Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____Комп'ютерних наук_____
(повна назва)

Кафедра _____Інформаційних управляючих систем_____
(повна назва)

АТЕСТАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти _____другий (магістерський)_____

_____Дослідження методів формування онлайн-рекомендацій в системах електронної комерції_____
(тема)

Виконав:
студент __2__ курсу, групи ___ІУСТм-19-1__
_____Кубота Данило Дмитрович_____
(прізвище, ініціали)

Спеціальність __122 Комп'ютерні науки_____
_____
(код і повна назва спеціальності)

Тип програми __освітньо-професійна_____
(освітньо-професійна або освітньо-наукова)

Освітня програма____Інформаційні управляючі системи та технології_____
(повна назва освітньої програми)

Керівник _____проф. каф. ІУС Чала О.В.___
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ ___Петров К.Е._____
(підпис) (прізвище, ініціали)

2020р.

Харківський національний університет радіоелектроніки

Факультет_____Комп'ютерних наук_____

(повна назва)

Кафедра____Інформаційних управляючих систем_____

(повна назва)

Рівень вищої освіти_____другий (магістерський)_____

Спеціальність _____122 Комп'ютерні науки_____
(код і повна назва)
Тип програми __освітньо-професійна_____
(освітньо-професійна або освітньо-наукова)
Освітня програма__Інформаційні управляючі системи та технології_____
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«____»_____ 20 ___ р.

ЗАВДАННЯ
НА АТЕСТАЦІЙНУ РОБОТУ

студентові_____Куботі Данилу Дмитровичу_____

(прізвище, ім'я, по батькові)

1.Тема роботи __ Дослідження методів формування онлайн-рекомендацій в системах_____ електронної комерції _____
затверджена наказом по університету від __27_ ___жовтня_____2020 р. № ____1455 Ст__
2. Термін подання студентом роботи до екзаменаційної комісії __17__ _грудня____ 2020 р.
3. Вихідні дані до роботи __Науково-технічні публікації та інтернет джерела з тематики____ атестаційної роботи_____
4. Перелік питань, що потрібно опрацювати в роботі вступ, дослідження методів формування онлайн-рекомендацій, структуризація рекомендаційної підсистеми в іс електронної комерції, дослідження підходів формування рекомендацій в системах електронної комерції, дослідження методів побудови рекомендацій в онлайн- режимі, дослідження методів формування рекомендацій на основі неявного зворотньому зв'язку, постановка задачі дослідження, дослідження методів колаборативної та контентної фільтрації, особливості методів колаборативної фільтрації, особливості методів контентної фільтрації, удосконалення методів формування онлайн-рекомендацій з урахуванням негативних неявних відгуків та заповненості профілю користувача, дослідження отриманих наукових результатів, порівняння методів формування онлайн-рекомендацій, опис технології побудови онлайн-рекомендацій з використанням удосконаленого методу, практичне використання досліджених результатів, висновки.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Аналіз літератури та Інтернет-джерел	2.11.20-4.11.20	
2	Постановка задачі	5.11.20-6.11.20	
3	Обробка матеріалу	8.11.20-10.11.20	
4	Дослідження сучасного стану вирішення проблеми	11.11.20-13.11.20	
5	Дослідження моделей та методів форм. рекомендацій.	14.11.20-20.11.20	
6	Дослідження особливостей формування онлайн-рекоменд.	21.11.20-25.11.20	
7	Експериментальна перевірка результатів дослідження	26.11.20-30.11.20	
8	Написання пояснювальної записки	1.11.20-5.11.20	
9	Підготовка презентації	6.12.20	
10	Надання роботи для перевірки на плагіат	7.12.20	
11	Надання роботи на підпис науковому керівникові	10.12.20	
12	Попередній захист	11.12.20	
13	Надання роботи на рецензію	12.12.20	
14	Надання роботи на підпис завідувачу кафедрою	14.12.20	
15	Надання підписаної завідувачем кафедрою роботи в ДЕК	16.12.20	
16	Захист	17.12.20	

Дата видачі завдання __2___ _____листопада_____2020 р.

Студент _____
(підпис)

Керівник роботи _____ _____проф. Чала О.В._____
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка до магістерської атестаційної роботи містить: 85 с., 4 розділи, 19 рис., 2 табл., 42 джерела.

ЕЛЕКТРОННА КОМЕРЦІЯ, ІНТЕРНЕТ-МАГАЗИН, КОЛАБОРАТИВНА ФІЛЬТРАЦІЯ, КОНТЕНТНА ФІЛЬТРАЦІЯ, НЕЯВНИЙ ЗВОРОТНИЙ ЗВ’ЯЗОК, РЕКОМЕНДАЦІЙНІ СИСТЕМИ, СИСТЕМИ ЕЛЕКТРОННОЇ КОМЕРЦІЇ

Магістерська робота містить у собі дослідження проблем підвищення точності рекомендацій. У роботі було виконано огляд методід формування онлайн-рекомендацій у системах електронної комерції.

Проведено аналіз існуючих методів формування онлайн-рекомендацій, а саме: колаборативної фільтрації та методу формування рекомендацій на основі вмісту. Для колаборитивної фільтрації було розглянуто проблему використання негативних неявних відгуків, а також заміщення її контентною фільтрацією у разі, коли поведінковий профіль користувача заповнений недостатньо для точного формування рекомендацій шляхом колаборативної фільтрації. Такий підхід дозволиляє збільшити точність формування рекомендацій для користувачів, профілем заповненим достатньо повно, а також знизити час формування рекомендацій, коли даних профілю недостатньо. На практиці метод дозволяє покращити числове значення показника точності приблизно на 4 %, а у випадку коли профіль користувача заповнений недостатньо повно надавати рекомендації за менший час використовуючи контентну фільтрацію.

Результати дослідження по магістерській роботі представлено на міжнародному молодіжного форуму «Радіоелектроніка та молодь у XXI столітті» 2020р. опубліковані тези доповіді «Дослідження методів формування онлайн-рекомендацій у системах електронної комерції».

ABSTRACT

Explanatory Note to master certification work contains 85 pages, 4 sections, 19 pictures, 2 tables, 42 sources.

ELECTRONIC COMMERCE, ELECTRONIC COMMERCE SYSTEMS, ONLINE STORE, COLLABORATIVE FILTRATION, CONTENT CONTENT, RECOMMENDATION SYSTEMS

The master's thesis includes research on the problems of improving the accuracy of recommendations.

The paper reviews the methods of forming online recommendations in e-commerce systems.

An analysis of existing methods of forming online recommendations, namely: collaborative filtering and the method of forming recommendations based on content. For collaborative filtering, the problem of using negative implicit feedback was considered, as well as replacing CF with content-based filtering in the case when the user's behavioral profile is not filled enough to accurately form recommendations by collaborative filtering. This approach increases the accuracy of recommendations for users whose profile is full enough to use collaborative filtering, and reduces the time for recommendations when profile data is insufficient.

In practical terms, the method allows to improve the numerical value of the accuracy indicator by a little more than 4%, and in the case when the user profile is filled not enough to provide recommendations in less time using content-based filtering.

The results of the research on the master's thesis were presented at the international youth forum "Radio Electronics and Youth in the XXI Century" in 2020. published abstracts of the report "Research of methods of forming online recommendations in e-commerce systems".

ПЕРЕЛІК СКОРОЧЕНЬ, УСЛОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ ТА ТЕРМІНІВ

ІП – інформаційне перевантаження;

ІС – інформаційна система;

КФ – колаборативна фільтрація;

AUC – Area under ROC curve;

CMS – Content Management System;

JSON – JavaScript Object Notation;

ROC – Receiver operating characteristic;

ЗМІСТ

ВСТУП

Системи електронної комерції пропонують персональний перелік товарів або послуг для споживача, що і забезпечило їх швидкий розвиток в останнє десятиріччя. Для побудови рекомендованого списку об’єктів в системах e-commerce використовується рекомендаційна підсистема. Дана підсистема формує рекомендації в офлайн- та онлайн-режимах. В першому режимі при побудові персонального переліку товарів або послуг використовується вся наявна інформація про минулі покупки споживача та про характеристики

цікавих для нього об’єктів. Такі рекомендації готуються заздалегідь, до входу користувача в систему електронної комерції. В онлайн-режимі виконується уточнення офлайн-рекомендацій. Онлайн-режим ставить підвищені вимоги до швидкості побудови рекомендацій з урахуванням особливостей поточної поведінки споживача. Проведений аналіз [1] методів побудови рекомендацій показав, що існуючі підходи розділяються на дві групи:

– на основі схожості товарів або послуг; предмети до рекомендації підбираються на основі подібності їх властивостей, які становлять цінність для споживача;

– на основі схожості користувачів; подібність вимог споживачів розраховується виходячи із історії покупок та рейтингів їх та схожих користувачів рекомендаційної системи.

При формуванні рекомендацій в онлайн-режимі необхідно комбінувати ці методи, в залежності від характеристик поточного споживача. Наприклад, для нового користувача необхідно буде показувати ті товари, які користуються популярністю. Для відомого споживача необхідно враховувати останні дані про перегляд сторінок сайту електронної комерції і, на цій основі, уточнити попередні рекомендації.

Для того щоб вирішити проблеми, що були наведені вище, необхідно використовувати рекомендаційні системи. Рекомендаційні системи - це інформаційні системи, що надають пропозиції щодо предметів, які будуть корисними для користувача. Пропозиції стосуються різних процесів прийняття рішень, наприклад, які товари купувати, яку музику слухати чи які новини в Інтернеті читати. Прикладом цього є система рекомендування книг, яка допомагає користувачам вибрати книгу для читання.

Оскільки рекомендації зазвичай персоналізовані, різні користувачі або групи користувачів отримують різноманітні пропозиції. Крім того, є також неперсоналізовані рекомендації. Їх набагато простіше створити, і вони зазвичай розміщуються в журналах чи газетах. Типовими прикладами є десятка найкращих книжок, компакт-дисків тощо. Хоча вони можуть бути корисними

та ефективними у певних ситуаціях, ці типи неперсоналізованих рекомендацій зазвичай не враховуються дослідженнями РС.

У цій роботі ми зосередимось на двох основних стратегіях методах формування онлайн-рекомендацій:

– контентна фільтрація;

– колаборативна фільтрація.

Контентний підхід створює профіль для кожного користувача чи продукту для характеристики його природи. Як приклад, профіль фільму може містити атрибути, що стосуються його жанру[18], акторів-учасників, популярності касових зборів тощо. Профілі користувачів можуть містити демографічну інформацію або відповіді на відповідну анкету. Отримані профілі дозволяють програмам асоціювати користувачів із відповідними продуктами. Однак стратегії, що базуються на вмісті, вимагають збору зовнішньої інформації, яка може бути недоступною або простою для збору.

Колаборативна фільтрація (КФ) - термін, придуманий розробниками першої системи рекомендацій. КФ аналізує взаємозв'язки між користувачами та взаємозалежності між продуктами, щоб виявити нові асоціації предметів та користувача. Наприклад, деякі системи КФ ідентифікують пари предметів, які, як правило, оцінюються однаково або однодумці користувачів зі схожою історією рейтингу чи покупок для виведення невідомих взаємозв’язків між користувачами та предметами. Єдиною необхідною інформацією є минула поведінка користувачів, яка може бути їх попередніми транзакціями або способом оцінки товарів. Незважаючи на те, що КФ, як правило, є більш точним, ніж методи, що базуються на вмісті, КФ страждає від проблеми холодного старту через його нездатність звертатися до нових продуктів системи, для яких підходи на основі вмісту були б достатніми. Рекомендовані системи покладаються на різні типи вхідних даних.

Однак ці методи формування рекомендацій зазвичай враховують окремо дані споживача або характеристики товарів, історію активності тощо. Наприклад, колаборативна фільтрація страждає від проблеми «холодного

старту» для користувачів, та у випадку з неявним зворотнім зв’язком, не використовує негативний відгук. Тоді як контентна фільтрація не бере до уваги вподобання користувачів.

Для вирішеня цієї проблеми необхідно удосконалити метод колаборативної фільтрації аби брати до уваги негативний зворотній зв’язок, наприклад повернення товарів назад до магазину, а також комбінувати колаборативну фільтрацію з контентною, щоб мати можливість рекомендувати користувачу товари у випадках, коли його вподобання невідомі.

Отриманим результатом роботи є удосконалений метод, який підвищує точність надання рекомендацій, частково вирішує проблему «холодного старту» дял користувачів, використовуючи контентну фільтрацію.

1 ДОСЛІДЖЕННЯ МЕТОДІВ ФОРМУВАННЯ ОНЛАЙН-РЕКОМЕНДАЦІЙ

1.1 Структуризація рекомендаційної підсистеми в ІС електронної комерції

Електронна комерція є такою формою торгівлі, при якій вибір і замовлення товарів здійснюється через комп'ютерні мережі, а розрахунки між покупцем і постачальником здійснюються з використанням електронних документів та / або коштів платежу[14].

Узагальнену структуру системи електронної комерції, що реалізована у вигляді інтернет-магазину, наведено на рисунку 1.1.

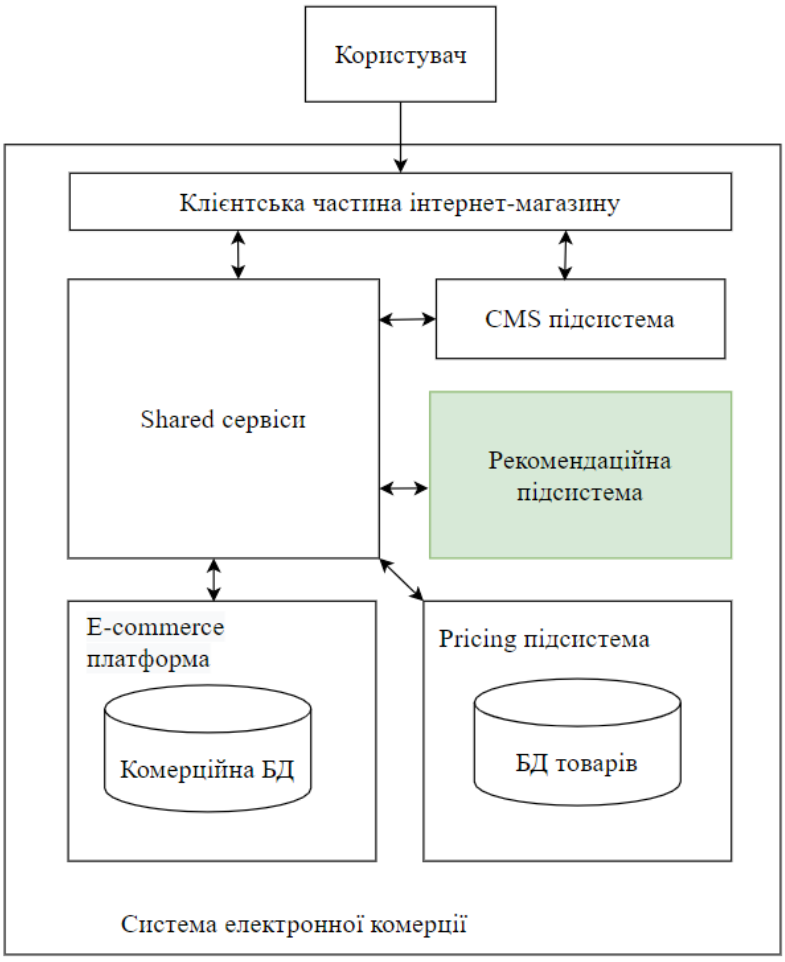

Рисунок 1.1 – Структура ІС електронної комерції

Система електронної комерції розподіляється на наступні підсистеми: Pricing підсистема, E-commerce платформа, Shared сервіси, CMS підсистема, Клієнтська частина інтернет-магазину та Рекомендаційна підсистема.

За такої реалізації клієнт системи електронної комерції напряму взаємодіє з клієнтською частиною інтернет-магазину[16], яка є, по суті, фасадом для усієї системи, і повертає користувачу дані про товари та зовнішній вигляд інтернет магазину. Ці дані вона отримує від відповідно Shared сервісів та CMS системи.

Shared сервіси – це сукупність мікросервісів, які валідують вхідні дані від клієнтів, отримують інформацію з інших підсистем, обробляють ці дані та повертають клієнту. Зазвичай ці дані інформації про клієнтів, товари і т.д.

CMS підсистема - це комп’ютерне програмне забезпечення, що використовується для управління створенням та модифікацією цифрового вмісту, такого як зовнішний вигляд сторінок сайту, його структуру і так далі.

E-commerce платформа – це підсистема, що дозволяє в єдиному інформаційному просторі управляти електронною комерцією: онлайн- і офлайн-точками взаємодії з клієнтами - сайтами, колл-центрами, соціальними мережами та друкованими матеріалами.

Pricing підсистема відповідає за зберігання, обробку, та конфігурацію даних про продукти системи електронної комерції, наприклад цін, назв, кількості товарів у сховищах і т.д.

Дані, необхідні для працездатності рекомендаційні системи отримуються через Shared сервіси з E-commerce платформи та Pricing потім зберігаються їх для подальшої обробки в автономному режимі. Однак обчислення можуть виконуватися в офлайн режимі або в режимі онлайн. Онлайн-обчислення можуть краще реагувати на останні події та взаємодію користувачів, але вони повинні реагувати на запити в режимі реального часу. Це може обмежити обчислювальну складність використовуваних алгоритмів, а також обсяг даних, які можна обробити.

Офлайн-обчислення мають менше обмежень на обсяг даних та обчислювальну складність алгоритмів, оскільки вони виконуються пакетним

способом із послабленими вимогами до часу[17]. Однак між оновленнями рекомендації може легко застаріти, оскільки найновіші дані не включені. Одне з ключових питань архітектури персоналізації полягає в тому, як поєднувати та керувати обчисленнями в онлайн та офлайн режимах безперешкодно.

Навчання моделі - це ще одна форма обчислень, яка використовує наявні дані для створення моделі, яка згодом буде використана під час фактичного обчислення результатів. На рисунку 1.2 продемонстрована структура онлайн та офлайн частин рекомендаційної підсистеми.

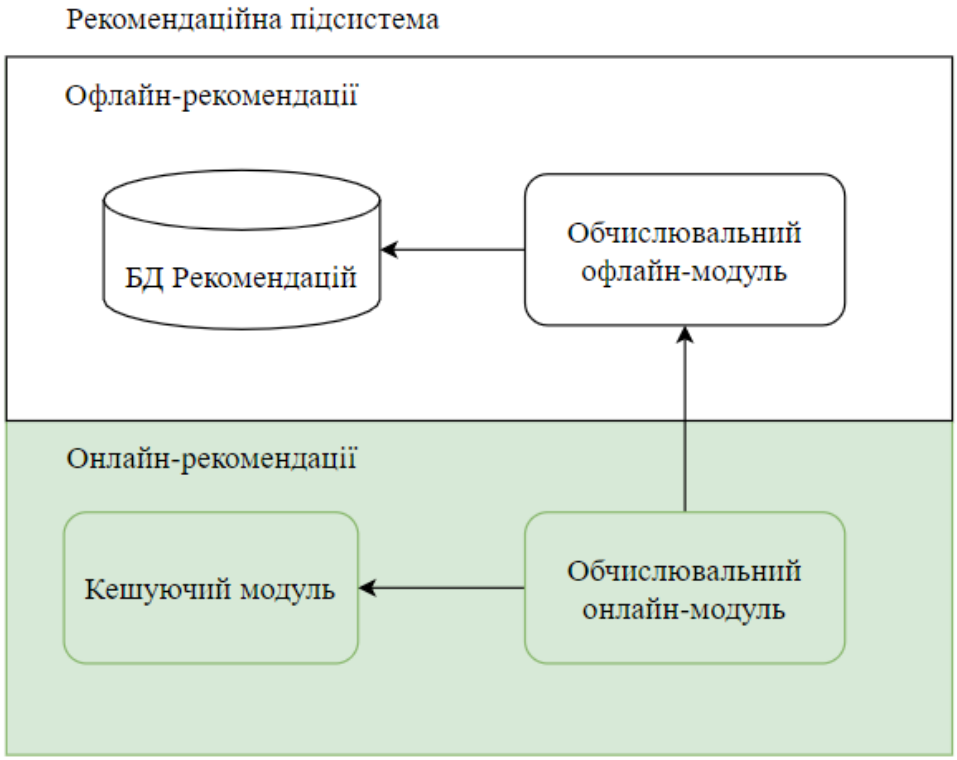

Рисунок 1.2 - Структура онлайн та офлайн частин рекомендаційної підсистеми.

1.2 Дослідження підходів формування рекомендацій в системах електронної комерції

Можна виділити шість різних класів рекомендаційних підходів:

– контентна фільтрація;

– колаборативна фільтрація;

– демографічна фільтрація;

– спільнотна фільтрація;

– РС засновані на знаннях;

– гібридні методи рекомендацій.

Контентна фільтрація. За цієї реалізації система вчиться рекомендувати предмети, схожі на ті, що сподобались користувачеві в минулому або, з якими він взаємодіє зараз. Подібність предметів обчислюється на основі ознак, пов’язаних із порівняними предметами. Наприклад, якщо користувач позитивно оцінив фільм, що належить до комедійного жанру, система може навчитися рекомендувати інші фільми цього жанру[10].

Колаборативна фільтрація. Найпростіша та оригінальна реалізація цього підходу рекомендує активному користувачеві предмети, які сподобались іншим користувачам зі схожими вподобаннями в минулому. Схожість вподобань двох користувачів обчислюється на основі подібності в історії рейтингів користувачів. Колаборативна фільтрація вважається найпопулярнішою та широко застосовуваною технікою в РС. Методи колаборативної фільтрації зосереджені на взаємозв'язку між предметами або, як альтернатива, між користувачами. Предметний підхід моделює перевагу користувача перед предметом на основі оцінок подібних предметів тим самим користувачем. Методи колаборативної фільтрації користуються значною здатністю давати точні та персоналізовані рекомендації. Ці методи перетворюють як предмети, так і користувачів в один і той же простір прихованих факторів. Потім прихований простір використовується для пояснення рейтингів, описуючи як товари, так і користувачів факторами, автоматично виведеними з відгуків користувачів. Також описують методи для усунення недоліків сусідських методів, пропонуючи більш суворі формулювання із використанням глобальних методів оптимізації. Використання таких методів дає можливість зняти обмеження розміру району та врахувати неявний зворотний зв'язок та часову динаміку[11]. Отримана точність наближається до моделей факторизації матриць, одночасно пропонуючи ряд практичних переваг.

РС засновані на демографічній фільтрації. Цей тип системи рекомендує предмети на основі демографічного профілю користувача. Припускають, що для різних груп людей, в залежності від їх розташування, слід формувати різні рекомендації. Багато веб-сайтів застосовують прості та ефективні персоналізаційні рішення на основі демографічних показників. Багато веб-сайтів застосовують прості та ефективні персоналізаційні рішення на основі демографічних показників. Наприклад, користувача перенаправляють на певні веб-сайти залежно від його мови або країни. Або пропозиції можуть бути налаштовані відповідно до віку користувача. Незважаючи на те, що ці підходи були досить популярними в маркетинговій літературі, було проведено порівняно мало належних досліджень РС щодо демографічних систем.

Профіль - це структуроване представлення інтересів користувачів, прийняте для рекомендації нових цікавих предметів. Процес рекомендації в основному полягає у зіставленні атрибутів профілю користувача з атрибутами об’єкта вмісту. Результат - оцінка відповідності, яка відображає рівень зацікавленості користувача в цьому об’єкті.

Якщо профіль точно відображає уподобання користувачів, це має надзвичайну перевагу для ефективності процесу доступу до інформації. Наприклад, його можна використовувати для фільтрування результатів пошуку, вирішуючи, зацікавлений користувач певною веб-сторінкою чи ні, а в негативному випадку запобігає її відображенню.

РС засновані на знаннях. Системи, що базуються на знаннях, рекомендують предмети на основі конкретних знань домену про те, як певні характеристики предметів відповідають потребам та уподобанням користувачів та про те, наскільки предмет корисний для користувача. Серед систем, рекомендацій, що базуються на знаннях, варто виокремити ті, що базуються на конкретних випадках. У цих системах функція подібності оцінює, наскільки потреби користувача відповідають рекомендаціям. Тут оцінку подібності можна прямо трактувати як корисність рекомендації для користувача[12].

Системи, засновані на обмеженнях, є іншим типом РС, заснованих на знаннях. З точки зору використаних знань, обидві системи схожі: збираються вимоги користувачів; та пояснюються результати рекомендацій. Основна різниця полягає в способі обчислення рішень. Рекомендатори, що базуються на конкретних випадках, визначають рекомендації на основі метрик подібності, тоді як рекомендатори, що базуються на обмеженнях, переважно використовують заздалегідь визначені бази знань, що містять чіткі правила щодо того, як співвідносити вимоги замовника з особливостями товару.

Спільнотна фільтрація. Цей тип системи рекомендує товари, виходячи з уподобань саме друзів або знайомих користувачів. Ця техніка слідує епіграмі «Скажи мені, хто твій друг, і я скажу тобі, хто ти». Докази свідчать, що люди, як правило, покладаються більше на рекомендації своїх друзів, ніж на рекомендації подібних, але анонімних осіб. Це спостереження, у поєднанні зі зростаючою популярністю відкритих соціальних мереж, породжує зростаючий інтерес до систем, що базуються на громадах, або, як зазвичай називають, соціальних рекомендаторів системи. Цей тип РС моделює та отримує інформацію про соціальні відносини користувачів та уподобання друзів користувача. Рекомендація базується на рейтингах, наданих друзями користувача. Насправді ці РС стежать за зростанням соціальних мереж і дають можливість простого та всебічного отримання даних[13], що стосуються соціальних відносин користувачів. Дослідження в цій галузі все ще перебувають на початковій фазі, і результати щодо продуктивності систем неоднозначні. Наприклад, загальні рекомендації, засновані на соціальних мережах, не є більш точними, ніж ті, що походять від традиційних підходів до КФ, за винятком особливих випадків, наприклад, коли рейтинги користувачів певного предмета дуже різноманітні (тобто суперечливі предмети) або для ситуацій холодного старту, тобто де користувачі не надали достатньо оцінок для обчислення схожості з іншими користувачами. Інші показали, що в деяких випадках дані соціальних мереж дають кращі рекомендації ніж дані про

подібність профілю, а додавання даних соціальних мереж до традиційних МВ покращує результати рекомендацій.

Гібридні методи рекомендацій. Ці РС комбінують у собі два або більше з вищезазначених методів. Гібридна система, що поєднує методи А і В, намагається використовувати переваги А, щоб виправити недоліки В. Наприклад, методи КФ страждають від проблем з новими предметами, тобто вони не можуть рекомендувати предмети, які не мають рейтинги. Це не обмежує підходи, орієнтовані на вміст, оскільки прогнозування нових предметів базується на їх описі (особливостях), які, як правило, легко доступні. Враховуючи два (або більше) базових методів RS, було запропоновано кілька способів поєднання їх створити нову гібридну систему.

1.3 Дослідження методів побудови рекомендацій в онлайн- режимі

Рекомендаційні системи (РС) - це інформаційні системи, що надають пропозиції щодо предметів, які будуть корисними для користувача.

Пропозиції стосуються різних процесів прийняття рішень, наприклад, які товари купувати, яку музику слухати чи які новини в Інтернеті читати[2].

«Предмет» - загальний термін, що використовується для позначення того, що система рекомендує користувачам. РС зазвичай фокусується на конкретному типі предметів (наприклад, на компакт-дисках чи новинах) та на його дизайні, графічному користувальницькому інтерфейсі системи електронної комерції та на основній методиці рекомендацій, що використовуються для формування рекомендацій щодо цього типу предмета.

РС спрямовані в першу чергу на осіб, яким бракує достатнього особистого досвіду чи компетенції для оцінки потенційно переважної кількості альтернативних предметів, які може запропонувати, наприклад, веб-сайт. Прикладом цього є рекомендаційна система книг, що допомагає користувачам обрати книгу для читання[3]. На популярному веб-сайті Amazon.com веб-сайт використовує RS для персоналізації Інтернет-магазину для кожного клієнта.

Оскільки рекомендації зазвичай персоналізовані, різні групи користувачів або конкретні користувачі отримують різноманітні пропозиції. Більш того, також можна виділити неперсоналізовані рекомендації. Останні набагато простіше створити, і вони зазвичай розміщуються у газетах чи журналах. Типовими прикладами є десятка найкращих книжок, компакт-дисків тощо. Хоча вони можуть бути корисними та ефективними у певних ситуаціях, ці типи неперсоналізованих рекомендацій зазвичай не враховуються дослідженнями РС.

У найпростішій формі персоналізовані рекомендації пропонуються як рейтингові списки предметів. Виконуючи цей рейтинг, РС намагаються передбачити, які продукти або послуги є найбільш підходящими, виходячи із уподобань та обмежень користувача. Для того, щоб виконати таке обчислювальне завдання, RS збирають від користувачів їх уподобання, які або явно виражаються, наприклад, як рейтинги для продуктів, або виводяться шляхом інтерпретації дій користувача[4]. Наприклад, РС може розглядати навігацію до певної сторінки товару[5] як неявну ознаки переваги користувача до предметів, показаних на цій сторінці.

РС активно збирають різні види даних для побудови своїх рекомендацій. Дані стосуються в першу чергу предметів, що пропонуються, та користувачів, які отримають рекомендації для цих предметів. Але оскільки джерела даних, доступних для рекомендаційних систем можуть бути дуже різноманітними, зрештою, чи можна їх використовувати чи ні, залежить від техніки рекомендацій.

Загалом, існують методи рекомендацій, які є обмеженими, тобто вони використовують дуже прості та основні дані, такі як оцінки користувачів для предметів. Інші методи набагато більше залежать від знань, наприклад, використання онтологічних описів користувачів або предметів, або обмежень, або соціальних відносин та діяльності користувачів. У будь-якому випадку, дані, що використовуються РС, поділяються на три види об'єктів: предметів, користувачів та транзакцій, тобто відносин між користувачами та предметами.

Предмети - це ті об'єкти, які рекомендуються. Предмети можуть характеризуватися своєю складністю та цінністю чи корисністю. Значення товару може бути позитивним, якщо товар корисний для користувача, або негативним, якщо товар не підходить і користувач прийняв неправильне рішення, вибравши його.

Користувачі. РС, як зазначалося вище, можуть мати дуже різноманітні цілі та характеристики. Для того, щоб персоналізувати рекомендації та взаємодію людина-комп'ютер[6], РС використовують широкий спектр інформації про користувачів. Ця інформація може бути структурована різними способами, а вибір інформації, яку перетворювати модель для подальшої обробки, залежить від техніки рекомендацій.

Наприклад, під час колаборативної фільтрації користувачі подаються у вигляді простого списку, що містить рейтинги, надані користувачем для деяких предметів. У демографічній РС використовуються соціально-демографічні атрибути, такі як вік, стать, професія та освіта. Дані користувача складають модель користувача. Модель користувача визначає користувача, тобто кодує його уподобання та потреби. Існує багато різних підходів до визначення користувачів за допомогою моделей, і, у певному сенсі[7], РС можна розглядати як інструмент, який формує рекомендації шляхом побудови та використання моделей користувачів. Оскільки жодна персоналізація неможлива без зручної для використання моделі користувача, якщо це звісно не непресоналізована рекомендація, як при виборі десяти найкращих чи найпопулярніших останнім часом предметів, модель користувача завжди відіграватиме центральну роль. Наприклад, розглядаючи метод колаборативної фільтрації, профіль користувачао описується безпосередньо за своїми рейтингами до предметів, або, використовуючи ці рейтинги, система виводить вектор значень коефіцієнтів, де користувачі різняться залежно від того, як кожен коефіцієнт зважує свою модель. Користувачів також можна описати за даними їх поведінки, наприклад, шаблонами перегляду веб-сайтів (у веб-системі рекомендацій) або шаблонами пошуку подорожей (у системі

рекомендацій подорожей). Більше того, дані користувачів можуть включати відносини між користувачами, наприклад рівень довіри цих відносин між користувачами. РС може використовувати цю інформацію, щоб рекомендувати предмети користувачам, які віддають перевагу подібним або довіреним користувачам[8].

Транзакції - це дані, подібні до журналу, які зберігають важливу інформацію, що генерується під час взаємодії людина-комп'ютер, і корисні для алгоритму генерації рекомендацій, який використовує система. Наприклад, журнал транзакцій може містити посилання на предмет, вибраний користувачем, та опис контексту (наприклад, ціль / запит користувача) для цієї конкретної рекомендації. Якщо транзакція доступна, ця транзакція може також містити явний відгук, наданий користувачем, наприклад, рейтинг для обраного товару. Рейтинги - це найпопулярніша форма даних про транзакції, яку збирає РС. Ці рейтинги можуть збиратися явно або неявно. Під час явної збірки оцінок користувачеві пропонується надати свою думку щодо товару за шкалою оцінок. Відповідно рейтинги можуть приймати найрізноманітніші форми:

Числові рейтинги. Прикладом таких рейтингів може стати система з «зірками». Користувачу пропонується оцінити предмет на 1–5 зірок, в залежнності від його вподобання до придбаного або переглянутого предмету.

Звичайні рейтинги, такі як «категорично погоджуюсь, погоджуюсь, нейтральний, не погоджуюсь, категорично не погоджуюсь», коли користувачеві пропонується вибрати термін, який найкраще вказує на її думку щодо того чи іншого предмета (зазвичай за допомогою анкети[9]).

Бінарні рейтинги, за яких користувачеві просто пропонується вирішити, хороший чи поганий певний предмет.

Унарні рейтинги. Такі оцінки можуть свідчити про те, що користувач спостерігав або купував товар або іншим чином позитивно оцінював товар. У таких випадках відсутність рейтингу вказує на те, що ми не маємо інформації, що стосується користувача до предмету (можливо, він придбав товар в іншому

місці). Інша форма оцінки користувача складається з тегів, пов’язаних користувачем з предметами, представленими системою.

1.4 Дослідження методів формування рекомендацій на основі неявного зворотньому зв’язку

Загалом, системи рекомендацій використовують на двох основних стратегіях (або їх комбінаціях):

Контентний підхід створює профіль для кожного користувача чи продукту для характеристики його природи. Як приклад, профіль фільму може містити атрибути, що стосуються його жанру[18], акторів-учасників, популярності касових зборів тощо.

Профілі користувачів можуть містити демографічну інформацію або відповіді на відповідну анкету. Отримані профілі дозволяють програмам асоціювати користувачів із відповідними продуктами. Однак стратегії, що базуються на вмісті, вимагають збору зовнішньої інформації, яка може бути недоступною або простою для збору.

Колаборативна фільтрація (КФ) - термін, придуманий розробниками першої системи рекомендацій. КФ аналізує взаємозв'язки між користувачами та взаємозалежності між продуктами, щоб виявити нові асоціації предметів та користувача.

Наприклад, деякі системи КФ ідентифікують пари предметів, які, як правило, оцінюються однаково або однодумці користувачів зі схожою історією рейтингу чи покупок для виведення невідомих взаємозв’язків між користувачами та предметами. Єдиною необхідною інформацією є минула поведінка користувачів, яка може бути їх попередніми транзакціями або способом оцінки товарів.

Незважаючи на те, що КФ, як правило, є більш точним, ніж методи, що базуються на вмісті, КФ страждає від проблеми холодного старту через його

нездатність звертатися до нових продуктів системи, для яких підходи на основі вмісту були б достатніми. Рекомендовані системи покладаються на різні типи вхідних даних.

Найзручнішим для розуміння вподобань користувача є високоякісний явний відгук, за якогокористувач повинен явно висловити свої прихильності до певного предмету. Наприклад, Netflix збирає рейтинги для фільмів, використовуючи систему «пальців вгору / вниз», що дозволяє зрозуміти, чи вподобав користувач певний фільм або ні. Однак явний відгук не завжди доступний[19].

Таким чином, рекомендуючі можуть зробити висновки щодо переваг користувачів на основі менш прозорих неявних відгуків, які побічно відображають думку через спостереження за поведінкою користувачів. До неявних відгуків включають такі дії як історія покупок, історія перегляду, шаблони пошуку або навіть рухи миші.

Наприклад, користувачеві, який придбав багато книг того самого автора, напевно подобається цей автор.

1.5 Постановка задачі дослідження

Об’єктом дослідження в рамках даної роботи є процес формування рекомендації для користувачів систем електронної комерції.

Предметом дослідження являються методи формування онлайн-рекомендацій в системах електронної комерції.

Метою даної роботи є вивчення методів формування рекомендацій на основі контенту(контентної фільтрації) та колаборативної фільтрації використовуваних у процесі формування онлайн-рекомендацій.

Для досягнення мети, потрібно дослідити:

– особливості рекомендаційної підсистеми в системі електронної комерції;

– методи формування онлайн-рекомендацій в системах електронної комерції;

– методи колаборативної фільтрації у рекомендаційних системах;

– методи контентної фільтрації у рекомендаційних системах;

– удосконалення методу колаборативної фільтрації з моделями прихованих факторів;

– розробка технології використання удосконаленого методу;

– експериментальна перевірка удосконалення та комбінування методів формування онлайн-рекомендацій.

2 ДОСЛІДЖЕННЯ МЕТОДІВ КОЛАБОРАТИВНОЇ ТА КОНТЕНТНОЇ ФІЛЬТРАЦІЇ

2.1 Особливості методів колаборативної фільтрації

Моделі CF намагаються відобразити взаємодію між користувачами та елементами та проаналізувати різницю у рейтингах, наприклад, для одного товару різними користувачами. Однак значна частина спостережуваних рейтингових відхилень зумовлена певними обставинами, пов'язаними або з користувачами, або з предметами, незалежно від їх взаємодії.

Основним прикладом є те, що типові дані КФ демонструють великі упередження певного користувача або упередження користувачів загалом щодо певного предмету - тобто систематичні тенденції для одних користувачів давати вищі оцінки, ніж інші, а для деяких предметів отримувати вищі оцінки, ніж інші.

Необхідно включити такі обставини, які такі, що не передбачають взаємодії між користувачем передвісники предметом, та визначити їх як базові передвісники. Оскільки ці передвісники мають тенденцію щоб захопити більшу частину спостережуваних вхідних даних, дуже важливо точно їх змоделювати. Таке моделювання дозволяє виділити ту частину сигналу, яка справді представляє взаємодію між користувачем-предметом, і піддати його більш відповідним моделям переваг користувача.

Базовий прогноз для невідомого рейтингу r_{ui} позначається b_{ui} та враховує ефекти користувача та товару:

$$b^{ui} = \mu + b^{u} + b^{i} \tag{2.1}$$

де b_u та b_i - спостережувані відхилення користувача u та предмета і відповідно від середнього,

μ - загальна середня оцінка.

Наприклад, припустимо, що нам потрібен базовий прогноз для рейтингу фільму "Титанік" від користувача Василя. Тепер скажімо, що середній рейтинг усіх фільмів, μ, становить 3,7 зірки. Крім того, "Титанік" кращий за середній фільм, тому його, як правило, оцінюють на 0,5 зірки вище середнього. З іншого боку, Василь є критичним користувачем, який, як правило, оцінюється на 0,3 зірки нижче середнього. Таким чином, базовим провісником для рейтингу "Титаніка" Василем було б 3,9 зірки, обчислюючи 3,7-0,3 + 0,5.

Але найпоширенішим підходом до КФ є алгоритм заснований на моделях сусідства. Його початкова форма, яку поділяли практично всі попередні системи КФ, орієнтована на користувача. Такі орієнтовані на користувача методи оцінюють невідомі рейтинги на основі записаних рейтингів однодумців. Пізніше аналогічний предметно-орієнтований підхід стали популярними. У цих методах рейтинг оцінюється з використанням відомих рейтингів, зроблених тим самим користувачем щодо подібних предметів.

Крім того, більш піддаються орієнтовані на предмети методи пояснення міркувань за прогнозами. Це пов’язано з тим, що користувачі знайомі з предметами, які раніше їм надавали перевагу, але, як правило, не знають тих, хто нібито сподобався однодумцям[20].

Центральним для більшості підходів, орієнтованих на предмети, є показник подібності між предметами, де s_{ij} позначає подібність i та j. Часто він базується на коефіцієнті кореляції Пірсона. Нашою метою є прогнозування r_{ui} - невизначене значення користувачем u для предмета i. Використовуючи міру подібності, ми визначаємо k предметів, оцінених u, які найбільш подібні до i. Цей набір k сусідів позначається $S^k(i; u)$. Прогнозоване значення r_{ui} приймається як середньозважене значення рейтингів для сусідніх предметів, представлене у формулі 2.2.)

(2.2)

Деякі вдосконалення цієї схеми добре практикуються для явного зворотного зв'язку, наприклад, виправлення помилок, спричинених різними середніми оцінками різних користувачів та предметів. Ці модифікації менш актуальні для неявних наборів даних зворотного зв’язку, де замість того, щоб мати рейтинги, які мають однакову шкалу, ми використовуємо частоти, на яких предмети споживає той самий користувач. Частоти для різних користувачів можуть мати дуже різний масштаб залежно від програми, і менш зрозуміло, як розрахувати подібності [21].

Рекомендовані системи покладаються на різні типи вхідних даних. Найзручнішим є високоякісний явний відгук, який включає явне введення користувачами щодо їх інтересу до продуктів. Наприклад, Netflix збирає рейтинги зірок для фільмів, а користувачі TiVo вказують свої уподобання до телевізійних шоу, натискаючи кнопки великих пальців вгору / вниз. Однак явні відгуки не завжди доступні. Таким чином, рекомендуючі можуть зробити висновки щодо переваг користувачів на основі більш рясних неявних відгуків, які побічно відображають думку через спостереження за поведінкою користувачів. Типи неявних відгуків включають історію покупок, історію перегляду, шаблони пошуку або навіть рухи миші. Наприклад, користувачеві, який придбав багато книг того самого автора, напевно подобається цей автор.

Переважна більшість літератури в цій галузі зосереджена на обробці явних відгуків; можливо, завдяки зручності використання цього виду чистої інформації. Однак у багатьох практичних ситуаціях системи, що рекомендують, повинні бути зосереджені на неявних зворотних зв’язках. Це може відображати небажання користувачів оцінювати товари або обмеження системи, яка не може зібрати явний відгук. У неявній моделі, коли користувач дає згоду на збір даних про використання, додаткові явні відгуки (наприклад, оцінки) від користувача не потрібні. Ця робота проводить дослідження алгоритмів, спеціально придатних для обробки неявного зворотного зв'язку. Він відображає деякі основні уроки та розробки, які були досягнуті під час розробки механізму рекомендацій для телевізійних шоу.

Наші обмеження заважають нам активно збирати явні відгуки користувачів, тому система базувалася виключно на неявних відгуках. Дуже важливо визначити унікальні характеристики неявного зворотного зв’язку, які перешкоджають безпосередньому використанню алгоритмів, розроблених з урахуванням явного зворотного зв’язку. Далі ми перелічимо основні характеристики [22]:

1. Відсутність негативних відгуків. Спостерігаючи за поведінкою користувачів, ми можемо зробити висновок, які товари їм, мабуть, подобаються, і, отже, вирішили споживати. Однак важко достовірно зробити висновок, які предмети користувачеві не сподобались. Наприклад, користувач, який не дивився певне шоу, міг би це зробити тому, що йому не подобається шоу або просто тому, що він не знав про шоу або воно було недоступне для перегляду. Ця фундаментальна асиметрія відсутня в явних відгуках, коли користувачі повідомляють нам і те, що їм подобається, і те, що їм не подобається. Це має кілька наслідків.

Наприклад, явні рекомендатори, як правило, зосереджуються на зібраній інформації - тих парах користувач-предмет, про які ми знаємо їх рейтинги, - що забезпечують збалансовану картину щодо переваг користувача. Таким чином, решта користувальницьких відносин, які, як правило, складають величезну кількість

Більшість даних розглядаються як «відсутні дані» і не включаються в аналіз. Це неможливо за умови неявного зворотного зв’язку, оскільки концентруючись лише на зібраному відгуку, ми залишатимемо позитивні відгуки, сильно спотворюючи повний профіль користувача. Отже, важливо вирішити також відсутні дані, саме там, як очікується, буде знайдено найбільше негативних відгуків.

2. Неявний зворотний зв'язок по своїй суті є галасливим. Хоча ми пасивно відстежуємо поведінку користувачів, ми можемо лише здогадуватися про їх уподобання та справжні мотиви. Наприклад, ми можемо розглядати поведінку

покупців для окремої людини, але це не обов'язково вказує на позитивну думку про товар.

Можливо, товар був придбаний у подарунок, або, можливо, користувач був розчарований товаром.

3. Числове значення явного зворотного зв'язку вказує на перевагу, тоді як числове значення неявного зворотного зв'язку свідчить про впевненість. Системи, засновані на явних зворотних зв’язках, дозволяють користувачеві висловити свій рівень переваг, наприклад оцінка зірками від 1 ("повністю не подобається") та 5 ("дуже подобається"). З іншого боку, числові значення неявного зворотного зв'язку описують частоту дій, наприклад, скільки часу користувач переглядав певне шоу, як часто користувач купує певний товар тощо. Більше значення не свідчить про вищі переваги [22].

Наприклад, найулюбленішим шоу може бути фільм, який користувач перегляне лише один раз, тоді як є серія, яка користувачеві дуже подобається, і тому вона дивиться щотижня. Однак числове значення зворотного зв'язку, безумовно, корисно, оскільки воно говорить нам про впевненість у певному спостереженні. Одноразова подія може бути спричинена різними причинами, які не мають нічого спільного з уподобаннями користувачів. Однак повторювана подія, швидше за все, відображатиме думку користувача.

Переважна більшість літератури в цій галузі зосереджена на обробці явних відгуків; можливо, завдяки зручності використання цього виду чистої інформації. Однак у багатьох практичних ситуаціях системи, що рекомендують, повинні бути зосереджені на неявних зворотних зв’язках. Це може відображати небажання, яке не в змозі отримати явний відгук. У неявній моделі, коли користувач дає згоду на збір даних про використання, додаткові явні відгуки (наприклад, рейтинги) не потрібні користувацької частини. Він відображає деякі основні уроки та розробки, які були досягнуті під час розробки механізму рекомендацій для телевізійних шоу. Наша настройка заважає нам активно збирати явні відгуки користувачів, тому система базувалася виключно на неявних зворотних зв’язках - аналізуючи звички спостереження анонімних

користувачів [23]. Дуже важливо визначити унікальні характеристики неявного зворотного зв'язку, які перешкоджають безпосередньому використанню алгоритмів які були розроблені з урахуванням чітких відгуків. Основні характеристики такого підходу:

1. Відсутність негативних відгуків. Спостерігаючи за поведінкою користувачів, ми можемо зробити висновок, які товари їм, мабуть, подобаються, і, отже, вирішили споживати. Однак важко достовірно зробити висновок, які предмети користувачеві не сподобались. Наприклад, користувач, який не дивився певне шоу, міг би це зробити тому що їй не подобається шоу або просто тому, що вона не знала про шоу або була недоступна для перегляду. Ця фундаментальна асиметрія відсутня в явних відгуках, коли користувачі повідомляють нам і те, що їм подобається, і те, що їм не подобається. Це має кілька наслідків.

Наприклад, явні рекомендатори, як правило, зосереджуються на зібраній інформації - тих парах користувач-предмет, про які ми знаємо їх рейтинги, - що забезпечують збалансовану картину щодо уподобань користувача. Таким чином, решта користувальницьких відносин, які, як правило, складають величезну кількість Більшість даних розглядаються як «відсутні дані» і не включаються в аналіз [24]. Це неможливо за умови неявного зворотного зв’язку, оскільки концентруючись лише на зібраному відгуку, ми залишатимемо позитивні відгуки, сильно спотворюючи повний профіль користувача. Отже, важливо вирішити також відсутні дані, саме там, як очікується, буде знайдено найбільше негативних відгуків.

2. Неявний зворотний зв'язок по своїй суті є галасливим. Хоча ми пасивно відстежуємо поведінку користувачів, ми можемо лише здогадуватися про їх уподобання та справжні мотиви. Наприклад, ми можемо розглядати поведінку покупців для окремої людини, але це не обов'язково вказує на позитивну думку про товар. Можливо, товар був придбаний у подарунок, або, можливо, користувач був розчарований товаром. Ми можемо бачити, що телевізор перебуває на певному каналі в певний час, але глядач може спати [25].

3. Числове значення явного зворотного зв'язку вказує на перевагу, тоді як числове значення неявного зворотного зв'язку свідчить про впевненість. Системи, засновані на явних зворотних зв’язках, дозволяють користувачеві висловити свій рівень переваг, наприклад оцінка зірками від 1 ("повністю не подобається") та 5 ("дуже подобається"). З іншого боку, числові значення неявного зворотного зв'язку описують частоту дій, наприклад, скільки часу користувач переглядав певне шоу, як часто користувач купує певний товар тощо. Більше значення не свідчить про вищі переваги. Для Наприклад, найулюбленішим шоу може бути фільм, який користувач перегляне лише один раз, тоді як є серія, яка користувачеві дуже подобається, і тому вона дивиться щотижня. Однак числове значення зворотного зв'язку, безумовно, корисно, оскільки воно говорить нам про впевненість у певному спостереженні. Одноразова подія може бути спричинена різними причинами, які не мають нічого спільного з уподобаннями користувачів. Однак повторювана подія, швидше за все, відображатиме думку користувача.

4. Оцінка рекомендатора з неявними відгуками вимагає відповідних заходів. У традиційній обстановці, де користувач вказує числовий бал, існують чіткі показники, такі як середньоквадратична помилка для вимірювання успіху в прогнозуванні. Однак при неявних моделях ми повинні враховувати наявність товару, конкуренцію за товар з іншими предметами та повторювати відгуки. Наприклад, якщо ми збираємо дані про перегляд телевізора, незрозуміло, як оцінювати шоу, яке переглядали неодноразово, або як порівнювати два шоу, які одночасно, і, отже, користувач не може переглядати обидва.

2.2 Особливості методів контентної фільтрації

Системи, що реалізують підхід, що ґрунтується на вмісті, аналізують набір документів та / або описи предметів, які раніше були оцінені користувачем, та будують модель або профіль інтересів користувача на основі особливостей об’єктів, оцінених цим користувачем.

Більшість систем рекомендацій, що базуються на вмісті, використовують відносно прості моделі пошуку, такі як узгодження ключових слів або Векторна космічна модель (VSM) з базовим зважуванням TF-IDF. VSM - це просторове представлення текстових документів. У цій моделі кожен документ представлений вектором у n-мірному просторі, де кожен вимір відповідає терміну із загального словника даної колекції документів[26].

Формально кожен документ представлений у вигляді вектора ваг термінів, де кожен ваговий коефіцієнт вказує на ступінь зв'язку між документом і терміном. Нехай D = {d1, d2, ..., dN} позначає набір документів або корпус, а T = {t1, t2, ..., tn} - словник, тобто набір слів у корпус. T отримується шляхом застосування деяких стандартних операцій з обробки природними мовами, таких як токенізація, видалення стоп-слів та стемінг. Кожен документ dj представлений як вектор у n-мірному векторному просторі, тому dj = {w1j, w2 j, ..., dn j}, де wk j - вага терму tk у документі dj.

Представлення документів у VSM порушує дві проблеми: зважування термінів та вимірювання подібності вектора ознак. Найчастіше застосовувана схема зважування термінів, зважування TF-IDF (Term Frequency-Inverse Document Frequency), базується на емпіричних спостереженнях щодо тексту:

– рідкісні терміни не менш актуальні, ніж часті (припущення IDF);

– багаторазове входження терміна в документ є не менш актуальним, ніж одиничне входження (припущення TF);

– довгі документи не надають перевагу коротким документам (припущення про нормалізацію).

Іншими словами, терміни, які часто трапляються в одному документі (TF = термін-частота), але рідко в решті корпусу (IDF = зворотна частота документа), швидше за все, мають відношення до теми документа. Крім того, нормалізація результуючих векторів ваги перешкоджає більшим шансам на отримання довших документів.

Як зазначалося раніше, для визначення близькості між двома документами необхідний показник подібності. Для опису близькості двох

векторів було проведено багато заходів схожості; серед цих мір найпоширенішою є подібність косинусів.

У системах рекомендацій на основі вмісту, що спираються на VSM, і профілі користувачів, і предмети представлені у вигляді зважених векторів термінів. Прогнози зацікавленості користувача певним предметом можна отримати шляхом обчислення подібності косинусів.

Недоліками такого підходу є:

– обмежений аналіз вмісту: якщо вміст не містить достатньо інформації для точної дискримінації предметів, рекомендація може бути неточною;

– надмірна спеціалізація: метод, що базується на вмісті, забезпечує граничний ступінь новизни, оскільки він повинен відповідати особливостям профілю та предметів. Повністю досконала фільтрація на основі вмісту може не припустити нічого "здивовуючого".

2.3 Удосконалення методів формування онлайн-рекомендацій з урахуванням негативних неявних відгуків та заповненості профілю користувача

Результатом даної роботи стане удосконалений метод формування онлайн-рекомендацій, що буде використовувати метод колаборативної фільтрації з неявним зворотнім зв’язком, а також контентну фільтрацію, що вирішить деякі проблеми колаборативної фільтрації [27]. Структура удосконаленого методу представлена на рисунку 2.3.

Основною відмінністю методу, що розробляється, стане тещо ми будемо комбінувати одразу два різних методи, один з яких удосконалимо, щоб усунути частину його недоліків. Такий підхід надасть можливість використовувати сильні сторони кожного з методів або нівелювати їх недоліки в залежності від заданого контексту в певний момент часу, що, в свою чергу, дозволить, збільшити ефективність та зменшити час надання рекомендацій.

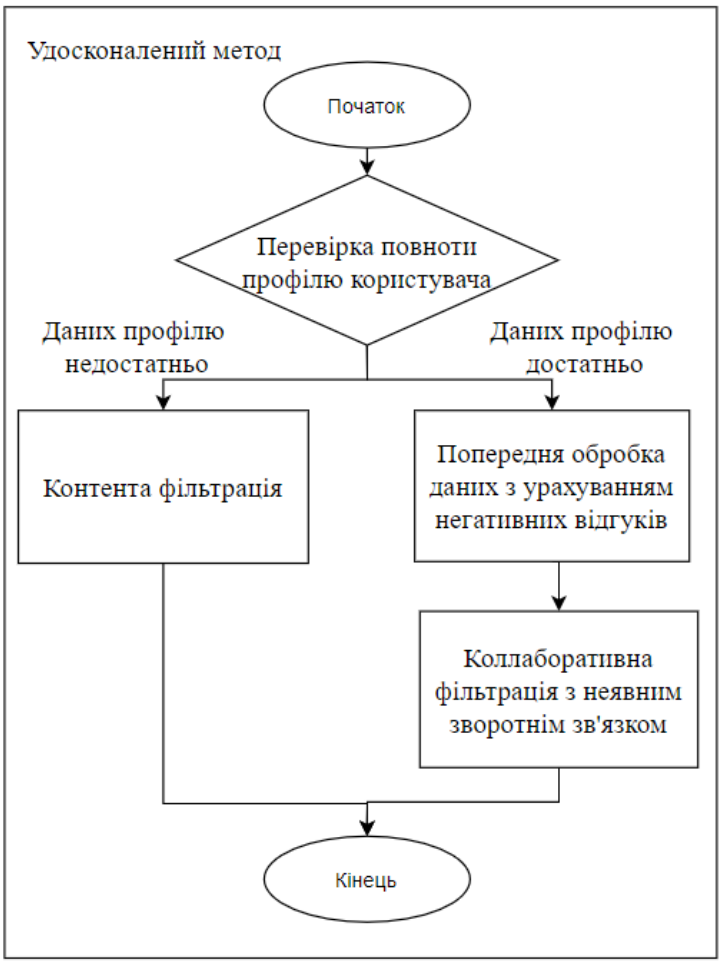

Рисунок 2.3 – Структура удосконаленого методу

Перевагою використовуваного методу є використання негативного відгуку користувача та та використання одразу обох методів зазначених вище, що дозволить впоратись з проблемою «холодного старту» для користувача а також підвищити точність надання рекомендацій для користувачів з уже заповненим профілем [28].

Недоліком удосконаленого методу є висока алгоритмічна складність при використанні колаборативної фільтрації, що в свою чергу потребуватиме великої обчислювальної здатності від використовуванної техніки, а також знижена точність надання рекомендацій для користувачів з незаповненим профілем.

Метод, що досліджується, використовує наступні вхідні дані.

Для колаборативної фільтрації використовуються дані про активність користувачів у інтернет магазині.

$$E = \{u^j, i^k, e^l\} \quad (2.3)$$

де - дані споживача;

i_k - ідентифікатор товару;

e_l - тип події(переглянув, додав у корзину і т.д.).

Для контентної фільтрації використовуються дані про товари магазину у наступному вигляді

$$G = \{t, i^k, pn^n, pv^n\} \quad (2.4)$$

де t – позначка часу;

i_k - ідентифікатор товару;

pn_n - назва властивості товару;

pv_n - значення властивості товару.

Метод, що досліджується, містить наступні фази:

– фаза 1. Вибір використовуваного методу;

– фаза 2. Формування рекомендацій за допомогою колаборативної фальтрації;

– фаза 3. Формування рекомендацій за допомогою контентної фальтрації.

Розглянемо ці фази одна за одною.

Під час фази 1. Ми робимо вибір алгоритму формування рекомендацій в залежності від повноти профілю користувача. Для цього отримаємо кількість взаємодій з системою для поточного користувача.

(2.5)

де – сума усіх взаємодій користувача .

Тепер отримаємо множину усіх сум взаємодій користувачів системи.

(2.6)

Після цього порівняємо суму взаємодій поточного користувачаз медіаною взаємодій з усіх користувачів.

(2.7)

де – функція медіани.

У такому випадку, коли сума взаємодій поточного користувача буде більшою за медіану зваємодій усіх користувачів, тоді ми використовуємо алгоритм колаборативної фільтрації з неявним відгуком. Якщо меншою – тоді використовуємо контенту фільтацію.

Далі розпишемо етапи фази 2, а саме формування рекомендацій для колаборативної фільтрації.

Етап 1.

По-перше, нам потрібно формалізувати поняття впевненості, яке вимірюють змінні r_{ui}. Складемо з цих оцінок матрицю оцінок R розміром M на N, де M - кількість користувачів, а N - кількість предметів. Ця матриця досить розріджена, оскільки більшість користувачів взаємодіють лише з декількома предметами. Немодифікована версія алгоритму, зазвичай, розцінює негативні значення відгуків, як помилкові. Для модифікованого методу, вони такими вважатися не будуть, а будуть використовуватися так само, як і у випадках роботи з явними відгуками[29].

Для того, щоб це виконати, потрібно обробити вхідні дані наступним шляхом. Спочатку перетворити тип події у її числовий еквівалент наступним чином.

(2.8)

де числовий еквівалент взаємодії користувача u та товару i.

Оскільки подій для кожної пари користувач-товар може бути декілька, нам потрібно откримати суму числових еквівалентів взаємодій в залежності від того були повернення товарів чи ні.

(2.7)

Результатом цього етапу стане матриця розміром M на N, де кожен предмет матриці r_{ui} – це неявна оцінка користувача u для предмету i.

На другому етапі нам необхідно розкласти цю матрицю на дві окремі менші матриці: одну з розмірами M на K, яка буде нашими прихованими векторними характеристиками для кожного користувача (X), і другу з розмірами K на N, яка матиме наші вектори прихованих предметів для кожного предмета (Y) . Помноження цих двох матриць об'єктів наближається до початкової матриці, але тепер у нас є дві матриці, щільні, включаючи ряд прихованих ознак K для кожного з наших предметів та користувачів. Для того, щоб вирішити для X та Y, ми могли б або використовувати SVD (який вимагав би інвертування потенційно дуже великої матриці і був би обчислювально дорогим), щоб точніше вирішити факторизацію, або застосувати ALS для його наближення[30]. У випадку ALS нам потрібно вирішувати лише один вектор ознак за раз, що означає, що його можна запускати паралельно. Для цього ми можемо випадковим чином ініціалізувати X і розв’язати для Y. Потім ми можемо повернутися назад і розв’язати X, використовуючи наше рішення для Y. Продовжуємо ітерацію вперед і назад, поки ми не отримаємо збіжність, що якнайкраще наближається до R.

$$x^u = (Y^T C^u Y + \lambda I)^{-1} Y^T C^u p(u) \qquad (2.8)$$

де x_u – вектор факторів користувача u;

Y - матриця прихованих факторів предметів;

λ – константа, що обмежує впевненість у наданні рекомендацій;

p(u) – вектор, що зберігає усі переваги користувача u;

C^u – діагональна матриця, що містить у собі усі змінні впевненості для переваг p(u).

$$y^i = (X^T C^i X + \lambda I)^{-1} X^T C^i p(i) \quad (2.9)$$

де x_u – вектор факторів предметів i; X - матриця прихованих факторів користувачів;

λ – константа, що обмежує впевненість у наданні рекомендацій;

p(i) – вектор, що зберігає усі переваги для предмету i;

C^i – діагональна матриця, що містить у собі усі змінні впевненості для переваг p(i).

Результатом цього етапу стануть матриці прихованих факторів користувачів та предметів X та Y.

Етап 3.

Після того, як це буде закінчено, ми можемо просто взяти точковий добуток X та Y, щоб побачити, яким буде прогнозований рейтинг для конкретної взаємодії користувача / предмета, навіть якщо попередньої взаємодії не було.

$$R' = XY \quad (2.10)$$

Результатом цього етапу стане матриця R', для якої кожен предмет r_{ui} – це оцінка предмету і користувачем u.

Тепер розберемо фазу 3, а саме - контенту фільтрацію.

На першому етапі цієї фази небхідно представити кожен із предметів як рядок, для подальшої обробки. Так, наприклад, для дубової шафи висотою в два метри, дзеркальними дверцятами та вішалками у комплекті рядок для обробки буде виглядати наступним чином: «Шафа дуб, 2м, дзеркальні дверцята, вішалки».

$$S^i = (f^{i1}, f^{i2}, \ldots, f^{in}) \quad (2.11)$$

де S_i – рядок, що репрезентує предмет i;

а f_{in} – значення особливості n предмету i.

Результатом цього етапу стане набір рядків, що позначають особливості предметів.

На другому етапі необхідно провести обчислення важливості особливості предметів за допомогою алгоритму TF-IDF, який оцінює важливість слова в контексті документа(в даному випадку строки, яка є репрезентацією предмету), що є частиною колекції документі. Вага деякого слова пропорційна частоті вживання цього слова в документі і обернено пропорційна частоті вживання слова в усіх документах колекції[31]. Складання матриці, що містить оцінку TF-IDF щодо особливостей документа або предмета в даному випадку. Крім того, алгоритм буде ігнорувати «слова зупинки» - це просто слова, які не додають значущої цінності нашій системі, наприклад, «an», «is», «the», а отже, ігноруються системою.

(2.12)

де це число входжень слова t в документ;

- загальне число слів в даному документі.

(2.13)

де - число документів в коллекції;

- число документів з колекції D, в яких зустрічається t (коли n не дорівнює 0)

$$x^i = tf(i, d) * idf(i, D) \quad (2.14)$$

де x_i – оцінка TF-IDF для предмету i.

Результатом цього етапу стане набір оцінок TF-IDF для кожного з предметів для усього набору предметів.

Етап 3.

Потім розрахуовуємо косинусну схожість кожного предмета з кожним іншим предметом набору даних за формулою лінійного ядра(англ. linear kernel), а потім розташуємо їх відповідно до їх подібності з предметом ik.

$$k(x, x) = x^{i^T} x^i \quad (2.15)$$

Результатом, цього етапу стануть значення схожості предметів між собою, на основі яких і будуть сформовані рекомендації.

Не залежно від обраного методу, наприкінці виконання алгоритму отримуємо набор рекомендованих товарів для користувача[32]. Різниця, таким чином, буде лише у даних, завдяки яким цей набор було складено: історія активності користувача, або набір атрибутів товарів.

3 ДОСЛІДЖЕННЯ ОТРИМАНИХ НАУКОВИХ РЕЗУЛЬТАТІВ

3.1 Порівняння методів формування онлайн-рекомендацій

Раніше було розглянуто два різних методи формування онлайн рекомендацій: колаборативну фільтрацію та контенту фільтрацію. Кожен з розглянутих підходів має певний набір переваг в порівнянні з іншим, так і поступається, якщо буде використаний у певному контексті. Тепер нам буде необхідно порівняти продуктивність з запропонованих методів, а також методу, що розробляється, при роботі з основними обмеженнями методів формування рекомендацій. Слабкі та сильні сторони кожного з методів будут описані та узагальнені у Таблиці 3.1.

Колаборативна фільтрація добре пряцює в умовах, коли профіль користувача складено, і є можливість достатньо точно передбачити його вподобання, використовуючи історію активності користувачів, яких можно вважати схожими на активного. Також, за даного підходу система може допомогти користувачу знайти нові вподобання, тому що система, при наданні певного набору рекомендацій, спирається на історію активності не лише активного користувача, а й істрію активності користувачів, що є схожими на нього.

Крім цього, системі не потрібно обробляти масиви даних, що стосуються інформації про користувача, або набору особливостей предметів. У цьому випадку системи потрібно знати лише які користувачі взаємодіяли з якими предметами.

Однак, у колаборативної фільтрації є і свої недоліки. Серед таких можна зазначити проблему «холодного старту» - знижену продуктивність системи щодо нових предметів та нових користувачів. Холодний старт можна розглядати як додаткову проблему охоплення, оскільки він вимірює охоплення системи певним набором предметів та користувачів. На додаток до вимірювання, наскільки великий обсяг предметів холодного запуску або

користувачів, також може бути важливим виміряти точність системи для цих користувачів та предметів[33]. Зосереджуючись на предметах холодного старту, ми можемо використовувати поріг, щоб визначитися з набором холодних предметів. Наприклад, ми можемо вирішити, що холодні предмети - це лише предмети, що не мають оцінок чи доказів використання, або предмети, що існують у системі менше, ніж певний проміжок часу (наприклад, день), або предмети, які мають менше ніж заздалегідь визначена кількість доказів (наприклад, менше 10 оцінок). Більш загальним способом є розгляд «холодності» предмета, використовуючи або кількість часу, який він існує в системі, або кількість даних, зібраних для нього. Тоді ми можемо налаштувати систему так, щоб частіше омендувати більш холодні предмети і менше гарячі предмети, які користуються популярністю. За таких умов система краще рекомендує холодні речі за ціною зниженої точності для більш гарячих. Це може бути бажаним через інші міркування, такі як новизна та випадковість. Тим не менше, при обчисленні точності системи на холодних предметах може бути розумно оцінити, чи є компроміс з усією точністю системи.

У противагу цьому контентна фільтрація не має такої проблеми, оскільки схожість між двома користувачами в методах, що базуються на користувачах, що визначає сусідів користувача, зазвичай отримується шляхом порівняння оцінок, зроблених цими користувачами для тих самих предметів[34]. Розглянемо систему, яка містить 10 000 оцінок, зроблених 1000 користувачами на 100 предметів, і припустимо, що для цілей цього аналізу оцінки розподіляються рівномірно по предметах. Середня кількість користувачів, доступних як потенційні сусіди, становить приблизно 650. Однак середня кількість загальних рейтингів, що використовуються для обчислення подібності лише 1. З іншого боку, метод контентної фільтрації зазвичай обчислює схожість між двома предметами, порівнюючи їхні характеристики. Тому знайти рекомендовані предмети буде набагато легше.

Також серед переваг такого підходу можна високу швидкість виконання, що є дуже важливою характеристикою для систем, що формують рекомендації

для систем електронної комерції, оскільки час відгуку для сторінки не повинен бути більше ніж одна секунда.

Звичайно, як і у будь-якого методу, у контентої філтрації є і свої недоліки. Серед них можна відзначити те, що їхня точнітсть дуже залежнить від наповненості даних про предмети. Наприклад, якщо для усіх товарів у системі електронної коммерції буде вказаний лише виробник, тоді система буде рекомендувати користувачу лише товари цього бренду, що може бути дуже поганим, якщо цей виробник представлений у цьому магазині лише однією шафою, яку переглядає користувач, та декількаами десятками приборів для освітлення. У такому випадку, користувачу, який хоче придбати шафу будуть рекомендовані лише освітлюючі прибори, і загальне враження від магазину[35], а разом із цим вірогідність того, що користувач обере саме його значно знизиться.

Як вже зазначалося раніше, метод, що досліджується, об’єднує в собі одразу два методи формування онлайн-рекомендацій: контенту фільтрацію та колаборативну фільтрацію на основі зворотнього зв’язку. При цьому, на відміну, від класичного підходу використовує негативний зворотній зв’язок.

І саме це можна відзначити як основну перевагу удосконаленого методу в порівнянні з класичною колаборативною фільтрацією на основі неявного зворотнього зв’язку. Такі системи не використовують негативні відгуки за різних причин. Серед таких можна виокремити неможливість їх отримати, як на приклад, у системах формування онлайн рекомендацій для систем, що надають своїм користувачам контент певного роду, як, наприклад, стриммінговий сервіс Netflix не може отримати негативний неявний зворотній зв’язок від користувача тому, що їх система не може знати напевно, чому користувач не дивився той чи інший фільм або серіал: через те, що від дивився його десь раніше, чув про нього, і він йому не подобається, або ж просто не знає про його існування. У рамках предметної області системи електронної комерції такі дані отримати можливо. Серед таких можуть бути видалення товару з корзини або повернення товару магазину після покупки. Таким чином ці дані дозволять

отримати більш повну картину про вподобання користовачів, тим самим трохи наблизивши за точністю рекомендацій удосконалений метод до методів колаборативної фільтрації з явним зв’язком.

Також, метод, що досліджується частково вирішує проблему «холодного старту» для нових користувачів, оскільки за недостатньої заповненості профілю користувача, система просто буде рекомендувати предмети, засновуючись на їх схожості між собою, та не беручи до уваги інші дані[36].

Але з останнього пункту витікає основний недолік методу, що досліджується. Він полягає у зниженій точності надання рекомендацій користувачам з недостатньо заповненим профілем. У такому випадку точність буде залежати від повноти даних про предмети, у нашому випадку товари, що переглядаються, купуються тощо.

Також до недоліків удосконаленого методу можна віднести те, що для нормальної працездатності йому необхідна техніка з великою обчислювальною здатністю, оскільки кількість даних, що буде оброблятися таким чином дуже велика.

Таблиця 3.1 - Порівняння різних методів формування онлайн-рекомендацій

Метод	Переваги	Недоліки
1	2	3
Колаборативна фільтрація Не має потреби аналізувати дані користувачів. Не має потреби аналізувати дані предметів.	Випадковість(англ. Serendipity) – можливість системи рекомендувати користувачам предмети, про які користувач міг би і не подумати[37].	Чим більше користувачів взаємодіють з певним товаром, тим менше система рекомендує його альтернативи.

Продовження таблиці 3.1 - Порівняння різних методів формування онлайн-рекомендацій

1	2	3

Якість надання рекомендацій збільшується разом із збільшенням кількості взаємодій користувачів з предметами.	Системі не потрібно обробляти будь-яку інформацію про користувача. Проблема холодного старту для нових товарів. Проблема холодного старту для нових предметів.	Необхідність працювати з дуже розрідженими даними.
Контентна фільтрація	Не потребує заповненості профілю користувача для формування рекомендацій. Системі не потрібно обробляти будь-яку інформацію про користувача. Не має проблеми «холодного старту» для користувачів.	Оскільки представлення предметів певною мірою розроблено вручну, цей метод вимагає знання знань домену. Отже, модель може бути настільки ж хорошою, як і ручно розроблені функції. Метод може давати рекомендації лише на основі існуючих інтересів користувача.

Продовження таблиці 3.1 - Порівняння різних методів формування онлайн-рекомендацій

1	2	3
Удосконалений метод	Якість надання рекомендацій збільшується разом із збільшенням кількості взаємодій користувачів з предметами. Випадковість(англ. Serendipity) – можливість системи рекомендувати користувачам предмети, про які користувач міг би і не подумати. Не потребує повної заповненості профілю користувача для формування рекомендацій.	Іншими словами, модель має обмежені можливості розширити існуючі інтереси користувачів. Велика залежність точності на заповненість даних про предмети.Точність за відсутності заповненого профілю користувача дуже залежить від заповненості даних про предмети. Необхідність у великій обчислювальній потужності від техніки.

	Частково вирішує проблему «холодного старту» для нових користувачів.	

3.2 Опис технології побудови онлайн-рекомендацій з використанням удосконаленого методу

Технологія використання удосконаленого методу, що комбінує у собі колаборативну фільтрацію з неявним зворотнім зв'язком та контенту фільтрацію складається з наступних етапів (рисунок 3.1).

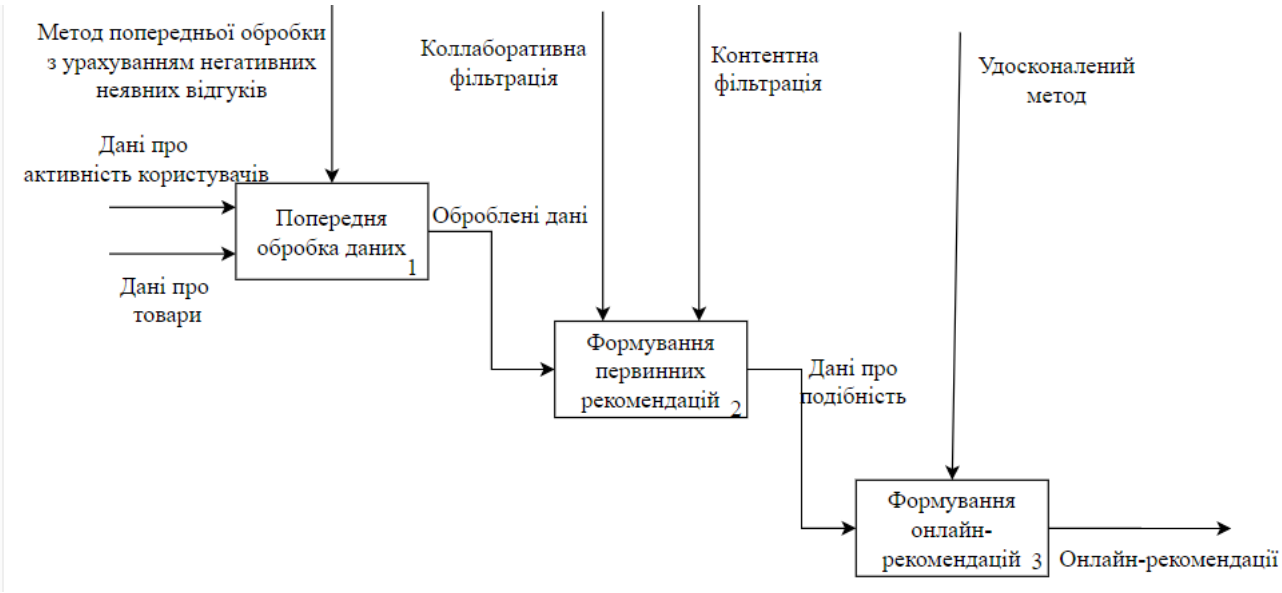

Рисунок 3.1 – Технологія використання удосконаленого методу

Етап попередньої обробки даних[38]. На цьому етапі для колаборативної фільтрації попередьо збережені дані про активність користувачів фільтруються та приводяться до такого вигляду, з яким простіше буде працювати на наступних етапах. Для контентної фільтрації дані про предмети приводяться до формату строки та зберігаються для подальшої обробки. На етапі формування первинних рекомендацій за методу колаборативної фільтрації будується матриця передбачених оцінок користувачів стосовно предметів. Для контентної фільтрації на цьому етапі обчислюються косинуси подібності для предметів. Для обох методів оброблені дані зберігаються для подальшого використання. На етапі формування онлайн-рекомендацій попередньо збережені дані повертаються, в залежності від контексту використання.

4 ПРАКТИЧНЕ ВИКОРИСТАННЯ ДОСЛІДЖЕНИХ РЕЗУЛЬТАТІВ

4.1 Програмна реалізація використання удосконаленого методу

Метою експерименту можна визначити знаходження значення точності надання рекомендацій у вигляді показника AUC (англ. Area under ROC curve, площа під ROC-кривої), тобто площу, обмежену ROC-кривою[39]. Чим вище показник AUC, що спостерігається, тим точнішими можна вважати рекомендації. За використання даного показника значення 0,5 демонструє непридатність наданих рекомендацій через занадто низьку вірогіднисть надання рекомендації, що зацікавить користувача.

Крива ROC або крива робочої характеристики, є графічним показником, який демонструє здатність двійкової системик ласифікаторів. У даному випадку він показує здатність системи надавати рекомендації, що будуть відповідати вподобанням користувачів.

Нам необхідно знайти показники AUC для классичної колаборативної фільтрації та модифікованого методу. При чому показник AUC буде розраховуватися як для випадку, коли профіль користувача заповнений, так і коли його повноти недостатньо, і буде використовуватися контентна фільтрація для предметів з якими користувач взаємодіє.

В якості мови програмування, що використовувалася для експерименту, була обрана java. Java - строго типізована об'єктно-орієнтована мова програмування загального призначення, розроблена компанією Sun Microsystems (в подальшому придбаної компанією Oracle). Вона була обрана, оскільки більшість сучасних систем електронної комерції/інтернет-магазинів, використовують платформи, що прискорюють розробку, а значна більшість цих платформ використовує java. Отже, використання java можна вважати доцільним, адже воно значно полегшить інтеграцію з існуючими інформаційними системами.

Експеримент проводився у IntelliJ IDEA - комерційному інтегрованому середовищі розробки для різних мов програмування (Java, Python, Scala, PHP та ін.) від компанії JetBrains.

Серед бібліотек на мові програмування java, що дозволять полегшити проведення експерименту, можна зазначити Apache Spark. Apache Spark (від англ. Spark - іскра, спалах) - фреймворк з відкритим вихідним кодом для реалізації розподіленої обробки неструктурованих і слабоструктурованих даних, що входить в екосистему проектів Hadoop.

В якості бази даних використовувалася MongoDB - документо-орієнтована система керування базами даних (СКБД) з відкритим вихідним кодом, яка не потребує опису схеми таблиць. MongoDB займає нішу між швидкими і масштабованими системами, що оперують даними у форматі ключ/значення, і реляційними СКБД, функціональними і зручними у формуванні запитів. Ця бд була обрана через наступні переваги:

– MongoDB - це база даних документів, в якій одна колекція містить різні документи, у яких кількість полів, зміст і розмір можуть відрізнятися від одного документа до іншого;

– структура окремого об'єкта чітка;

– жодних складних об’єднань;

– глибокі можливості запитів. MongoDB підтримує динамічні запити в документах з використанням мови запитів на основі документів, яка майже така ж потужна, як SQL;

– простота масштабування - MongoDB легко масштабувати;

– перетворення / зіставлення об'єктів програми в об'єкти бази даних не потрібне;

– використовує внутрішню пам’ять для зберігання (віконного) робочого набору, що забезпечує швидший доступ до даних.

Отже, для початку нам потрібні дані з якими ми будемо працювати.

Для експерименту була використана вибірка, складається з трьох файлів: файлу з даними поведінки, файлу із властивостями предметів та файлу[40], що

описує дерево категорій. Дані зібрані з реального веб-сайту електронної комерції. Це необроблені дані, тобто без будь-яких перетворень вмісту. Метою публікації є мотивація досліджень у галузі рекомендованих систем із неявним зворотним зв'язком.

Дані про поведінку - такі події, як кліки, додання товарів в кошики, транзакції, являють собою взаємодії. Відвідувач може здійснити чотири типи подій, а саме «view», «addtocart», «transaction», «remove». Загалом є 2 759 469 подій, включаючи 2 664 312 переглядів, 69 332 доданих до кошиків, 22 457 транзакцій, 3368 повернень товарів, здійснених 1 407 580 унікальних відвідувачів. Як приклад події представленної у вибірці можна переглянути наступний рядок:

“1439694000000,1, view, 100», означає що користувач з ідентифікатором рівним 1, переглянув предмет з id = 100 на 1439694000000 (мітка часу Unix).

«1439694000000,2, transaction, 1000, 234» означає, що користувач = 2 придбав товар з ідентифікатором = 1000 у транзакції з ідентифікатором = 234 на 1439694000000 (мітка часу Unix)

Файл із властивостями предметів містить 275 902 рядки, тобто різні властивості, що описують 47 053 унікальних предметів. Оскільки властивість товару може змінюватися в часі (наприклад, ціна змінюється з часом), кожен рядок у файлі має відповідну позначку часу. Іншими словами, файл складається з об’єднаних знімків кожного тижня у файлі з даними про поведінку. Однак, якщо властивість предмета є постійним протягом спостеріганого періоду, у файлі буде присутнє лише одне значення моментального знімка.

Файл властивостей предмета містить стовпець із позначкою часу, оскільки всі вони залежать від часу, оскільки властивості можуть змінюватися з часом, наприклад ціна, категорія тощо.

Файл дерева категорій має 1669 рядків. Кожен рядок у файлі вказує дочірню категорію ідентифікатора та відповідного батька.

Візьмемо як приклад наступні рядки:

Рядок «100,200» означає, що ідентифікатор категорії = 1 має батьківський з ідентифікатором категорії = 200

Рядок «300» означає, що ідентифікатор категорії не має батьківського предмета у дереві

Для подальшої обробки даних їх потрібно зберегти у базі даних, аби кожного разу не відкривати файл, і не робити там змін, якщо вони потрібні.

Для цього виконаємо наступну команду: mongoimport --type csv -d recommendation -c products --headerline --drop products.csv

Тут mongoimport – це інструмент імпортує вміст із розширеного експорту JSON, CSV або TSV, створеного mongoexport, або, можливо, іншим стороннім інструментом експорту.

-type: Формат вводу для імпорту: json, csv або tsv. Ми використовуємо csv, тому саме це ми вказуємо.

-d: Вказує, яку базу даних використовувати. Ми використовували тестову базу даних.

-c: Вказує, яку колекцію використовувати. Ми використовували колекцію під назвою products.

-headerline: Вказує, що першим рядком у нашому файлі csv мають бути імена полів.

-drop: Вказує, що ми хочемо скинути колекцію перед імпортом документів.

Цю команду виконаємо для усіх файлів вибірки. Після чого нам необхідно виконати початкову обробку даних.

Як вже було описано раніше типи подій у вибірці записані як «view», «addtocart», «transaction», «remove», що не може використовуватися методами формування онлайн-рекомендацій, оскільки їм потрібні числові значення. Для цього, для кожного типу події оберемо числовий коефіцієнт зацікавленості користувача. Так для view поставимо 0.3, для addtocart - 0.6, transaction - 1, remove - -1. Для події повернення товару поставили негативне значення, оскільки користувач повернув товар, то за якоюсь причиною він йому не

сподобався, а це означає, що вірогідність того, що він захоче його купити наближається до нуля. Для цього напишемо, представлений на рисунку 4.1, скрипт для перетворення даних.

```
var allEvents = db.getCollection('events').find({})
var mapping = {
        'view': 0.3,
        'addtocart': 0.6,
        'transaction': 1,
        'remove': -1
    }

allEvents.forEach(function(event){
    if(!event.timestamp) {
        return;
    }
    var rand = Math.random();
    if(rand < 0.15) {
        event.event = 'remove';
    }

    db.getCollection('events').update(
        {_id: event._id},
        {
            StockCode: event.itemid,
            Quantity: mapping[event.event],
            CustomerID: event.visitorid
        }
    )
```

Рисунок 4.1 – Перетворення даних про події взаємодій користувачів з товарами

Таким чином отримаємо дані у вигляді, предстеваленому на рисунку 4.2.

(10) ObjectId("5fa810d7e90ec7ac2607bf65")	{ 4 fields }	Object
_id	ObjectId("5fa810d7e90ec7ac2607bf65")	ObjectId
StockCode	253185.0	Double
Quantity	0.3	Double
CustomerID	483717.0	Double

Рисунок 4.2 – Перетворення даних про події взаємодій користувачів з товарами

Тут StockCode – це ідентифіктор товару, Quantity - числова оцінка зацікавленості користувача у товарі, що раніше була типом події, а CustomerID – ідентифікатор користувача[41].

Тепер необхідно перетворити дані про продукти таким чином, щоб, коли буде використовуватися конента фільтрація непотрібно було їх обробляти заново. Скрипт для перетворення даних наведено на рисунку 4.3. Код представлений на ньому перетворить властивості предмету у один рядок.

```
db.getCollection('products').find({}).limit(100).forEach(function (el) {
    db.getCollection('products-mapped').insertOne(
        {
            itemid: el.itemId,
            description: JSON.stringify(el, 2, 'no-divider')
        }
    )
})
```

Рисунок 4.3 – Перетворення даних товари

Тепер ми можемо під’єднатися до бази даних MongoDB за допомогою драйверу представленному бібліотекою maven mongo-java-driver.

Для початку розберемо метод колаборативної фільтрації з неявним зворотнім зв’язком. Для цього методу нам необхідно отримати список усіх користувачів проводивших взаємодії з товарами а також список усіх товарів з якими ці зваємодії проводилися. Фрагмент коду, який виконує ці дії представлений на рисунку 4.4.

```
MongoClient mongoClient = new MongoClient( host: "localhost",  port: 27017);
MongoDatabase database = mongoClient.getDatabase( databaseName: "events");

MongoCollection events = database.getCollection( s: "events");
MongoCursor eventsIterator = events.find().iterator();
List<Long> customers = new ArrayList<Long>();
List<Long> products = new ArrayList<Long>();
while (eventsIterator.hasNext()) {
    Event eventObject = new Gson().fromJson(eventsIterator.next().toString(), Event.class);
    if(eventObject != null) {
        customers.add(eventObject.getCustomerId());
    }
    if(eventObject != null) {
        products.add(eventObject.getItemId());
    }
}
```

Рисунок 4.4 – Під’єднання до бази даних та отримання даних про користувачів та товари.

Замість того, щоб представляти явний рейтинг, тип події може представляти “впевненість” з точки зору того, наскільки сильною була взаємодія. Предмети, які були куплені. Дають більшу впевненість у вподобанні користувачем, ніж ті що лише були відкриті. З іншого боку, предмети, що були повернені, дають розуміння того, що користувачу вони не сподобалися. Далі для немодифікованої колаборативної фільтрації оцінки рівні або менші за нуль прибирають, як потенційно помилкові. У випадку з модифікованим методом, ми будемо використовувати їх так само, як і для оцінки користувачів у разі явного відгуку[42].

В якості наступного кроку нам потрібно створити розріджену матрицю рейтингів користувачів та предметів. Код представлений на рисунку 4.5 відображає цей процес для класичної колаборативної фільтрації, а код предстаклений на рисунку 4.6 – для модифікованого методу.

```
MongoCursor iter = eventsCollections.find().iterator();
List<Float> marks = new ArrayList<~>();
while (iter.hasNext()) {
    marks.add(new Gson().fromJson(eventsIterator.next().toString(), Event.class).getQuantity());
}

List<Event> events = Lists.newArrayList();
MongoCursor eventsIt = eventsCollections.find().iterator();
while (eventsIt.hasNext()) {
    events.add(new Gson().fromJson(eventsIterator.next().toString(), Event.class));
}
Iterable<Tuple3<Object, Object, Object>> eventsData = new LinkedList<>();
events.stream()
        .filter(event -> event.getQuantity() > 0)
        .map(event -> new Tuple3<>(event.getCustomerId(), event.getItemId(), event.getQuantity()))
        .forEach(eventsData::copyToArray);
SparseMatrix purchasesSparse = SparseMatrix.fromCOO(customers.size(), products.size(), eventsData);
```

Рисунок 4.5 – Створення розрідженої матриці рейтингів користувачів та предметів для немодифікованого методу

```
events.stream()
        .filter(event -> event.getQuantity() != 0)
        .map(event -> new Tuple3<>(event.getCustomerId(), event.getItemId(), event.getQuantity()))
        .forEach(eventsData::copyToArray);
SparseMatrix purchasesSparse = SparseMatrix.fromCOO(customers.size(), products.size(), eventsData);
```

Рисунок 4.6 – Створення розрідженої матриці рейтингів користувачів та предметів для модифікованого методу

Потім нам необхідно використати функцію, що прийме в оригінальній матриці предмета користувача та "замаскує" відсоток від початкових рейтингів, де відбулася взаємодія предмета користувача для використання в якості тестового набору. Тестовий набір міститиме всі оригінальні рейтинги, тоді як навчальний комплект замінює вказаний відсоток їх нулем у вихідній матриці рейтингів. Цей процес представлено на рисунку 4.7.

```
List<Tuple3<Long, Long, Float>> eventsData = events.stream()
        .filter(event -> event.getQuantity() != 0)
        .map(event -> new Tuple3<>(event.getCustomerId(), event.getItemId(), event.getQuantity()))
        .collect(Collectors.toList());
JavaRDD<Object> trainData = spark.read().json(eventsData.stream()
        .map(GSON::toJson)
        .toArray(String[]::new))
        .javaRDD()
        .map(Rating::parseRating);
```

Рисунок 4.7 – Фрагмент вихідного коду

Тепер нам потрібно перевірити, чи відповідає порядок рекомендацій для кожного користувача товарам, які вони придбали. Часто використовуваною метрикою для такого роду проблем є область під кривою робочої характеристики приймача (або ROC). Більша площа під кривою означає, що ми рекомендуємо товари, які в кінцевому підсумку купуються вгорі списку рекомендованих товарів. Зазвичай ця метрика використовується в більш типових задачах двійкової класифікації, щоб визначити, наскільки модель може передбачити позитивний приклад порівняно з негативним. Це також буде добре працювати для наших цілей ранжування рекомендацій.

Для цього нам потрібно написати функцію, яка може обчислити середню площу під кривою (AUC) для будь-якого користувача, який мав принаймні один маскований предмет. В якості еталону ми також підрахуємо, якою була б середня AUC, якби ми просто рекомендували найпопулярніші предмети. Популярність, як правило, важко перемогти у більшості системних проблем, що рекомендують, тому це дає гарне порівняння.

Для цього давайте створимо просту функцію, яка може обчислити нашу AUC. Фрагмент програмної реалізації цієї функції представлений на рисунку 4.8.

```
return positivePredictions.join(negativePredictions).values().mapToDouble(t -> {
    long correct = 0;
    long total = 0;
    for (Rating positive : t._1()) {
        for (Rating negative : t._2()) {
            if (positive.rating() > negative.rating()) {
                correct++;
            }
            total++;
        }
    }
    if (total == 0) {
        return 0.0;
    }
    return (double) correct / total;
}).mean();
```

Рисунок 4.8 – Фрагмент функції обчислення площі під кривою ROC.

Тепер ми можемо отримати числове значення площі під кривою ROC, і порівняти точність надання рекомендацій використовуючи класичну колаборативну фільтрацію та модифікований метод. Значення показників будуть наведені на рисунку 4.9 та рисунку 4.10.

```
AUC score. Classic collaborative filtering: 0.814

Process finished with exit code 0
```

Рисунок 4.9 – Показник AUC для немодифікованого методу колаборативної фільтрації.

```
AUC score. Modified collaborative filtering: 0.857

Process finished with exit code 0
```

Рисунок 4.10 – Показник AUC для модифікованого методу колаборативної фільтрації.

Тепер ми можемо заміряти час формування-онлайн рекомендацій подивитися на результати. На рисунку 4.11 на першому блоці представлена історія взаємодій користувача з предметами, а на другому рекомендовані товари для нього.

```
User purchase history
    StockCode Action                          Description
    22906     transaction          12 Indoor/Outdoor Rug Assorted 6-ft x 8-ft
    13258     transaction          CANVAS Darby Outdoor Rug, 8 x 10-ft
    30466     view                 PDG Storage Deck Box, 99-gal

Recommendations
  StockCode                            Description
   23133       CANVAS Quince Outdoor Rug 5 x 7-ft
   22862             CANVAS Outdoor Rug 5 x 7-ft
   90204   CANVAS Tribune Outdoor Rug 5-ft x 7-ft
   90114                       SUMMER DAISIES BAG
   22855                        FINE WICKER HEART
--- Execution time 7.65806865692139 seconds ---
```

Рисунок 4.11 – Рекомендації для користувача сформовані за методом колаборативної фільтрації

Тепер нам необхідно знаходити рекомендації засновані на контентній фільтрації. Для цього нам потрібно обчислити важливості особливостей предметів за допомогою алгоритму TF-IDF, який оцінює важливість слова в контексті документа(в даному випадку строки, яка є репрезентацією предмету),

що є частиною колекції документі. Вага деякого слова пропорційна частоті вживання цього слова в документі і обернено пропорційна частоті вживання слова в усіх документах колекції. Після цього на підставі косинусів подібності предметів ми будемо надавати рекомендації. Фрагмент логіки цього алгоритму буде представлений на рисунку 4.12. Приклад таких рекомендацій наведено на рисунку 4.13.

```
public double getTF(String docName, String term) {
    term = term.toLowerCase();
    final Map<String, AtomicInteger> doc = this.getDoc(docName);
    if(doc != null) {
        double total = 0.0;
        if(isPhrase(term)) {
            for (String key : doc.keySet()) {
                if (key.equals(term)) {
                    total += doc.get(key).get();
                } else if (key.contains(term)) {
                    total += doc.get(key).get() / 2.0;
                }
            }
        } else if(doc.containsKey(term)) {
            total = doc.get(term).get();
        }

        return total / Double.valueOf(doc.size());

    }

    return 0.0;
}

public double getIDF(String term) {
    term = term.toLowerCase();
    final int numberOfDocsWithTerm = this.numberOfDocsWithTerm(term);
    if(numberOfDocsWithTerm != 0) {
        return Math.log(Double.valueOf(this.corpusBow.size()) / Double.valueOf(numberOfDocsWithTerm));
    } else {
        return 0;
    }
}

public double getTFIDF(String docName, String term) {
    return getTF(docName, term) * getIDF(term);
}
```

Рисунок 4.12 – Оцінка важливості особливостей предметів

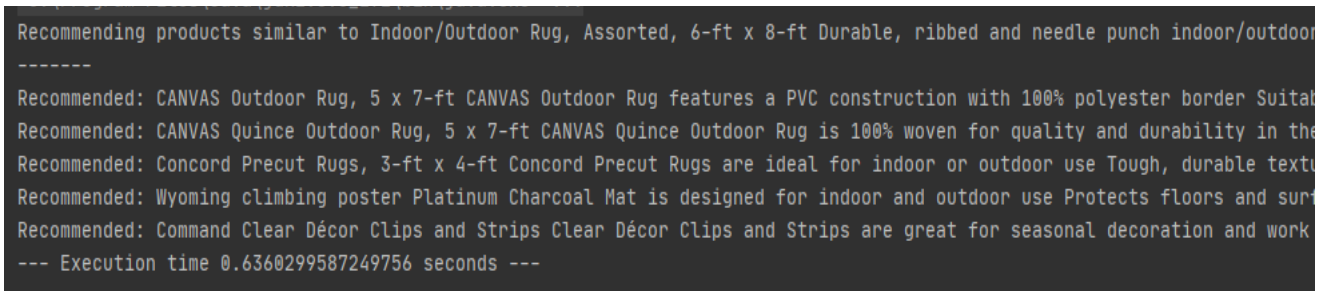

Рисунок 4.13 – Рекомендаціїї складені шляхом контентної фільтрації

4.2 Експериментальна перевірка удосконаленого методу

Тепер, коли ми отримали показник AUC для методів колаборативної фільтрації, а також отпримали результати рекомендацій як для колаборативної фільтрації, так і для контентої, ми можемо порівняти результати експерименту для різних методів.

Таблиця 4.2 Порівняння різних методів формування онлайн-рекомендацій

Метод	Значення AUC-показника	Час формування рекомендацій у секундах
Колаборативна фільтрація з неявним зворотнім зв’язком	0,814	7,03
Модифікована колаборативна фільтрація	0,857	7,65
Контентна фільтрація	Не обчислюється	0,6

На підставі результатів експерименту можна зробити висновок, що використання негативного відгуку у системах формування онлайн-рекомендацій, заснованих на методі колаборативної фільтрації з неаявним зворотнім зв’язком є доцільним, оскільки їх точність надання тих самих рекомендацій дещо підвищується, що можно побачити у зрості значення AUC-показника на 0,043, тобто десь чотири відсотки. Збільшена точність таких рекомендацій можуть позитивно вплинути на продажі систем електронної комерції а заразом із цим на прибуток для компаній, що їх використовують.

Також, можемо помітити, що час формування рекомендацій за використання модифікованого методу дещо зростає, що обумовлено більшим об’ємом даних, що система повинна обробити.

Але, час за який формуються рекомендації за колаборативної фільтрації все одно достатньо великий, як для систем, що використовуються для інтернет магазинів, тому у випадках, коли профіль користувача заповнений недостатньо, точність рекомендацій може бути достатньо низькою, оскільки, користувачі, що також взаємодіяли з системою електронної коммерції схожим чином, у реальному житті можуть мати зовсім інші вподобання, і тому рекомендації для таких користувачів будуть неактуальними. Із цього витікає, що їх формування за такий достатньо великий час може бути недоцільним.

Звісно у такому випадку точність формування рекомендацій дуже залежить від схожості конкретних товарів, з якими взаємодіє користувач. За час проведення експерименту схожість товарів, представлена косинусом кута подібності для найбільш подібних товарів варіювалася від 0.15(представлено на рисунку 4.14) до 0.7 (представлено на рисунку 4.15)

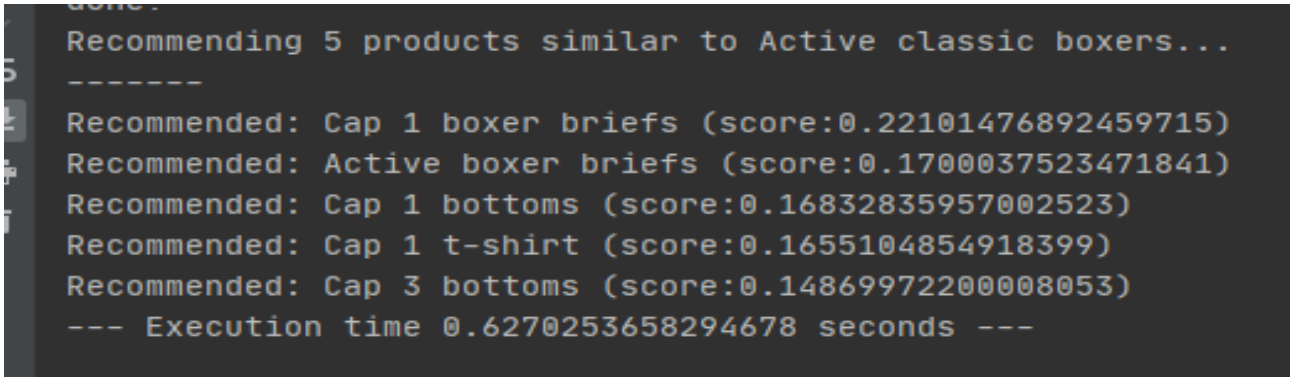

Рисунок 4.14 – Найгірші знайдені показники для рекомендацій, сформованих методом контентної фільтрації

```
Recommending 5 products similar to Indoor/Outdoor Rug, Assorted, 6-ft x 8-ft..
-------
Recommended: CANVAS Outdoor Rug, 5 x 7-ft (score:0.7391296832588594)
Recommended: CANVAS Quince Outdoor Rug, 5 x 7-ft (score:0.6613950201190658)
Recommended: Concord Precut Rugs, 3-ft x 4-ft (score:0.628414746146246)
Recommended: Wyoming climbing poster (score:0.5209471336717464)
Recommended: Lead an examined life poster (score:0.5203360575018386)
--- Execution time 0.7855191230773926 seconds ---
```

Рисунок 4.15 – Найкращі знайдені показники для рекомендацій, сформованих методом контентної фільтрації

У таких випадках є сенс використовувати контенту фільтрацію, оскільки час формування рекомендацій, використовуючи цей метод швидше більш ніж у дванадять з половиною разів.

ВИСНОВКИ

Магістерська робота присвячена вивченню проблем підвищення точності рекомендацій з урахуванням відмов від товарів а також вибору схожих користувачів та схожих товарів

В результаті виконання магістерської роботи були досліджені методи формування онлайн-рекомендацій у системах електронної комерції.

Були проаналізовані існуючі методи формування онлайн-рекомендацій, а саме: метод колаборитвної фільтрації, заснований на неявному відгуці, а також метод контентної фільтрації, заснований на алгоритмі TF-IDF та косинусах подібності.

Була розглянута проблема колаборативної фільтрації з використанням негативного неявного відгуку, а також заміщення її контентною фільтрацією у разі, коли поведінковий профіль користувача заповнений недостатньо для точного формування рекомендацій. Це дозволило збільшити точність надання рекомендацій для користувачів, у яких такий профіль заповнений достатньо повно, а також знизити час формування рекомендацій, коли даних профілю недостатньо.

В практичному аспекті метод дозволяє покращити показник AUC від 0,814 до 0,857, тобто трохи більше ніж на 4 %, а у випадку коли профіль користувача заповнений недостатньо повно надавати рекомендації використовуючи контентну фільтрацію.

Результати дослідження по магістерській роботі представлено на міжнародному молодіжного форуму «Радіоелектроніка та молодь у XXI столітті» 2020р. опубліковані тези доповіді «Дослідження методів формування онлайн-рекомендацій у системах електронної комерції».

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Методичні вказівки щодо розробки та оформлення магістерської атестаційної роботи за спеціальністю 8.05010101 – Інформаційні управляючі системи та технології. Освітньо-кваліфікаційний рівень – магістр / Упоряд.: Левикін В.М., Міхнов Д.К., Саєнко В.І., Євланов М.В., Міхнова А.В., Керносов М.А. – Харків: ХНУРЕ, 2012. – 28 С.

2. G. Adomavicius and A. Tuzhilin, “Towards the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions”, IEEE Transactions on Knowledge and Data Engineering 17, 2005. С. - 634–749.

3. R. Bell and Y. Koren, “Scalable Collaborative Filtering with Jointly Derived Neighborhood Interpolation Weights”, IEEE International Conference on Data Mining (ICDM’07) , 2007. С. - 43–52.

4. R. Bell, Y. Koren and C. Volinsky, “Modeling Relationships at Multiple Scales to Improve Accuracy of Large Recommender Systems”, Proc. 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2007.

5. J. Bennet and S. Lanning, “The Netflix Prize”, KDD Cup and Workshop, 2007. www.netflixprize. com.

6. D. Blei, A. Ng, and M. Jordan, “Latent Dirichlet Allocation”, Journal of Machine Learning Research 3, 2003. С. - 993–1022.

7. M. Deshpande, G. Karypis, “Item-based top-N recommendation algorithms”, ACM Trans. Inf. Syst. 22, 2004. С. - 143-177.

8. S. Funk, “Netflix Update: Try This At Home”, http://sifter.org/˜simon/journal/20061211.html, 2006.

9. D. Goldberg, D. Nichols, B. M. Oki and D. Terry, “Using Collaborative Filtering to Weave an Information Tapestry”, Communications of the ACM 35, 1992. С. - 61– 70.

10. J. L. Herlocker, J. A. Konstan, A. Borchers and John Riedl, “An Algorithmic Framework for Performing Collaborative Filtering”, Proc. 22nd ACM SIGIR Conference on Information Retrieval, 1999. С. - 230–237.

11. J. L. Herlocker, J. A. Konstan, and J. Riedl. “Explaining collaborative filtering recommendations”, In Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work, ACM Press, 2000. C. - 241-250

12. T. Hofmann, “Latent Semantic Models for Collaborative Filtering”, ACM Transactions on Information Systems 22, 2004. C. - 89–115.

13. Z. Huang, D. Zeng and H. Chen, “A Comparison of Collaborative-Filtering Recommendation Algorithms for E-commerce”, IEEE Intelligent Systems 22, 2007. C. - 68–78.

14. G. Linden, B. Smith and J. York, “Amazon.com Recommendations: Item-to-item Collaborative Filtering”, IEEE Internet Computing 7, 2003. C. - 76–80.

15. D.W. Oard and J. Kim, “Implicit Feedback for Recommender Systems”, Proc. 5th DELOS Workshop on Filtering and Collaborative Filtering, 1998. C. - 31–36.

16. Paterek, “Improving Regularized Singular Value Decomposition for Collaborative Filtering”, Proc. KDD Cup and Workshop, 2007.

17. R. Salakhutdinov, A. Mnih and G. Hinton, “Restricted Boltzmann Machines for Collaborative Filtering”, Proc. 24th Annual International Conference on Machine Learning, 2007. C. - 791–798.

18. R. Salakhutdinov and A. Mnih, “Probabilistic Matrix Factorization”, Advances in Neural Information Processing Systems 20 (NIPS’07) , 2008. 1257–1264.

19. B. M. Sarwar, G. Karypis, J. A. Konstan, and J. Riedl, “Application of Dimensionality Reduction in Recommender System – A Case Study”, WEBKDD’2000.

20. B. Sarwar, G. Karypis, J. Konstan and J. Riedl, “Itembased Collaborative Filtering Recommendation Algorithms”, Proc. 10th International Conference on the World Wide Web, 2001. C. - 285-295.

21. G. Takacs, I. Pilaszy, B. Nemeth and D. Tikk, “Major Components of the Gravity Recommendation System, 2008.

22. Y. F. Hu, Y. Koren, C. Volinsky, Collaborative Filtering for Implicit Feedback Datasets, IEEE International Conference on Data Mining (ICDM 2008), IEEE, Winner of the 2017 IEEE ICDM 10-Year Highest-Impact Paper Award, 2008. 10c.

23. V. Singh, B. Kumar, and T. Patnaik, “Feature Extraction Techniques for Handwritten Text in Various Scripts: a Survey,” International Journal of Soft Computing & Engineering, vol. 3, 2013. C. - 238-241

24. Чалий С.Ф., Прібильнова І.Б. Ситуаційне представлення споживачів рекомендаційної системи. Матеріали дев’ятої міжнародної науково-технічної конференції. С.35.

25. X. Chen, S. F. Li, and Y. F. Wang, “The feature extraction of the text based on the deep learning,” Advanced Science and Industry Research Center. Proceedings of the 2014 International Conference on Network Security and Communication Engineering (NSCE 2014), 2014, C. - 5.

26. Chalyi S., Leshchynskyi V., Leshchynska I. Багаторівнева персоналізація пояснень в рекомендаційних системах. Сучасні інформаційні системи, 2020.Том 4, № 2. С. 170-175.

27. Y. Tian, J. Zhang, “Improvement of Linked Data Fusion Algorithm Based on Bag of Words,” Library Journal, vol. 35, 2016. C. - 17-22.

28. W. Zhang, T. Yoshida, and X. Tang, “A comparative study of TF*IDF, LSI and multi-words for text classification,” Expert Systems with Applications, vol. 38, 2010. C. - 2758-2765.

29. M. Y. Jiang, R. Liu, and F. Wang, “Word Network Topic Model Based on Word2Vector,” IEEE Explore, C. - 241-247.

30. Y. Kim, et al, “DBCURE-MR: An efficient density-based clustering algorithm for large data using MapReduce,” Information Systems, vol. 42, 2013. C. - 162-166.

31. S. Q. Xue, and Y. J. Niu, “Research on Chinese text similarity based on vector space model,” Electronic Design Engineering, 2016.

32. L. H. Patil, and M. Atique, “A novel feature selection based on information gain using WordNet,” Science and Information Conference IEEE, 2013. С. - 625-629.

33. Chalyi S. Доповнення вхідних даних рекомендаційної системи в ситуації циклічного холодного старту з використанням темпоральних обмежень типу «next» / S. Chalyi, V. Leshchynskyi, I. Leshchynska // Системи управління, навігації та зв’язку. Збірник наукових праць. – Полтава: ПНТУ, 2019. – Т. 4 (56). – С. 105-109. – doi:https://doi.org/10.26906/SUNZ.2019.4.105.

34. Чалий С.Ф., Лещинський В.О., Лещинська І.О. Доповнення вхідних даних рекомендаційної системи в ситуації циклічного холодного старту з використанням темпоральних обмежень типу «NEXT». Системи управління, навігації та зв'язку, 2019. Вип. 4(56). С. 105-109.

35. Y. Lu, and M. Liang, “Improvement of Text Feature Extraction with Genetic Algorithm,” New Technology of Library & Information Service, 2014. С. - 523–525.

36. L. H. Wang, “An Improved Method of Short Text Feature Extraction Based on Words Co-Occurrence,” Applied Mechanics & Materials, vol. , 2014. С. - 842-845.

37. H. Liang, et al, “Text feature extraction based on deep learning: a review:,” Eurasip Journal on Wireless Communications & Networking, vol. 2017, 2017. С. - 211.

38. S. Zhou , et al, “Characteristic representation method of document based on Word2vector,” Journal of Chongqing University of Posts & Telecommunications, vol. 30, 2018. С. - 272-279.

39. Y. Chen, et al, “A fast clustering algorithm based on pruning unnecessary distance computations in DBSCAN for high-dimensional data,” Pattern Recognition, 2018. С. - 83.

40. S. Tan, “Neighbor-weighted K-nearest neighbor for unbalanced text corpus,” Expert Systems with Applications. vol. 28, 2005. С. - 667-671

41. T. Denoeux, “A k-nearest neighbor classification rule based on dempster-shafer theory” IEEE Transactions on Systems Man and Cybernetics vol. 25 no. 5, 1995. C. - 804-813.

42. He Shaojun, et al, “The Capability Analysis on the Characteristic Selection Algorithm of Text Categorization Based on F1 Measure Value,” IEEE Explore, C. - 742 -746.