомусь навчив нас за останні двадцять років, так це тому, що інформація зростає в геометричній прогресії, і ми не можемо прочитати її всю.

Обмеження в кількості потенційних елементів, які можна переглянути в Інтернеті, практично не існує. Усі ми отримали сотні мільйонів звернень із пошуку Google або сторінки за сторінками із зображеннями, коли шукали ключове слово із зображенням. Переповнення інформації є звичним явищем у наш час. Ми вийшли за межі необхідності «перебувати там» для перегляду, оскільки фактичний фізичний об’єкт, зображений або описаний, зберігається лише як бітовий потік на сервері з високою доступністю.

У перші роки існування Інтернету, коли покупки в Інтернеті рухалися повільними темпами, спочатку здавалося, що стара парадигма «відбору шляхом огляду» працювала. Фізичний перегляд і відвідування вітрин були буквально переведені на стрибок з елементарного інтернет-магазину в інший. У мережі було небагато продуктів на вибір, і все ж менше веб-сайтів із створеними онлайн-магазинами.

Відтоді це було нескінченне експоненційне збільшення потоку інформації, що навіть на фізичному мережевому рівні спочатку були проблеми із забезпеченням. Але досить скоро обмеження швидкості, встановлене ранніми модемами, змінилося високошвидкісним ADSL або кабельним, і з цього моменту веб-сторінки змагалися одна з одною за відображення більше графіки, зображень і тексту. Зародилася інформаційна ера, і концепція інформаційного перевантаження стала загальновідомою [3].

Як онлайн-покупець повинен відфільтрувати величезну кількість варіантів, які пропонуються зараз? Завдання справді стало схожим на пошук голки в копиці сіна. Використання старомодного передінтернетного методу довіряти власному смаку чи смаку близького друга, здається, більше не допомагає. Видимість інформації значно зменшилася. Як відкривати нові приховані речі? Нові продукти? Нові ідеї?

Інтернет-магазини зіткнулися з питанням цільової рекомендації досить рано, можливо, розуміючи, що інформація з часом буде лише розширюватися, і що людям знадобиться допомога, щоб знайти потрібний товар. Їх пошуки привели їх до розробки алгоритмів, які могли б або дізнатися власний смак, або виміряти «подібність» між нашим смаком і смаком інших, і дати нам обґрунтовані рекомендації щодо того, що ми хотіли б переглянути далі.

Основна передумова, що лежить в основі їхньої передумови, досить проста, але потужна: кількісна оцінка та класифікація смаку може слугувати платформою, на якій їм можна запропонувати інші товари, які можуть сподобатися покупцям. Очевидно, що допомога покупцям у перегляді веб-сайтів, коли кількість товарів занадто велика, щоб охопити їх фізично, має на меті збільшити ймовірність перетворення випадкового погляду на розпродаж.

На сьогоднішний день важко представити систему яка надає контент користувачу, та не пропонує йому схожий матеріал або послугу [4]. Компанії, які використовують системи рекомендацій, зосереджуються на збільшенні продажів у результаті дуже персоналізованих пропозицій та покращеного досвіду клієнтів.

Зазвичай рекомендації прискорюють пошук і полегшують користувачам доступ до контенту, який їх цікавить, і дивують їх пропозиціями, які вони ніколи б не шукали.

Більше того, компанії можуть залучати та утримувати клієнтів, розсилаючи електронні листи з посиланнями на нові пропозиції, які відповідають інтересам одержувачів, або пропозиції фільмів і телешоу, які відповідають їхнім профілям. Користувач починає відчувати себе відомим і зрозумілим і з більшою ймовірністю купуватиме додаткові продукти або споживатиме більше контенту. Знаючи, чого хоче користувач, компанія отримує конкурентну перевагу, а загроза втрати клієнта від конкурента зменшується.

Забезпечити додану цінність користувачам шляхом включення рекомендацій у системи та продукти є привабливою можливістю. Крім того, це дозволяє компаніям випередити своїх конкурентів і в кінцевому підсумку підвищити свої прибутки.

Системи рекомендацій сьогодні є повсюдно поширеною технологією, яка виявилася неоціненною для роботи з експоненційним зростанням інформації. Новини, продукти, музика, пости — все це можна адаптувати, щоб скористатися перевагами відповідної сили алгоритму та персоналізованого профілю. Успіх технології можна легко побачити, вивчивши одне з найважливіших удосконалень Google у своїй пошуковій системі: відстеження історії перегляду. Вивчіть звички користувачів під час перегляду та використовуйте їх, щоб рекомендувати додавання для потенційних відвідувань веб-сайту.

В даний час спостерігається бурхливий зріст сервісів із впровадженою рекомендаційною системою. Системи рекомендацій – це системи, які призначені для того, щоб рекомендувати користувачеві речі на основі багатьох різних факторів. Ці системи передбачають найбільш імовірний продукт, який користувачі, швидше за все, придбають і який представляє інтерес.

Система рекомендацій має справу з великим обсягом наявної інформації, фільтруючи найважливішу інформацію на основі даних, наданих користувачем, та інших факторів, які враховують переваги та інтереси користувача. Він з’ясовує збіг

між користувачем і елементом і приписує схожість між користувачем та елементами для рекомендації.

До вимог ефективності роботи рекомендаційної системи можна віднести точність роботи та швидкість, адже як зазначають дослідження, сучасні користувачі негативно відносяться до затримок у роботі їх улюблених сервісів.

Зробимо огляд наявних аналогів.

1.2 Огляд наявних аналогів

До наявних аналогів можна віднести будь-яку платформу яка надає рекомендації на основі вподобань користувачів.

Netflix — це потоковий сервіс на основі підписки, який дозволяє користувачам дивитися телешоу та фільми без реклами на пристрої, підключеному до Інтернету. Їх продукт — це модель обслуговування за передплатою, яка пропонує персоналізовані рекомендації, щоб допомогти користувачам знайти серіали та фільми, які їх цікавлять. Для цього вони створили власну комплексну систему рекомендацій.

Щоразу, коли користувач отримуєте доступ до служби Netflix, їхня система рекомендацій прагне допомогти користувачу знайти шоу чи фільм, який буде довподоби відвідувачу. Вони оцінюють ймовірність того, що користувач перегляне певний фільм в їхньому каталозі, виходячи з ряду факторів, зокрема:

- взаємодія користувача з сервісом (наприклад, історія переглядів і те, як користувач оцінював інші фільми);
- інші учасники з подібними смаками та уподобаннями на сервісі (метод колаборативної фільтрації);
- інформація про фільми, такі як їх жанр, категорії, актори, рік випуску.

На додаток до вибору, які фільми включати в рядки на домашній сторінці користувача Netflix, їхня система також оцінює кожен фільм у рядку, а потім ранжує самі рядки, використовуючи алгоритми та комплексі системи, щоб забезпечити персоналізований досвід. Іншими словами, коли користувач дивиться на свою домашню сторінку Netflix, їхні системи ранжують назви таким чином, щоб

представити найкращий можливий порядок назв, які можуть сподобатися користувачу (див. рис. 1.1).

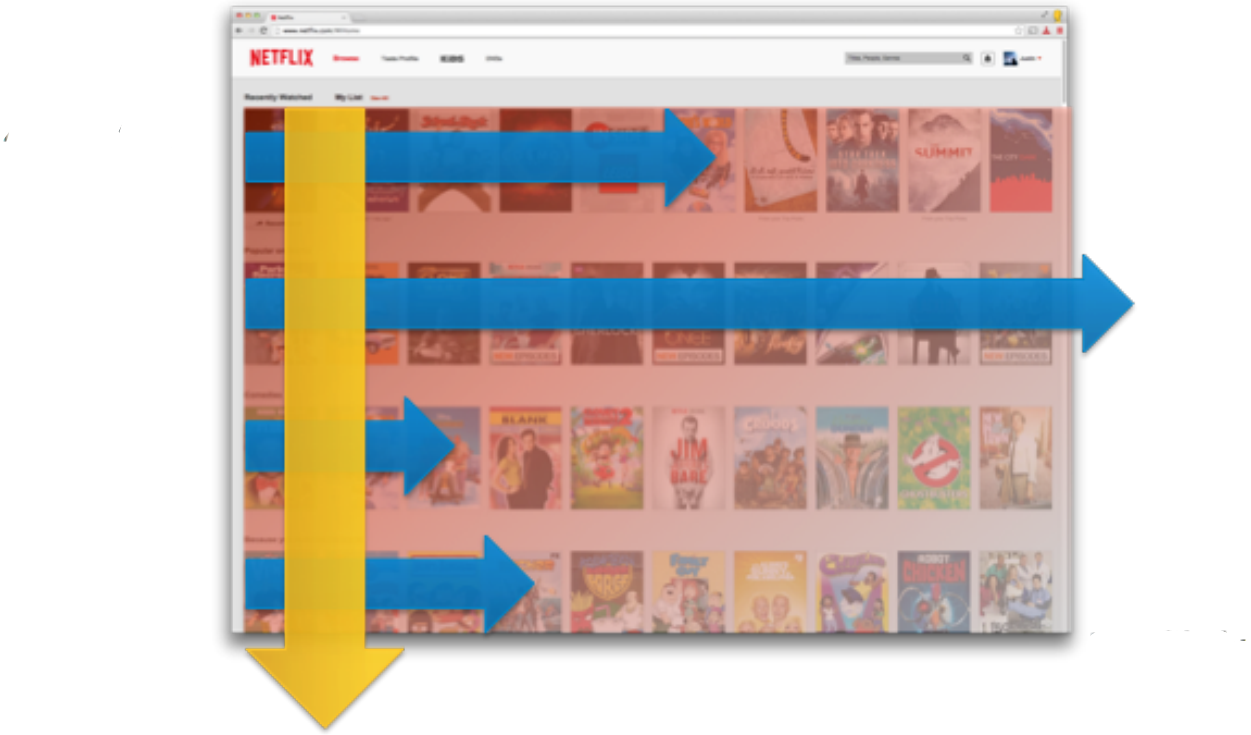

Рисунок 1.1 – Ранжована стрічка Netflix

Ще одна платформа яка використовує рекомендаційну систему це Spotify. Spotify — це сервіс цифрової потокової передачі музики, який надає користувачам доступ до мільйонів пісень, подкастів і відео від виконавців з усього світу. Даний сервіс надає рекомендації щодо пісень, які теоретично можуть сподобатися користувачу [5].

Рекомендаційна система даного сервісу агрегує два підходи для надання рекомендацій: фільтрацію на основі вмісту, метою якої є опис композиції шляхом вивчення самого вмісту та колаборитивну фільтрацію, метою якої є описати трек у його зв'язку з іншими треками на платформі шляхом вивчення створеною користувачами активності.

Сьогодні модель колаборативної фільтрації Spotify навчається на вибірці з приблизно 700 мільйонів створених користувачами списків відтворення, вибраних

із набагато ширшого набору всіх створених користувачами списків відтворення на платформі.

Можемо виконати постановку задачі.

1.3 Постановка задачі

Підбиваючи підсумки проведеного предметного аналізу та огляду наявних продуктів-аналогів можна сформувати специфікацію задачі, яка буде реалізована у даній кваліфікаційній роботі.

Як можна було побачити із проведеного аналізу, метод колаборативної фільтрації повсюдно використовується у рекомендаційних системах. Колаборативна фільтрація наразі є одним із найбільш часто використовуваних підходів і зазвичай забезпечує кращі результати, ніж рекомендації на основі вмісту.

Метод колаборативної фільтрації використовує взаємодію користувачів для фільтрації предметів, що цікавлять. Можна візуалізувати набір взаємодій за допомогою матриці, де кожен запис (i, j) представляє взаємодію між користувачем (i, i) та елементом (j, j). Цікавий спосіб поглянути на спільну фільтрацію – це розглядати її як узагальнення класифікації та регресії. Хоча в цих випадках ми прагнемо передбачити змінну, яка безпосередньо залежить від інших змінних (особливостей), у спільній фільтрації немає такої відмінності між змінними ознак і змінними класу.

Системи спільної фільтрації ґрунтуються на припущенні, що якщо користувачеві подобається елемент A, а іншому користувачеві подобається той самий елемент A, а також інший елемент, елемент B, перший користувач також може зацікавитися другим елементом. Отже, вони прагнуть передбачити нові взаємодії на основі історичних. Для досягнення цієї мети існують два типи методів: на основі користувачів та на основі предметів інтересу.

Таким чином, необхідно провести порівняльний аналіз даних методів колаборативної фільтрації, розглянути варіанти реалізацій, виявити слабкі та сильні сторони кожного із варіантів, а також можливість інтеграції колаборативної фільтрації з іншими рекомендаційми підходами.

2 ОПИС ПРИЙНЯТИХ ПРОЕКТНИХ РІШЕНЬ

2.1 Аналіз методів побудови рекомендаційних систем

Рекомендаційна система — це підклас системи фільтрації інформації, яка прагне передбачити «оцінку» або «перевагу», яку користувач надасть елементу. Існує декілька підходів до побудови рекомендаційного алгоритму, узагальнену схему методів для створення системи надання рекомендацій можна побачити на рис. 2.1.

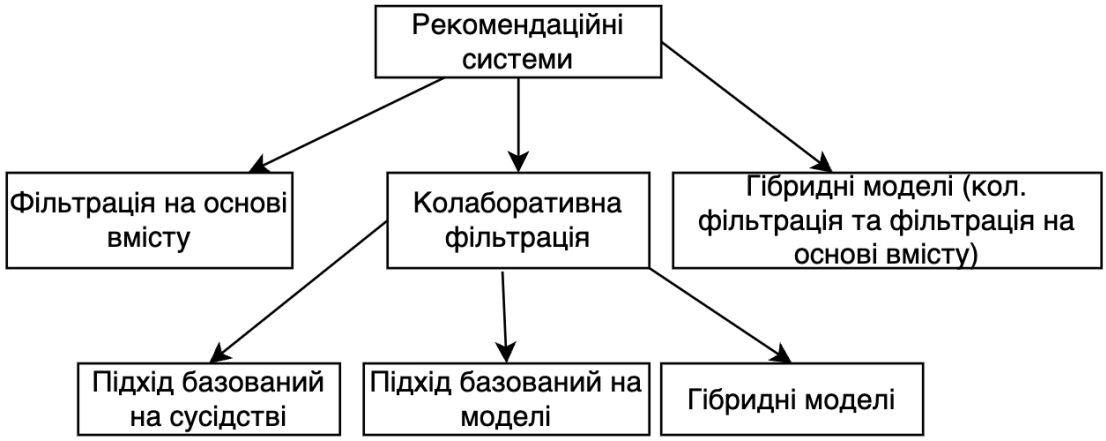

Рисунок 2.1 – Методи побудови рекомендаційних систем

Як можна побачити з даної схеми, системи рекомендацій зазвичай використовують одну або обидві фільтрації для співпраці та фільтрацію на основі вмісту (також відому як підхід на основі особистості), а також інші системи, такі як системи на основі знань. Підходи до колаборативної фільтрації будують модель на основі минулої поведінки користувача (предмети, які були раніше придбані чи вибрані та/або числові оцінки цих елементів), а також подібні рішення, прийняті іншими користувачами. Потім ця модель використовується для прогнозування елементів (або оцінок предметів), які можуть зацікавити користувача. Підходи до фільтрації на основі вмісту використовують серію дискретних, попередньо

позначених характеристик елемента, щоб рекомендувати додаткові елементи зі схожими властивостями.

Проведемо аналіз методу фільтрації на основі вмісту.

2.2 Аналіз методу фільтрації на основі вмісту

Методи фільтрації на основі вмісту засновані на описі елемента та профілю уподобань користувача. Ці методи найкраще підходять для ситуацій, коли є відомі дані про елемент (назва, місцезнаходження, опис тощо), але не про користувача. Рекомендації на основі вмісту розглядають рекомендацію як специфічну для користувача проблему класифікації та вивчають класифікатор уподобань користувача на основі особливостей елемента (див. рис. 2.2).

Рисунок 2.2 – Основна ідея методу фільтрації на основі вмісту

У таких системах ключові слова використовуються для опису елементів, а профіль користувача створюється, щоб вказати тип елемента, який подобається цьому користувачеві. Іншими словами, ці алгоритми намагаються рекомендувати предмети, подібні до тих, які подобалися користувачеві в минулому або розглядаються зараз. Він не покладається на механізм входу користувача для створення цього часто тимчасового профілю. Зокрема, різні предмети-кандидати

порівнюються з елементами, попередньо оціненими користувачем, і рекомендуються елементи, які найкраще відповідають. Цей підхід сягає корінням у дослідженнях пошуку інформації та фільтрації інформації.

Наприклад, для створення профілю користувача, система повинна фокусуватися на двох типах інформації: моделі уподобань користувача та на історії взаємодії користувача з системою рекомендацій.

В основному, ці методи використовують профіль елемента (тобто набір дискретних атрибутів і ознак), що характеризують об’єкт в системі. Для абстрагування особливостей елементів у системі застосовується алгоритм представлення предметів. Широко використовуваним алгоритмом є представлення tf–idf [6] (також зване представленням векторного простору).

Система створює профіль користувачів на основі вмісту на основі зваженого вектора характеристик елемента. Вагові коефіцієнти вказують на важливість кожної функції для користувача і можуть бути обчислені з індивідуально оцінених векторів вмісту за допомогою різноманітних методів. Прості підходи використовують середні значення вектора оціненого елемента, тоді як інші складні методи використовують методи машинного навчання, такі як байєсівські класифікатори, кластерний аналіз, дерева рішень та штучні нейронні мережі, щоб оцінити ймовірність того, що користувачеві сподобається елемент.

Ключовою проблемою фільтрації на основі вмісту є те, чи може система дізнаватися про переваги користувачів із дій користувачів щодо одного джерела вмісту та використовувати їх для інших типів вмісту. Коли система обмежена рекомендаціями вмісту того самого типу, який вже використовує користувач, значення системи рекомендацій значно менше, ніж коли можна рекомендувати інші типи вмісту з інших служб. Наприклад, корисно рекомендувати статті новин на основі перегляду новин. Проте, було б набагато корисніше, коли на основі перегляду новин можна рекомендувати музику, відео, продукти, обговорення тощо з різних сервісів. Щоб подолати це, більшість рекомендаційних систем на основі контенту тепер використовують певну форму гібридної системи, про яку згадувалося вище.

Системи рекомендацій на основі вмісту також можуть включати системи рекомендацій на основі думок. У деяких випадках користувачам дозволяється залишати текстові огляди або відгуки про товари. Ці створені користувачами тексти є неявними даними для системи рекомендацій, оскільки вони є потенційно багатими ресурсами як характеристик/аспектів товару, так і оцінки/позиці користувачів до цього елемента. Функції, отримані з створених користувачами оглядів, є покращеними метаданими елементів, оскільки вони також відображають такі аспекти елемента, як метадані, витягнуті функції викликають широке занепокоєння користувачів. Настрої, отримані з оглядів, можна розглядати як оцінки користувачів за відповідними функціями. Популярні підходи системи рекомендацій на основі думок використовують різні методи, включаючи аналіз тексту, пошук інформації, аналіз настроїв і глибоке навчання.

Проведемо аналіз підходу колаборативної фільтрації.

2.3 Аналіз підходу колаборативної фільтрації

Колаборативна фільтрація має два значення: вузький і більш загальний. У новішому, більш вузькому сенсі, спільна фільтрація — це метод автоматичного прогнозування (фільтрації) щодо інтересів користувача шляхом збору вподобань або інформації про смаки від багатьох користувачів (спільна робота). Основне припущення підходу спільної фільтрації полягає в тому, що якщо людина A має таку ж думку, що й особа B, щодо проблеми, A, швидше за все, матиме думку B щодо іншого питання, ніж думка випадково обраної людини. Наприклад, рекомендаційна система спільної фільтрації для переваг у телевізійних програмах може робити прогнози щодо того, яке телешоу має сподобатися користувачеві, з огляду на частковий список уподобань цього користувача (подобається чи не подобається). Необхідно зауважити, що ці прогнози є специфічними для користувача, але використовують інформацію, отриману від багатьох користувачів. Це відрізняється від простішого підходу, коли надається середня (неспецифічна) оцінка для кожного пункту, що цікавить, наприклад, на основі його кількості голосів.

У більш загальному сенсі, спільна фільтрація — це процес фільтрації інформації або шаблонів з використанням методів, що передбачають співпрацю між кількома агентами, точками зору, джерелами даних тощо. Застосування спільної фільтрації зазвичай включають дуже великі набори даних. Колаборативні методи фільтрації були застосовані до багатьох різних видів даних, включаючи: дані зондування та моніторингу, наприклад, при розвідці корисних копалин, зондування навколишнього середовища на великих площах або численні датчики; фінансові дані, наприклад установи фінансових послуг, які об'єднують багато фінансових джерел; або в електронній комерції та веб-додатках, де акцент зосереджений на даних користувача тощо.

Мотивація для спільної фільтрації походить від ідеї, що люди часто отримують найкращі рекомендації від когось із схожими на них смаками. Алгоритми спільної фільтрації часто вимагають активної участі користувачів, простого способу представлення інтересів користувачів і алгоритмів, які можуть знайти людей зі схожими інтересами.

Як правило, робочий процес системи колаборативної фільтрації наступний:

– користувач висловлює свої уподобання, оцінюючи елементи системи (наприклад, книги, фільми чи музичні записи). Ці рейтинги можна розглядати як приблизне відображення інтересу користувача до відповідного домену;
– система порівнює оцінки цього користувача з оцінками інших користувачів і знаходить людей з найбільш «схожими» смаками;
– з подібними користувачами система рекомендує товари, які подібні користувачі високо оцінили, але ще не оцінені цим користувачем (відсутність оцінки часто розглядається як незнайомство товару).;

Основну ідею методу колаборативної фільтрації можна побачити на рис. 2.3.

Рисунок 2.3 – Основна ідея методу колаборативної фільтрації

Проведемо аналіз типів колаборативної фільтрації базованої на пам’яті.

2.4 Аналіз типів колаборативної фільтрації базованої на пам’яті

Існують два різних типи алгоритмів спільної фільтрації на основі пам’яті, а саме на основі користувача та на основі елемента, залежно від того, який параметр оцінки користувацьких елементів матриця використовує для пошуку подібності. Кожна з цих двох стратегій має плюси і мінуси.

Алгоритми на основі користувачів шукатимуть «однодумців», виходячи з припущення, що схожі користувачі люблять подібні предмети. Алгоритми на основі елемента шукатимуть предмети однаково оцінені різними користувачами, виходячи з припущення, що якби оцінило багато користувачів два предмети однаково, вони, ймовірно, будуть подібними та заслуговують на рекомендацію; знову ж таки, схожі товари, як правило, подобаються подібним користувачам.

Друга стратегія, також названа рекомендацією по елементам, була вперше запропонована у [7] і розроблено Amazon.com як частину їхньої внутрішньої системи рекомендацій. Метою підходу на основі елемента є виправлення того, що

вважається важливим недоліком алгоритму на основі користувача на реальній практиці: він погано масштабується.

Обидва методи повинні ідентифікувати N-найближчих сусідів цільової вибірки, будь то користувач або елемент. Однак у базах даних, як правило, набагато більше користувачів, ніж у елементах, і тому пошук у вимірі користувачів набагато важчий.

Іншими словами, у реальних програмах користувачі, як правило, взаємодіють (і, отже, змінюються) набагато швидше, ніж елементи; останні, як правило, досить фіксовані для даного інтернет-магазину або спільноти.

Недолік масштабованості підходу на основі користувача не завжди є центральною проблемою під час вибору одного алгоритму перед іншим. Насправді це рішення пов’язане набагато більше з природою домену. Наприклад, [8] стверджує, що в контексті новин розмір елемента змінюється набагато швидше, ніж база користувачів, і тому система рекомендацій має віддавати перевагу підходу на основі користувача.

Стверджувати, що подібні користувачі придбали товар, менш переконливо, ніж стверджувати, що даний товар рекомендований, оскільки він подібний до інших товарів, придбаних у минулому.

Виконаємо аналіз оцінок користувачів у колаборативній фільтрації.

2.5 Аналіз оцінок користувачів у колаборативній фільтрації

Системи рекомендацій спільної фільтрації покладаються на рейтинги для надання рекомендацій. Ці уподобання можуть мати дві форми: вони можуть бути явними, як у випадку, коли користувач призначає оцінку товару, або неявними, як у випадку, коли система робить висновок, що користувачеві або подобається, або не подобається продукт, вивчаючи їхні дії, наприклад перегляд покупок або кліків на екрані.

У контексті спільної фільтрації рейтинги — це числове значення, яке відображає, наскільки користувачеві подобається чи не подобається певний

елемент, і є основою для вимірювання схожості, що використовується для пошуку однодумців. Існує явний та неявний рейтинги.

Явна оцінка, яка виникає, коли користувач дає пряму оцінку товару, очевидно, має величезну цінність, оскільки системі фактично повідомляють, що користувач відчуває до предмета, про який йде мова. Однак людська психологія є складною, і висновок про смак з розмов не такий послідовний, як можна було б сподіватися. Користувач може високо оцінити певний продукт, але з невідомої причини вирішить поставити низьку оцінку. Соціальний тиск. Вірусні тенденції. Усі ці зовнішні фактори впливають на оцінку користувачами та на рекомендації системи.

Більше того, шкала рейтингів, навіть якщо вона зазвичай стандартизована, може сприйматися різними виборцями по-різному. Двоє людей, яким дуже подобається фільм, можуть оцінити його абсолютно по-різному, тільки тому, що один більш щедрий у оцінці, а другий постійно ставить нижчу, щоб уникнути знецінення.

Ще одна проблема з явним механізмом оцінки - це пам'ять. Зазвичай користувачі оцінюють предмет близько до моменту, коли предмет був перевірений; щойно прочитана книга або фільм, який зараз переглянули. Ці предмети набагато свіжіші в пам’яті в порівнянні з іншими предметами, які були переглянуті в минулому. Це заважає користувачеві бути об’єктивним і релятивізувати оцінку до шкали, яка використовується в інших випадках.

Як наслідок, оцінка «відмінно» для конкретного фільму, переглянутого місяць тому, цілком могла б не витримати конкуренції нового фільму, переглянутого зараз, але в рівній мірі мати оцінку «відмінно». І оскільки користувачі, як правило, не змінюють стару оцінку, обидва елементи тепер мають однаковий рівень оцінки. Але навіть якщо користувач добре пам’ятає старий предмет, смак змінюється з часом. І якщо рейтинги не відкоригувати з урахуванням нових уподобань, старі й надалі впливатимуть на будь-яку рекомендацію системи, що зробить її менш ефективною.

Неявні оцінки намагаються мінімізувати вплив людської психології, не залучаючи користувача до отримання оцінки. Висновок уподобань менш упереджений, оскільки на них не впливає те, що, на думку користувача, йому слід оцінити. Однак виведена система не позбавлена проблем. Одна з поширених стратегій вгадати смак користувача — це зробити висновок, вивчаючи, скільки часу він переглядав певний елемент; але як алгоритм дізнається, чи дійсно користувач вивчає предмет із згодою чи незгодою?

Сторінки перегляду та придбані предмети також використовувалися як засіб для визначення смаку. Обгрунтування полягає в тому, що придбаний товар є предметом, який сподобався, а відвідана сторінка — це сторінка, яка представляє інтерес. Природно, можна купити товар для друга і опинитися на сторінці, просто натиснувши посилання помилково. Крім того, в деяких випадках кілька людей переглядають з одного облікового запису, створюючи ефективно попурі смаків з невеликою цінністю націлювання.

Основна перевага неявних оцінок перед явними полягає в тому, що користувача не турбує необхідність відповідати на запитання, щоб навчити алгоритм. Однак обидві стратегії можна легко поєднати, щоб покращити знання смаків конкретного користувача.

Оглянемо шкали оцінок у колаборативній фільтрації.

2.6 Шкали оцінок у колаборативній фільтрації

На відміну від системи рекомендацій на основі вмісту, в якій вміст елемента видобувається на схожість із вмістом інших елементів, у колаборативній фільтрації пошук схожості є суб’єктивними вподобаннями користувачів, і, таким чином, вимагає певної форми показника для використання для того, щоб прирівняти їх. Загалом вони мають одну з трьох форм: унарний (гарний або куплений), бінарний (подобається або не подобається), числовий (певна оцінка).

Унарна шкала використовується для сигналу про те, що користувач виявив інтерес до певного елемента. У ній нічого не розкривається про елементи, які

користувачеві не подобаються, але вона ненав’язлива, і в багатьох випадках її достатньо, щоб дізнатися про користувача.

Бінарна шкала є явною і має форму двох кнопок, одна з яких передає позитивне відчуття, а інша – негативне. Приклад такої шкали можна знайти на YouTube.com, де користувачі можуть натиснути піктограму «великий палець вгору» або «великий палець униз». Вони створюють загальні відчуття щодо предмета без особливої гами, однак вони є хорошими поляризаторами з мінімальною нав’язливістю.

Числова шкала подібна до рейтингу зірок готелю або ресторану, а також є явною. Вона дає найбільший спектр для вираження смаку, оскільки дозволяє користувачеві висловити себе не тільки в загальному напрямку. Ця оцінка зазвичай використовується при оцінці фільмів або книг.

Оглянемо нормалізацію оцінок.

2.7 Нормалізація оцінок

Одна з проблем, з якою стикається використання рейтингів як засобу представлення смаку, полягає в тому, що кожен користувач має особисту інтерпретацію шкали. У той час як один оцінювач може давати високі оцінки фільмам, які йому подобаються, інший оцінює найвищі оцінки за виняткові фільми. Тоді виникає питання, наскільки обидва користувачі схожі та як правильно використовувати їхні оцінки.

Існують дві широко використовувані системи для компенсації варіацій у підході до оцінки різними користувачами, одна називається середньоцентруванням, а інша – Z-оцінкою [9].

Алгоритм центрування середнього значення повторно відображає оцінку користувача, віднімаючи середнє значення всіх його оцінок, фактично показуючи, чи є конкретна оцінка позитивною чи негативною в порівнянні із середнім. Позитивні значення представляють оцінки вище середнього, негативні – нижчі за середні, а нуль – середній рейтинг.

Вираховується за наступною формулою:

$$h(r_{ui}) = r_{ui} - r'_u,$$

де $h(r_{ui})$ – являє собою середньоцентровану оцінку користувача u для елемента i;

r_{ui} – фактичний рейтинг користувача u для елемента i;

r'_u – представляє середню оцінку користувача u за всіма оціненими елементами.

Підхід з оцінкою Z дуже схожий на центрування середнього значення, але він нормалізується стандартним відхиленням σu або σi залежно від того, застосовуємо ми підхід на основі користувача чи елемента:

$$h(r_{ui}) = \frac{r_{ui} - \bar{r}_u}{\sigma_u}$$

Нормалізація рейтингів супроводжується застереженням. Якщо користувач поставив високі оцінки лише всім перевіреним елементам, то середньоцентрований підхід дасть їм середнє значення, а будь-яка оцінка нижче найвищої буде негативною, навіть якщо вона досить висока за шкалою. Аналогічно, якщо користувач оцінив усі предмети з однаковою точною оцінкою, стандартне відхилення буде нульовим, а Z-показник не можна буде обчислити.

Проаналізуємо функції подібності.

2.8 Функції подібності

В основі алгоритму колаборативої фільтрації лежить сильне припущення, що «подібним користувачам подобаються подібні елементи». Хоча це припущення, здається, в значній мірі справедливе, ключовим питанням стає: як можна визначити та ідентифікувати подібних користувачів, щоб знайти схожі кандидатури, які ми можемо рекомендувати? Будь-яке таке формулювання має враховувати наявні у нас дані, розріджену матрицю налаштувань елемента користувача, де багато

користувачів, ймовірно, оцінили лише невелику частку багатьох доступних елементів.

Яку б функцію подібності не використовувала система рекомендацій, важливо пам’ятати, що кожен користувач оцінює товари по-своєму і що рейтинги подібності поширених популярних елементів не обов’язково відображають подібність суперечливих елементів. що в кінці може мати більше значення в рекомендації.

Оглянемо кореляцію Пірсона.

2.8.1 Кореляція Пірсона

Функція подібності кореляції Пірсона є, мабуть, найвідомішим алгоритмом для систем рекомендацій на основі користувачів.

При використанні кореляції Пірсона подібність представлена шкалою від -1 до +1, де позитивне високе значення говорить про високу кореляцію, негативне високе значення означає обернену високу кореляцію (коли один каже «Правда», інший каже «Неправда»), і, нарешті, нульова кореляція вказує на некорельовані вибірки.

Рівняння подібності кореляції Пірсона на основі користувачів визначається як:

$$s(u,v) = \frac{\sum\limits_{i \in \mathcal{I}_{uv}} (r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\sqrt{\sum\limits_{i \in \mathcal{I}_{uv}} (r_{ui} - \bar{r}_u)^2 \sum\limits_{i \in \mathcal{I}_{uv}} (r_{vi} - \bar{r}_v)^2}},$$

де s(u, v) – представляє подібність користувачів u і v;

Iuv – являє собою набір елементів, оцінених користувачами u і v;

r_{ui} і r_{vi} – представляють рейтинг користувача u або v для елемента i;

$\bar{r}_u$ і $\bar{r}_v$ – представляють середню оцінку користувача u або v для всіх оцінених елементів.

У випадку систем рекомендацій на основі елементів, формула виглядає так:

$$s(i,j) = \frac{\sum_{u \in \mathcal{U}_{ij}} (r_{ui} - \bar{r}_i)(r_{ui} - \bar{r}_j)}{\sqrt{\sum_{u \in \mathcal{U}_{ij}} (r_{ui} - \bar{r}_i)^2 \sum_{u \in \mathcal{U}_{ij}} (r_{ui} - \bar{r}_j)^2}}$$

,

де s(i, j) – представляє подібність елементів і та j;

U_{ij} – представляє набір звичайних користувачів, які оцінили пункти і та j;

$\overline{r}_i$ та $\overline{r}_j$ – представляють середню оцінку елемента і або j для всіх користувачів, які оцінили товар.

Оглянемо косинус подібності.

2.8.2 Косинус подібності

Функція подібності косинусної відстані розглядає два зразки для порівняння та вивчає косинус кута між ними, щоб кількісно визначити подібність. Шкала, що використовується в цьому випадку, така ж, як і в кореляції Пірсона, де +1 і -1 представляють високу (пряму та зворотну) кореляцію, а нуль означає відсутність кореляції.

У векторній формі косинусна відстань визначається як:

$$cos(\vec{a}, \vec{b}) = \frac{\vec{a}.\vec{b}}{\|\vec{a}\|^2 * \|\vec{b}\|^2}$$

Рівняння подібності косинусів на основі користувача, визначене як підсумовування, стає:

$$s(u,v) = \frac{\sum\limits_{i \in \mathcal{I}_{uv}} r_{ui} r_{vi}}{\sqrt{\sum\limits_{i \in \mathcal{I}_u} r_{ui}^2 \sum\limits_{i \in \mathcal{I}_v} r_{vi}^2}},$$

де s(u, v) – представляє подібність користувачів u та v за всіма елементами, які зазвичай оцінюються.

У випадку рекомендаційних систем на основі елементів формулювання стає таким:

$$s(i,j) = \frac{\sum\limits_{u \in \mathcal{U}_{ij}} r_{iu} r_{ju}}{\sqrt{\sum\limits_{u \in \mathcal{U}_i} r_{iu}^2 \sum\limits_{u \in \mathcal{U}_j} r_{ju}^2}},$$

де s(i, j) – представляє подібність елементів i та j для всіх користувачів, які оцінили такі елементи.

Оглянемо евклідову відстань.

2.8.3 Евклідова відстань

Подібність евклідової відстані визначається як:

$$s(u,v) = \sqrt{\frac{\sum\limits_{i \in \mathcal{I}_{uv}} (r_{ui} - r_{vi})^2}{|\mathcal{I}_{uv}|}},$$

де s(u, v) – представляє подібність користувачів u та v за всіма елементами, які зазвичай оцінюються.

2.9 Метод найпопулярніших N-сусідів

Теоретично значення подібності всіх користувачів-кандидатів можуть використовуватися під час рекомендації елемента цільовому користувачеві, однак це не лише непрактичний підхід з великими базами даних користувачів, але й у

тому числі користувачів із низькою кореляцією до цільового користувача. перешкоджає рекомендації, оскільки збільшує її похибку.

У підході найпопулярніших N-сусідів найкращі N корельованих кандидатів відфільтровуються, а інші відфільтровуються та не використовуються на наступних етапах процесу рекомендації. Чим більше набір включених користувачів, тим більше, дисперсія в їхніх оцінках буде усереднена, але загальна якість передбачення знижується. Це також призводить до втрати випадковості, яка, як стверджується, є хорошою рисою колаборативної фільтрації, оскільки дає несподівані хороші рекомендації.

Ідеальна кількість користувачів для включення — це параметр, який потрібно встановити шляхом перехресної перевірки, оскільки він сильно залежить від набору даних. Одним із важливих результатів використання найпопулярніших N-сусідів є те, що система рекомендацій може пізніше підтримати рекомендацію, надавши докази через список сусідів. Повсякденне комерційне використання рекомендаційних систем навчило, що підтримка рекомендації дуже важлива, оскільки вона пояснює користувачеві, чому був рекомендований конкретний елемент, і розвіює відчуття, що йому пропонують щось абсолютно навмання.

Одна з проблем цієї техніки полягає в тому, що певна кількість сусідів навряд чи задовольнить кожного користувача в системі. Деяких користувачів може бути легше співвідносити з іншими, тоді як деяким іншим користувачам може знадобитися більше сусідів, щоб обчислити хороші прогнози, оскільки вони погано корелюють з більшістю користувачів у базі даних. Крім того, в деяких випадках збільшення кількості користувачів лише додасть шуму, тому що ці погано корельовані кандидати ефективно знижують якість рекомендацій.

Підхід порогової фільтрації має відповідати мінімальному задовільному рівню подібності для сусіда, який буде включено до набору вихідних даних. Ця стратегія вирішує проблему використання єдиного розміру сусідства для всіх кандидатів, але не позбавлена власних проблем.

Виконаємо проектування алгоритму побудови рекомендаційної системи.

2.10 Проектування алгоритму побудови рекомендаційної системи

Враховуючи вищезазначену інформацію, можемо спроектувати псевдокод для побудови рекомендаційної системи. Кроки алгоритму можна побачити на рис. 2.4.

1 **Input**: User-Item Rating matrix R
2 **Output**: Recommendation lists of size l
3 **Const** l: Number of items to recommended to user u
4 **Const** v: Maximum number of users in $\mathcal{N}(u)$, the neighbours of user u
5 **For** each user u **Do**
6 **Set** $\mathcal{N}(u)$ to the v users most similar to user u
7 **For** each item i that user u has not rated **Do**
8 Combine ratings given to item i by neighbours $\mathcal{N}_i(u)$
9 **End**
10 Recommend to user u the top-l items with the highest predicted rating $\hat{r}_{ui}$
11 **End**

Рисунок 2.4 – Псевдокод алгоритму колаборативної фільтрації

Алгоритм приймає як вхідні дані (рядок 1) розріджену матрицю User-Item з рейтингами R і виводить (рядок 2) список рекомендацій розміру l (рядок 3), який встановлюється відповідно до потреб системи. Параметр k (рядок 4), який зазвичай встановлюється за допомогою перехресної перевірки, диктує розмір околиці користувачів, які потрібно використовувати, щоб знайти схожих кандидатів на ціль.

Основна частина алгоритму прокручує список користувачів (рядок 5), і для кожного з них він знаходить k користувачів, найбільш подібних до цієї цілі (рядок 6). Подібні користувачі отримуються за допомогою функції подібності, причому найчастіше використовуються кореляція Пірсона та косинусна відстань. Вибір найбільш подібних користувачів здійснюється за допомогою алгоритму вибору сусідства, звичайним вибором є алгоритми найпопулярніших N-сусідів або підхід порогової фільтрації.

Потім для кожного об’єкту, який цільовий користувач не оцінив (рядок 7), ми об’єднуємо оцінки, надані сусідами-кандидатами (рядок 8), щоб отримати єдину оцінку кожної відсутньої оцінки. Нарешті (рядок 10), ми рекомендуємо цільовому користувачеві предмети, які отримали найвищу оцінку (рядки 7-9), припускаючи, що елементи, не оцінені користувачем, є предметами, про які користувач не знає.

Оглянемо типові проблеми алгоритмів колаборaційної фільтрації.

2.11 Типові проблеми алгоритмів колаборaційної фільтрації

Успішна рекомендація є скоріше функцією хорошого набору даних, ніж хорошого алгоритму чи стратегії. Без достовірних даних жоден алгоритм не може дати задовільних результатів, і за своєю природою проблема, з якою стикаються алгоритми спільної фільтрації.

Однією з найскладніших проблем, з якою потрібно впоратися під час створення рекомендацій, є нестача інформації. Користувач, як правило, оцінює лише кілька пунктів, і, як правило, система тримає у файлі велику кількість користувачів із невеликими оцінками для кожного. Іншими словами, вхідна матриця користувачів і елементів за своєю суттю дуже розріджена.

Це має глибокий вплив на систему рекомендацій. Може бути багато дійсно схожих користувачів-кандидатів на цільового користувача, але всі алгоритми схожості працюють, збігаючи оцінки, якими поділилися ці користувачі, і без оцінок подібні користувачі не будуть ідентифіковані.

Загальна стратегія вирішення проблеми розрідженості представляється як нова модель, де невідомі значення рейтингу замінюються середніми відомими оцінками, введеними користувачем. Але ця стратегія передбачає, що відсутні рейтинги ефективно моделюються середньою функцією, хоча насправді вони можуть бути нижче або вище відомих середніх рейтингів.

Розрідженість ускладнює консолідацію околиць подібних користувачів і, таким чином, впливає на охоплення алгоритмом, оскільки деякі цілі, ймовірно, не будуть зіставлені з достатньою кількістю кандидатів, з яких можна було б отримати рекомендацію. Насправді, чим щільніше набір даних, тим краща продуктивність.

В основі алгоритму спільної фільтрації лежить оцінка користувачів. Без оцінок жодна рекомендація не може мати місце для будь-якого користувача системи. Ця проблема називається холодним запуском і описує нездатність системи рекомендацій почати пропонувати рекомендації. Проблема, з якою стикаються стрибкові запуски цих систем, полягає в тому, що без рекомендацій інші користувачі, як правило, ухиляються і не рекомендують самі, змушуючи систему повністю зупинитися.

Один із способів порушити цю закономірність – змусити нових користувачів оцінити ряд елементів, перш ніж дозволити їм використовувати систему. Ця стратегія гарантує, що в систему завжди надходять нові оцінки, а також вирішує другу проблему, яка стосується кожного новачка: новому користувачеві, якому ще належить ввести власні налаштування, система взагалі нічого не рекомендує.

Хоча можна запропонувати узагальнені нецільові рекомендації новому користувачеві, це не є задовільним рішенням, оскільки потужність системи рекомендацій повністю втрачена. Пропонувати рейтинг, який працює для більшості, є повною протилежністю пошуку новизни та орієнтації рекомендацій на певний профіль користувача.

Відсутність нових оцінок користувачів – не єдина проблема в системі рекомендацій колаборативної фільтрації. Нові додані елементи також не матимуть жодних оцінок, і тому не будуть рекомендовані нікому.

Тут може допомогти стратегія примушування користувачів оцінювати новий елемент, однак, загалом, проблема розглядається як менш критична, ніж сценарій нового користувача, оскільки нові елементи, як правило, знаходять швидко, а оцінки слідують цьому прикладу. Хороша стратегія для використання з новими елементами — негайно виставляти їх на дисплей, щоб користувачі могли легко їх знайти та оцінити.

3 ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ

3.1 Огляд метрик оцінювання

Охоплення – це міра відсотка оцінок, які вдалося виконати алгоритму системи рекомендацій. Як правило, через вибір методу сусідства охоплення становить не 100%, а нижче значення.

Вибір типу стратегії сусідства для використання має важливий вплив на кількість оцінок, які може створити алгоритм. Наприклад, при використанні методики порогової фільтрації, чим вищий поріг, тим більше подібні вибірки, але менша околиця, що, у свою чергу, призводить до того, що деяка кількість вибірок не знаходить подібних кандидатів, а отже, їх нерейтинговані елементи не оцінюються.

Велике охоплення означає, що система рекомендацій може рекомендувати елементи багатьом користувачам, незалежно від їх особливостей (деяких користувачів важко знайти). Чим більше околиця, тим більше відхилень приймається, і тим гіршими стають рекомендації.

Точність є мірою продуктивності алгоритму. Він кількісно визначає, наскільки оцінки, вироблені стратегією, відхиляються від своїх відомих значень. Чим менше помилка, тим краще працює система рекомендацій. Точність тісно пов’язана з охопленням у тому сенсі, що коли один з них високий, інший має тенденцію бути низьким. Необхідно знайти компроміс між цими двома параметрами, щоб кожна система рекомендацій працювала найкраще.

У сучасності широко використовуються два типи вимірювань точності, це середня абсолютна помилка (MAE):

$$MAE(r, \hat{r}) = \frac{\sum\limits_{a \in \mathcal{N}_{ui}} |r_a - \hat{r}_a|}{\|r\|}$$

Середньоквадратична помилка (RMSE):

$$RMSE(r, \hat{r}) = \sqrt{\frac{\sum\limits_{a \in \mathcal{N}_{ui}} (r_a - \hat{r}_a)^2}{\|r\|}}$$

MAE та RMSE мають деякі недоліки, коли вони використовуються для вимірювання точності системи рекомендацій. Наприклад, якщо рейтинг недоступний, ця оцінка не відображається в помилці. Крім того, у випадку MAE помилка є лінійною, що означає, що розмір помилки однаковий на верхньому та нижньому кінцях подібності. RMSE більше штрафує більші помилки, що може мати сенс у контексті систем рекомендацій.

Обидві сусідні стратегії, при правильному налаштуванні відфільтровують низько корельовані вибірки від введення рекомендацій і знижують продуктивність алгоритму. Однак стратегії сусідства не враховують той факт, що не всі прийняті зразки однаково співвідносяться з цільовою метою. Деякі зразки мають більше спільних елементів, ніж інші, що робить їх кращими під час обчислення рекомендації.

Сформуємо мету програмної реалізації.

3.2 Мета програмної реалізації

Метою програмної реалізації є реалізація обох типів колаборативної фільтрації: базованої на користувачах та базованої на предметах. При реалізації рекомендаційної системи будуть використані різні типи функцій подібності: кореляція Пірсона, косинус подібності, Евклідова відстань. Результатом має стати модуль який можна легко інтегрувати у майбутню комплексну рекомендаційну систему.

Також метою програмної реалізації є порівняння метрик MAE та RMSE для кожного алгоритму, а також порівняння точності робити алгоритмів з використанням різних функцій подібності.

Також було реалізовано рекомендаційну систему, яка може надавати персоналізовані рекомендації фільмів користувачам, базуючись на їх вподобаннях.

Оглянемо реалізацію програмного модуля.

3.3 Реалізація програмного модуля

Для реалізація програмного модуля було обрано мову програмування Python. Python – це потужна та швидка скриптова мова програмування, має безліч бібліотек для роботи з математичними виразами, машинним навчанням тощо.

Щоб перевірити роботу алгоритмів, було обрано набір даних MovieLens 100k, який є дуже популярним набором даних у контексті рекомендаційних систем. Він широко використовується в дослідженнях, і базові цифри легко доступні для порівняльних цілей.

Набір даних містить загалом 100 000 оцінок від набору з 943 користувачів, які висловлюють уподобання до 1682 фільмів. Шкала оцінки – це типове ціле число від 1 до 5, 1 – низька оцінка, а 5 – висока. Набір даних представлений у вигляді єдиного списку із 100 тис. записів потрійних значень, розділених комами (формат файлу CSV), що представляють ідентифікатор користувача, ідентифікатор елемента та рейтинг.

Розроблений програмний модуль містить скрипт similarity.py, який реалізує функції подібності. У цьому модулі реалізується функції косинунусу подібності, кореляція Пірсона, Евклідова відстань.

У скрипті collaborative_filtering.py наведено реалізації методів колаборативної фільтрації на основі користувачів та на основі предметів інтересу.

Цей скрипт будує модель на основі провадженого набору даних, далі базуючись на методі N-сусідів та типі даних (унарні, бінарні або числові) вираховує прогнозований рейтинг. Використовуючи найближчого сусіда, щоб рекомендувати фільми для користувача U, система шукає інших користувачів, які мають подібну історію транзакцій до користувача U. Отримавши список користувачів, схожих на користувача U, система шукає фільми, придбані користувачами, схожими на користувач U і рекомендує товари, які не були придбані користувачем U,

відсортовані за критеріями найбільшого перегляду. Після знаходження N-найближчих сусідів, які мають історію оцінок фільму, рекомендація робиться з використанням їх вирахуваної схожості.

4 ОПИС ЕКСПЕРИМЕНТАЛЬНИХ ДОСЛІДЖЕНЬ

4.1 Планування експерименту

У ході експерименту замірялися такі метрики як MAE та RMSE для кожного типу алгоритму колаборативної фільтрації. Також, була порівняна точність для кожного типу побудови функцій подібності: кореляція Пірсона, косинусова подібніть та Евклідова відстань.

Оглянемо результати експерименту.

4.2 Результати експерименту

Точність для кожного типу побудови функцій подібності (див. рис. 4.1) була розрахована за допомогою стратегії перехресної перевірки без виключення (вилучення вибірки з набору даних та оцінювання її рейтингу на основі решти вибірок).

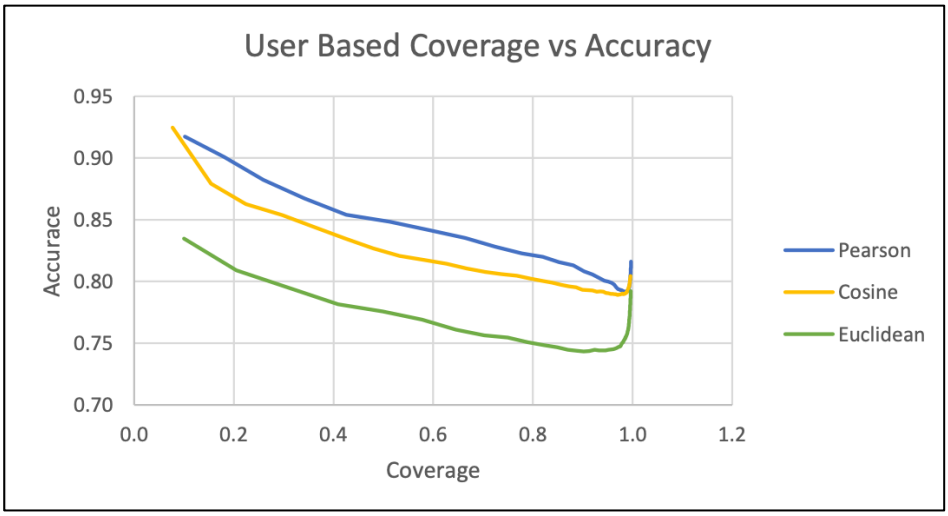

Рисунок 4.1 – Покриття на основі користувачів і прогрес точності

Глобальна точність була розрахована з використанням отриманої індивідуальної точності. Покриття обчислювалося шляхом усереднення кількості обчислених оцінок. Значення MAE та RMSE були отримані як міра точності кожного алгоритму.

Нижче наведено графіки результатів, де показуються результати на основі користувачів у вигляді трикутників, результати на основі елементів у вигляді кола

і результат базової лінії у вигляді квадрата для всіх досліджених функцій подібності. Хороші результати, як правило, розміщуються в нижньому правому куті графіка.

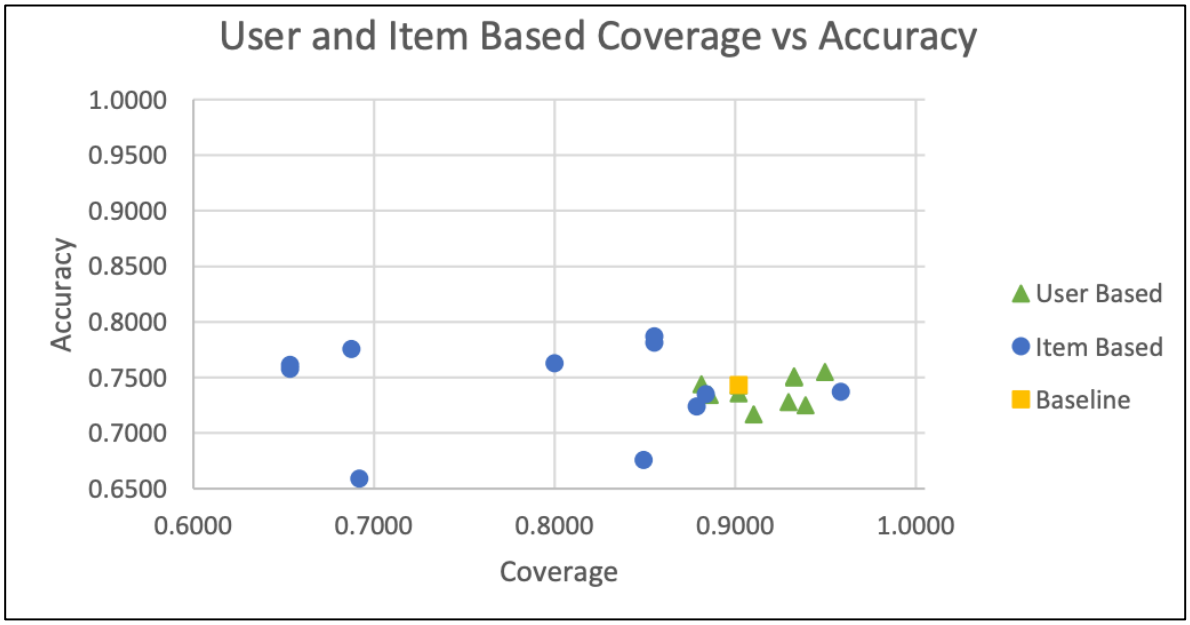

Рисунок 4.2 – Подібність евклідової відстані покриття і прогрес точності

Зосередивши нашу увагу на евклідовій відстані яка дала найкращі загальні результати, ми бачимо, що більшість значень на основі користувачів знаходяться поблизу нашої базової лінії. Чим далі вони знаходяться в нижньому правому квадранті графіка, тим вони кращі, оскільки це означає, що точність нижча, а охоплення вище.

Значення на основі елементів, як правило, менш кластеризовані, і хоча кілька прикладів показують досить високу точність, вони роблять це з низьким покриттям. Єдиним винятком є алгоритм голосування користувачів за замовчуванням на основі елементів, який покращив точність нашої базової лінії, а також покращив охоплення. Ми бачимо цей зразок на рис. 4.2 у вигляді крайнього правого «синього» кола.

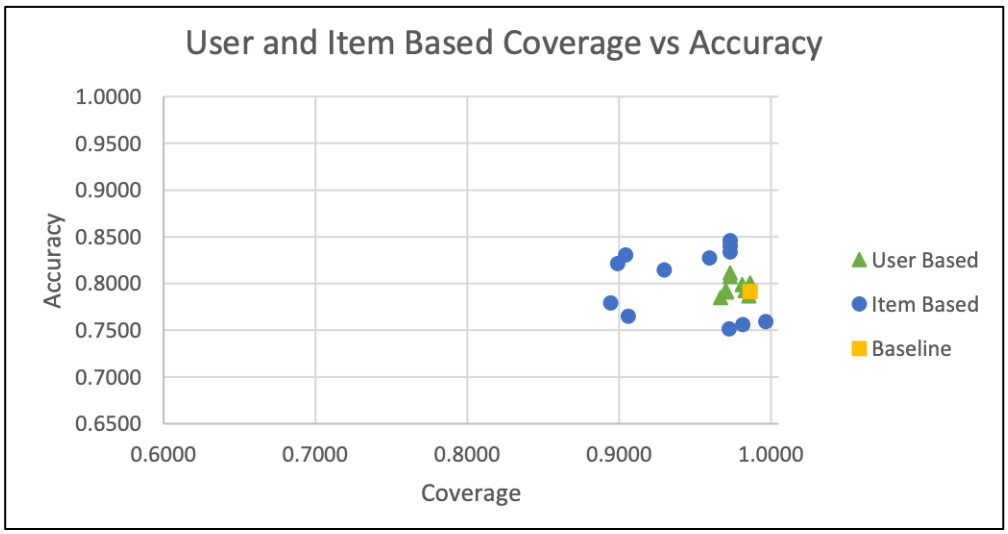

Рисунок 4.3 – Кореляція Пірсона покриття і прогрес точності

На рис. 4.3 показані результати роботи алгоритмів при використанні функції подібності кореляції Пірсона. Цікаво, що всі результати на основі користувача та елемента з цією функцією подібності, як правило, групуються на основі базової лінії, при цьому оцінки на основі користувачів набагато ближче до базової лінії, а оцінка на основі елемента — далі.

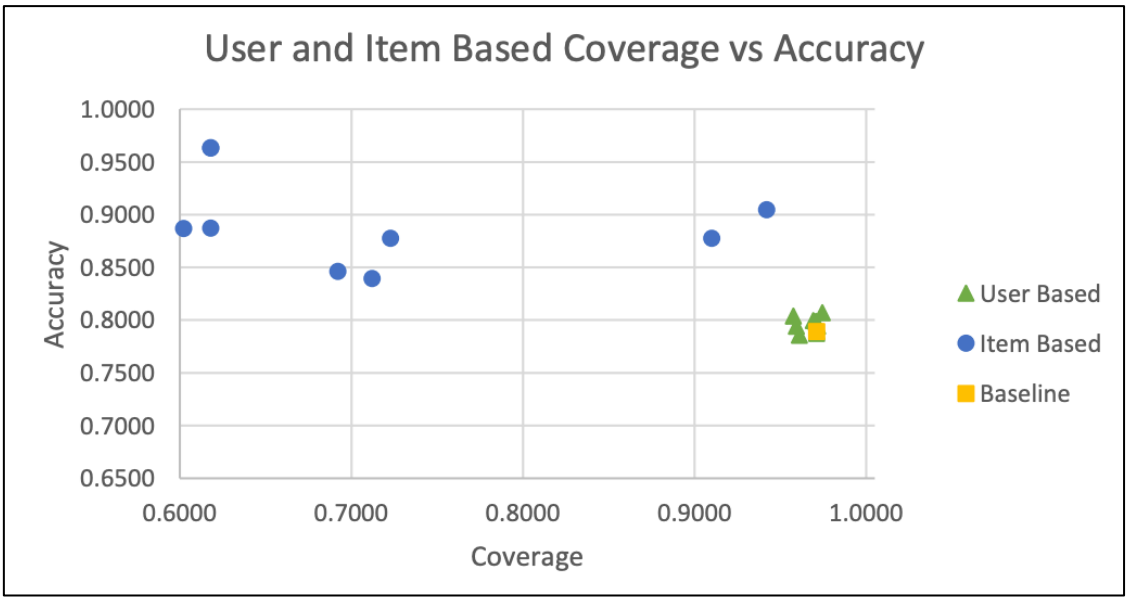

Рисунок 4.4 – Косинус подібності покриття і прогрес точності

У випадку косинусної відстані, зображеної на рис. 4.4, лише результати на основі користувача знаходяться поблизу базової лінії, при цьому всі результати на основі елемента є занадто низькою якістю, щоб їх розглядати далі.

Загалом ми бачимо, що подібність евклідової відстані була кращою, ніж кореляція Пірсона та косинусна відстань, і що пропозиція непрямої оцінки на основі користувача покращила результати точності звичайної стратегії без зменшення покриття, але значного збільшення часу обробки.

Для розробленого алгоритму рекомендації кінофільмів результатом були точні рекомендації за жанрами кінофільму.

```
rec.recommend_item_based(2018)
>> Тому що вам сподобвася мультфільм Бебмі (1942), ми можемо вам
порекомендувати:
```

	movie_id	title	genres
0	364	Lion King, The (1994)	Animation|Children's|Musical
1	588	Aladdin (1992)	Animation|Children's|Comedy|Musical
2	594	Snow White and the Seven Dwarfs (1937)	Animation|Children's|Musical
3	595	Beauty and the Beast (1991)	Animation|Children's|Musical
4	596	Pinocchio (1940)	Animation|Children's
5	1022	Cinderella (1950)	Animation|Children's|Musical
6	1025	Sword in the Stone, The (1963)	Animation|Children's
7	1029	Dumbo (1941)	Animation|Children's|Musical
8	1032	Alice in Wonderland (1951)	Animation|Children's|Musical
9	1033	Fox and the Hound, The (1981)	Animation|Children's

Рисунок 4.5 – Результати розробленої рекомендаційної системи

На рис 4.5 зображено роботу рекомендаційної системи. Для мультфільму «Бембі» алгоритм надав рекомендації до перегляду дитячих мультфільмів з тваринами. Таким чином можна зробити висновок, що реалізовані моделі колаборативної фільтрації, дають гарні рекомендації.

ВИСНОВКИ

В ході виконання атестаційної роботи були досліджені основні аспекти типів колаборативної фільтрації для побудови рекомендаційних систем.

Було проведено аналіз специфіки побудови рекомендаційних систем за допомогою основаної на користувачі та основаної на предметі колаборативної фільтрації. Розглянуті їх переваги та недоліки, особливості використання та реалізації.

Були спроектовані та реалізовані програмні методи щодо реалізації вищезазначених методів колаборативної фільтрації. Під час реалізації було використано мову програмування Python та різноманітні бібліотеки для роботи з математичними виразами.

Був спланований та реалізований експеримент з імплементацією різних типів колаборативної фільтрації. Було порівняно як різні функції подібності впливають на результат виконання рекомендаційної системи.

Результати експериментального дослідження дозволили сформулювати рекомендації щодо побудови рекомендаційних систем та методів колаборативної фільтрації.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Netflix – Википедия. URL: https://ru.wikipedia.org/wiki/Netflix (дата звернення: 06.04.2022).

2. Голян В. В., Бітюкова Є. І. Дослідження технологій та інструментів управління проектами // Монографія: SCIENCE, RESEARCH, DEVELOPMENT#27/ NECHNICS AND TECHNOLOGY – 2020.– С. 33.

3. Н.В. Голян, В.В. Голян, Л.Д. Самофалов. Алгебра понятий как формальный аппарат моделирования действий интеллекта над понятиями // Системи управління, навігації та зв'язку. 2016.– С. 38 – 41.

4. Golyan V., Golyan N. EFFECTIVE METHODS OF INTELLIGENT ANALYSIS IN BUSINESS PROCESSES// 2019 IEEE . International Conference/ April. 2019.

5. How Spotify's Algorithm Works? A Complete Guide to Spotify Recommendation System [2022] | Music Tomorrow Blog. URL: https://www.music-tomorrow.com/blog/how-spotify-recommendation-system-works-a-complete-guide-2022 (дата звернення: 06.04.2022).

6. Understanding TF-IDF for Machine Learning. URL: https://www.capitalone.com/tech/machine-learning/understanding-tf-idf (дата звернення: 14.05.2022).

7. Zhao Y., Zhang B. Research on Recommendation Algorithms Based on Collaborative Filtering. Journal of Physics: Conference Series. 2019. Т. 1237. С. 022094. URL: https://doi.org/10.1088/1742-6596/1237/2/022094 (дата звернення: 13.04.2022).

8. Rafter R. Evaluation and Conversation in Collaborative Filtering. College of Engineering Mathematical and Physical Sciences. 2010. С. 15.

9. Resnick P., Iacovou N., Suchak M. GroupLens: an open architecture for collaborative filtering of netnews. Masloriy Kikabidze – College of Engineering Mathematical and Physical Sciences. 1994. С. 219.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ ЗА НАУКОВИМИ НАПРЯМАМИ КЕРІВНИКА ТА НАУКОВЦІВ КАФЕДРИ ПРОГРАМНОЇ ІНЖЕНЕРІЇ

2. Голян В. В., Бітюкова Є. І. Дослідження технологій та інструментів управління проектами // Монографія: SCIENCE, RESEARCH, DEVELOPMENT#27/ NECHNICS AND TECHNOLOGY – 2020.– С. 33.

3. Н.В. Голян, В.В. Голян, Л.Д. Самофалов. Алгебра понятий как формальный аппарат моделирования действий интеллекта над понятиями // Системи управління, навігації та зв’язку. 2016.– С. 38 – 41.

4. Golyan V., Golyan N. EFFECTIVE METHODS OF INTELLIGENT ANALYSIS IN BUSINESS PROCESSES// 2019 IEEE . International Conference/ April. 2019.