Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Штучного інтелекту
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Дослідження та розробка знання-орієнтованих методів побудови персоналізованого переліку товарів та послуг в рекомендаційних системах
(тема)

Виконав:
студент 2 курсу, групи СШІм-21-2
Чмутов Д.С.
(прізвище, ініціали)

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

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

Освітня програма Системи штучного інтелекту
(повна назва спеціалізації)

Керівник проф. Чалий С.Ф.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ В.О. Філатов
(підпис) (прізвище, ініціали)

2023 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Штучного інтелекту
(повна назва)

Рівень вищої освіти другий (магістерський)

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

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

Освітня програма Системи штучного інтелекту (СШІ)
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«_____»_____20____р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Чмутову Данилу Станіславовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження та розробка знання-орієнтованих методів побудови персоналізованого переліку товарів та послуг в рекомендаційних системах

затверджена наказом університету від 31 березня 20 23 р. № 306Ст

2. Термін подання студентом роботи до екзаменаційної комісії 19 травня 20 23 р.

3. Вихідні дані до роботи Операційна система – Windows 10, Частота процессору 2.5 ГГц, Аналітика, Метод, Системи Рішення, HTML, WEB-APP, WEB

4. Перелік питань, що потрібно опрацювати в роботі

1) Детальний аналіз задач побудови

2) Постановку задач дослідження, формалізація

3) Аналіз алгоритму та систем

4) Виконання практичної реалізації

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри)_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

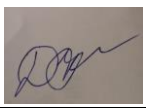
6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

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

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Аналіз предметної області	05.04.2023	Виконано
2	Аналіз методів і підходів	10.04.2023	Виконано
3	Аналіз ефективності методів	20.04.2023	Виконано
4	Підготовка пояснювальної записки	01.05.2023	Виконано
5	Підготовка презентації та доповіді	02.05.2023	Виконано
6	Нормоконтроль, рецензування	13.05.2023	Виконано
7	Занесення диплома в електронний архів	30.05.2023	Виконано
8	Написання пояснювальної записки	07.05.2023	Виконано
9	Захист перед ЕК	19.05.2023	

Дата видачі завдання 3 квітня 2023 р.

Студент  _____

(підпис)

Керівник роботи _____ проф. Чалий С.Ф.

(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 77 с., 2 табл., 19 рис., 13 форм., 1 дод., 35 джерел.

ANALYTICS, DECISION-SYSTEMS, HTML, METHOD, OBJECT, PROJECT, RECOMMENDATION, SYSTEM, WEB-APP, WEB.

Об’єкт дослідження – рекомендаційні системи.

Предмет дослідження – знання-орієнтовані методи побудови персоналізованого переліку товарів та послуг.

Мета та завдання – дослідження та розробка знання-орієнтованих методів побудови персоналізованого переліку товарів та послуг в рекомендаційних системах. Для досягнення поставленої мети у роботі необхідно виконати низку завдань.

Методи дослідження – для вирішення поставлених завдань використовувалися як загальнонаукові, так і спеціальні методи наукового пізнання. Системний аналіз, дедуктивний та індуктивний методи використовувалися при визначенні суті понять «Рекомендаційна система», «Персоналізований перелік». Також використовувались основні положення теорії ймовірностей та математичної статистики, методи математичного моделювання, нечіткої логіки, теорії множин.

Інформаційну базу дослідження складають чинні вітчизняні законодавчі та нормативно-правові акти, документація сучасних розробок у сфері інформатизації та цифровізації.

Методологічну основу методів дослідження складає системнийпідхід при вирішенні проблеми забезпечення високого рівня якостіобробки даних.

ABSTRACT

Explanatory note: 77 p., 19 fig., 2 tabl., 13 form., 1 ann., 35 sources.

ANALYTICS, DECISION-SYSTEMS, HTML, METHOD, OBJECT, PROJECT, RECOMMENDATION, SYSTEM, WEB-APP, WEB.

The object of research is recommender systems.

Subject of research – knowledge-based methods for building a personalised list of goods and services.

The aim and objectives are to research and develop knowledge-based methods for building a personalised list of goods and services in recommender systems. To achieve this goal, a number of tasks need to be performed.

Research methods – both general scientific and special methods of scientific cognition were used to solve the tasks. System analysis, deductive and inductive methods were used to define the essence of the concepts of «Recommendation system» and «Personalised list». The main provisions of probability theory and mathematical statistics, methods of mathematical modelling, fuzzy logic, and set theory were also used.

The information base of the study includes current domestic legislative and regulatory acts, documentation of modern developments in the field of informatisation and digitalisation.

The methodological basis of the research methods is a systematic approach to solving the problem of ensuring a high level of data processing quality.

ЗМІСТ

ВСТУП

На сьогоднішній день існує безліч сайтів, що надають будь-який контент, наприклад новини, блоги, музика та кіно. Кожен з них містить величезну кількість інформації, але не вся вона може виявитися цікавою для конкретного відвідувача сайту. Для підбору контенту, який буде корисним для певного користувача, використовуються рекомендаційні системи. На відміну від пошукових систем, щоб отримати відповідь, рекомендаційна система не потребує чіткого запиту. Користувачеві пропонується оцінити деякі об'єкти з колекції і на підставі його оцінок будуються припущення та повертаються найближчі до них результати. Рекомендаційні системи дуже затребувані нині, оскільки значно зменшують час знаходження корисної інформації.

За останні кілька десятиліть, з появою Youtube, Amazon, Netflix та багатьох інших подібних веб-сервісів, системи рекомендацій стали займати дедалі більше місця у житті сучасної людини. Починаючи з електронної комерції (пропонуючи покупцям статті, які можуть їх зацікавити) і закінчуючи рекламою в Інтернеті (пропонуючи користувачам правильний зміст, що відповідає їх перевагам), рекомендаційні системи сьогодні є актуальними в щоденних онлайн-подорожах.

Загалом, рекомендаційні системи – це алгоритми, призначені для пропозиції користувачам відповідних предметів (предметів, фільмів, тексту для читання, продуктів для покупки або чогось іншого, залежно від галузі).

Рекомендаційні системи дійсно важливі в деяких галузях, оскільки вони можуть приносити величезний прибуток, коли вони ефективні, або можуть значно відрізнятися від конкурентів.

1 РЕКОМЕНДАЦІЙНІ СИСТЕМИ ТА АЛГОРИТМИ ПІДБОРУ

1.1 Аналіз задач побудови рекомендацій

Термін рекомендацій система відноситься до всіх програмних інструментів і методик, які, використовуючи знання, про користувачів та предмети, дають рекомендації щодо елементів, які найімовірніше становлять інтерес для конкретного користувача. Рекомендації можуть бути пов'язані з різними процесами прийняття рішень, наприклад, які продукти купувати, яку слухати музику або які фільми дивитися. У цьому контексті елемент – це загальний термін, який використовується визначення того, що система рекомендує користувачам. Системарекомендацій зазвичай орієнтована на певний тип або клас предметів, таких як книги для покупки, статті для читання новин або готелі для бронювання [1].

Рекомендаційні системи становлять одну з основних частин екосистеми електронної комерції, що дозволяє користувачам відсіювати великі обсяги інформації та продуктів. Дослідження у цій галузі дозволяють зрозуміти, як технологія рекомендацій може застосовуватися уконкретних галузях. Наявність безлічі дослідницьких робіт, що демонструють різні алгоритми рекомендацій та з поєднанням, проведення конференцій, вказують на те, що надання якісної рекомендації є актуальною проблемою [2].

Хоча головна мета рекомендацій для електронної комерції – допомогти компаніям продавати більше товарів, вони також мають багато переваг з погляду користувача. Користувачі постійно перевантажені вибором: які новини читати, які продукти купувати, які дивитись і т.д. Зі збільшенням обсягу каталогу товарів користувачі більше не можуть переглядати всі з них. У цьому сенсі механізми рекомендацій надають «індивідуальний» досвід, допомагаючи людям швидше знаходити те, що

вони шукають чи те що може бути цікавим. Кінцевим результатом є те, що задоволення користувачів буде вищим, тому що вони отримають релевантні результати в більш короткі терміни.

Способи аналізу даних бувають трьох видів (рисунок 1.1).

Рисунок 1.1 – Способи аналізу даних

Системи рекомендацій розрізняються за способом, яким вони аналізують дані, щоб виявити відносини між користувачами тапредметами. Більшість систем рекомендацій використовують три основні підходи [3]:

– контентна фільтрація;
– колаборативна фільтрація;
– на основі знань;
– гібридний підхід.

Існують різні причини, через які все більше постачальників послуг використовують цей набір інструментів і методів, у тому числі:

– збільшення кількості проданих товарів. Це одна з найважливіших функцій: допомогти бізнесу продати більше товарів. Через величезні каталоги товарів доступних у більшості магазинів, користувачі часто не можуть знайти те, що хочуть;

– підвищення задоволеності користувачів. Правильно спроектований інтерфейс рекомендацій покращує взаємодію користувача з програмою. Поєднання ефективних, точних рекомендацій та зручного інтерфейсу збільшує суб'єктивну оцінку системи користувачем;

– найкраще розуміння того, що хоче користувач. Рекомендаційна система створює модель переваг користувача. яка може повторно використовувати бізнес для низки інших цілей, таких як поліпшення рекламних кампаній або оптимізація управління запасами.

За деякими з перерахованих вище причин, графічне представлення даних та аналіз на основі графів може полегшити завдання, спрощуючи управління даними, аналіз даних та обмін даними.

1.2 Аналіз методів побудови рекомендацій

Колаборативна фільтрація є альтернативою контентної фільтрації і спирається тільки на поведінку користувачів у минулому, наприклад, на попередні покупки або рейтинги елементів, або на думку існуючої спільноти користувачів, що дозволяє передбачити, які елементи користувачам найбільше сподобаються або будуть їм цікаві, не вимагаючи створення явного профілю для елементів та користувачів [4].

Основна ідея полягає в припущенні, що якщо користувачі поділяли одні й ті ж інтереси в минулому – наприклад, якщо вони переглядали або купували одні й ті самі книги – у них у майбутньому також будуть однакові уподобання. Таким чином, якщо, наприклад, користувач А такористувач В мають історію покупок, яка сильно перетинається, і

користувач А недавно купив товар 1, яку користувач В ще не бачив, то цілком можливо, що цей товар йому також сподобається [5].

Методи спільної фільтрації, як правило, поділяються на два основні підходи:

– на основі сусідства (memory – based);
– на основі моделі (model – based).

Методи, засновані на пам'яті, видають рекомендації для користувача, ґрунтуючись на обчисленні міри схожості за всіма даними, отриманими в ході аналізу. Цей метод можна розділити на два підтипи:

– на основі схожості користувачів (user – based);
– на основі схожості елементів (item – based).

Підхід на основі схожості користувачів є найчастіше використовуваним у колаборативній фільтрації. Щоб передбачити, які елементи повинні відображатися або рекомендуватися користувачеві, система покладається на аналіз сусідства цього конкретного користувача. Найближчі сусіди обчислюються з урахуванням минулих взаємодій. До них включаються користувачі, які вчинили аналогічні дії для інших предметів. Потім кожного предмета, який поточний користувач А ще бачив, обчислюється прогноз з урахуванням оцінок р, зроблених найближчими сусідами [5].

Продуктивність цього підходу страждає при розріджених даних. При великих наборах елементів та невеликій кількості оцінок або покупок виникають моменти, коли не можна дати рекомендації для користувача, який не має загальних оцінок з іншими користувачами. Одним із рішень проблеми з розрідженістю даних є використання графа для відображення дій користувачів [6].

На відміну від розглянутого алгоритму спільної фільтрації, заснованого на користувачах, підхід на основі елементів розглядає набір елементів, оцінюваних цільовим користувачем А, і обчислює, наскільки

вони схожі на цільовий елемент 1, а потім вибирає k найбільш схожих елементів.

Друга група алгоритмів – це підхід на основі моделі. Створюються «моделі» для користувачів та елементів, які описують їхню поведінку за допомогою набору «факторів» або «функцій» та ваги. Для користувачів кожен фактор висловлює те, наскільки користувачеві подобаються елементи з високою оцінкою відповідного фактора. У цих методах початкові дані (набір даних користувач-елемент) спочатку обробляються в автономному режимі, інформація про рейтинги або попередні покупки використовується для створення цієї прогностичної моделі. Моделі прихованого фактора є найбільш поширеними підходами в цьому класі. Вони намагаються пояснити рейтинги, характеризуючи як предмети, так і користувачів [7].

Алгоритми у цій категорії використовують ймовірнісний підхід і розглядають процес спільної фільтрації як обчислення очікуваної оцінки користувача, враховуючи його оцінки з інших елементів. Процес побудови моделі виконується за допомогою різних алгоритмів машинного навчання, таких як байєсовська мережа, кластеризація та підходи на основі правил. Модель байєсівської мережі формулює ймовірну модель для проблеми спільної фільтрації. Модель кластеризації розглядає спільну фільтрацію як проблему класифікації [6] і працює шляхом кластеризації схожих користувачів в одному класі та оцінки ймовірності того, що конкретний користувач знаходиться у певному класі C, і звідти обчислює умовну ймовірність рейтингів. Підхід, що ґрунтується на правилах, застосовує алгоритми виявлення правил асоціації, щоб знайти зв'язок між купленими предметами, а потім генерує рекомендації з предметів на основі сили зв'язку між предметами [7].

Нещодавні дослідження показують, що сучасні підходи, засновані на моделях, перевершують алгоритми на основі пам'яті при прогнозуванні рейтингів [3], [4], однак також є роботи, де зроблені висновки з приводу

того, що одна тільки точність прогнозу не гарантує користувачам ефективний і задовільний досвід [5].

Як було вже вище сказано, одні з основних причин, з яких компанії впроваджують рекомендаційні системи, полягають у підвищенні задоволеності та лояльності користувачів та продажу різноманітнихтоварів. Рекомендація товару є великою цінністю, якщо користувач не знавпро цей товар, але, швидше за все, виявив би цей товар сам [5]. Цейприклад показує її. Ще один важливий фактор, який відіграє важливу роль в оцінці користувачів системи рекомендацій: це випадковість [8]. З іншого боку, підходи на основі сусідства фіксують локальні асоціації даних. Отже, система рекомендації, заснована на такому підході, може рекомендувати користувачеві елементи, що сильно відрізняються від його звичайного смаку, або елементи, які не дуже відомі, якщо його найближчі сусіди (користувачі) дали йому великий рейтинг.

1.3 Аналіз методів побудови рекомендацій на основі знань

Методи, що ґрунтуються на знаннях, вимагають від користувача описати свої вимоги до потрібних йому об'єктів. А потім шукають з використанням своєї бази знань об'єкти, які відповідають вимогам.

Рекомендаційні системи, засновані на знаннях, добре підходять для рекомендацій товарів, які купуються нерегулярно. Крім того, у таких доменах товарів користувачі, як правило, більш активно висловлюють свої вимоги. Користувач часто може бути готовий прийняти рекомендацію щодо фільму без особливого введення, але вона не захоче прийняти рекомендації щодо будинку чи автомобіля, не маючи детальної інформації про конкретні характеристики предмета. Тому рекомендовані системи, засновані на знаннях, підходять до типів предметних доменів, відмінних від спільних систем і систем, що базуються на вмісті. Загалом системи рекомендацій, засновані на знаннях, підходять у таких ситуаціях:

1. Клієнти хочуть чітко вказати свої вимоги. Тому інтерактивність є найважливішою складовою таких систем. Зауважте, що спільні системи та системи на

основі вмісту не дозволяють цей тип детального зворотного зв'язку.

2. Важко отримати оцінки для конкретного типу товару через більшу складність домену продукту з точки зору типів товарів і доступних опцій.

3. У деяких доменах, таких як комп'ютери, оцінки можуть залежати від часу. Оцінки старого автомобіля чи комп'ютера не дуже корисні для рекомендацій, оскільки вони змінюються зі зміною доступності продукту та відповідними вимогами користувачів[10].

Вирішальною частиною систем, заснованих на знаннях, є більший контроль, який користувач має під час керування процесом рекомендацій. Цей більший контроль є прямим результатом необхідності мати можливість визначити детальні вимоги в складній за своєю суттю проблемній області. На базовому рівні концептуальні відмінності в трьох категоріях рекомендацій описані на (рисунок 1.2).

Approach	Conceptual Goal	Input
Collaborative	Give me recommendations based on a collaborative approach that leverages the ratings and actions of my peers/myself.	User ratings + community ratings
Content-based	Give me recommendations based on the content (attributes) I have favored in my past ratings and actions.	User ratings + item attributes
Knowledge-based	Give me recommendations based on my explicit specification of the kind of content (attributes) I want.	User specification + item attributes + domain knowledge

Рисунок 1.2: Концептуальні цілі різних систем рекомендацій

Зверніть увагу, що існують також значні відмінності у вхідних даних, які використовуються різними системами. Рекомендації щодо систем, що базуються на контенті, і систем для спільної роботи в основному базуються на історичних даних, тоді як системи, що базуються на знаннях, базуються на прямих специфікаціях користувачам те, що вони хочуть. Важливою відмінною характеристикою систем, заснованих на знаннях, є високий рівень адаптації до конкретної області [12]. Це налаштування досягається за допомогою використання бази знань, яка кодує відповідні знання предметної області у формі обмежень або показників подібності. Деякі системи, засновані на знаннях, також можуть використовувати user атрибути (наприклад, демографічні атрибути) на додаток до атрибутів елемента, які вказуються під час

запиту. У таких випадках знання домену можуть також кодувати зв’язки між атрибутами користувача та атрибутами елемента. Однак використання таких атрибутів не є універсальним для систем, заснованих на знаннях, у яких більша увага приділяється вимогам користувача[12].

Рекомендаційні системи на основі знань можна класифікувати на основі інтерактивної методології користувача та відповідних баз знань, які використовуються для полегшення взаємодії. Існує два основних типи систем рекомендацій на основі знань:

1. Системи рекомендацій на основі обмежень: у системах на основі обмежень користувачі зазвичай вказують вимоги або обмеження (наприклад, нижні або верхні межі) щодо атрибутів елемента. Крім того, специфічні для домену правила використовуються для зіставлення вимог або атрибутів користувача з атрибутами елемента. Ці правила представляють предметно-специфічні знання використовування системою [14].

2. Рекомендаційні системи на основі прецендентів: у системах рекомендацій на основі випадків конкретні випадки вказуються користувачем як цілі або опорні точки. Показники подібності визначаються в атрибутах елемента для отримання подібних елементів до цих цілей [14]. Показники подібності часто ретельно визначаються в доменно-специфічний спосіб. Таким чином, метрики подібності формують знання предметної області, які використовуються в таких системах. Зауважте, що в обох випадках система надає користувачеві можливість змінити вказані вимоги. Однак спосіб, у який це робиться, у двох випадках відрізняється. У системах, заснованих на регістрах, приклади (або випадки) використовуються як опорні точки для керівництва пошуком у поєднанні з показниками подібності, тоді як у системах, заснованих на обмеженнях, для керівництва пошуком використовуються конкретні критерії/правила (або обмеження). В обох випадках представлені результати використовуються для зміни критеріїв для пошуку подальших рекомендацій.

1.4 Постановка задачі дослідження

Метою даної роботи є дослідження та розробка знання-орієнтованих методів побудови персоналізованого переліку товарів та послуг в

рекомендаційних системах.

Для досягнення поставленої мети у роботі необхідно:

– розкрити ситуацію холодного старту та графові бази даних;

– дослідити принципи побудови персоналізованого переліку товарів та послуг;

– визначити вимоги до системи;

– проаналізувати алгоритм побудови персоналізованого переліку товарів та послуг в рекомендаційних системах;

– розробити рекомендаційну систему персоналізованого переліку товарів та послуг.

1.5 Висновки до розділу

У рамках першого розділу розглянуто поняття рекомендаційної системи, здійснено загальну постановку завдання, описано не персональні рекомендації та класичні підходи (content-based і колаборативну фільтрацію), а також окреслено тему обґрунтування рекомендацій. У цілому нині двох цих підходів цілком достатньо побудови production-ready рекомендаційної системи.

2 УДОСКОНАЛЕНИЙ МЕТОД ПОБУДОВИ ПЕРСОНАЛІЗОВАНОГО ПЕРЕЛІКУ ТОВАРІВ ТА ПОСЛУГ В РЕКОМЕНДАЦІЙНИХ СИСТЕМАХ

2.1 Формалізація задачі

Задача:

Дано:

$U = (u_i)_{i=1}^{N}$ – безліч N користувачів;

$V = (v_a)_{a=1}^{M}$ – безліч M товарів;

$R = (r_{ia})_{i=1,a=1}^{N,M}$ – сильно розріджена матриця оцінок.

Необхідно спрогнозувати невідомі елементи r_{ia} для кожного і-го користувача та а-го товару.

Для вирішення цієї задачі формулюються припущення, що оцінки користувачів на всьому відрізку часу постійні і що схожі користувачі схильні ставити схожі оцінки одним товарам. Для контролю якості системи, що розробляється, використовуються наступні метрики [7]. Оцінки точності – MAE (середня абсолютна помилка), RMSE (середня квадратична помилка):

$$MAE = \sum_{i=1}^{N}\sum_{a=1}^{M}\frac{|r'_{ia} - r_{ia}|}{N}; \quad (2.1)$$

$$RMSE = \sqrt{\sum_{i=1}^{N}\sum_{a=1}^{M}\frac{(r'_{ia} - r_{ia})^2}{N}}. \quad (2.2)$$

Де $\frac{|T|}{Ln}$ – точність, описує релевантність позицій у списку;T – безліч рекомендацій;

Ln – довжина всього списку рекомендацій;

повнота – $Recall = \frac{|T|}{(|T| + |G|)}$;

G – безліч елементів, яких не було в списку рекомендацій, але які мали бути рекомендовані.

Для подальшого порівняння було взято сучасні алгоритми колаборативної фільтрації – Baseline predictors, KNN, SVD, SVD++, Slope one, Co-clustering, які детально представлені в [8], [9], [10], [11].

У якості тестової бази для аналізу ефективності РС використовувалася Retailrocket dataset. Дані про поведінку, тобто події,такі, як кліки, додавання в кошик, транзакції, є взаємодіями з товарами інтернет-магазину, які були зібрані протягом 4.5 міс. Відвідувач може здійснити три типи подій, а саме «перегляд товару», «додавання товару в кошик» або «купівля». Всього є 2 756 101 подія у базі, у томучислі 2 664 312 переглядів, 69 332 додавань до кошика та 22 457 покупок, зроблених 1 407 580 унікальними відвідувачами. Ця тестова база повністю підходить для поставлених перед нами завдань.

Дані у таблиці 2.1 отримано після перехресної перевірки аналітичної моделі алгоритмів. Датасет розбивається на п'ять частин, чотири частини використовуються для навчання моделі та одна частина – для тестування. Вибір множин для навчання та тестування повторюється п'ять разів. Критеріями порівняння алгоритмів було обрано:

– rmse – середня квадратична помилка;
– mae – середня абсолютна помилка;
– fit time – час роботи алгоритму, вказаний у секундах;
– precision – релевантність позицій у списку рекомендацій (точність системи);

– recall – повнота системи.

Найточніші прогнози видає алгоритм SVD++. Найвища помилка виявлено у алгоритму Co-Clustering, причому інші величини помилок займають середні значення. Найбільш швидкими є засновані на обчисленні середніх величин різниці між оцінками Baseline predictors та Slope one. Далі йдуть KNN і SVD, що мають все ще прийнятну швидкість роботи. І явно виділяється SVD++, швидкість роботи якого у кілька десятків разів менша, ніж у попередньої групи. Також виділяється три основні групи з приблизно однаковими значеннями точності та повноти. Найкращими характеристиками володіють алгоритми KNN, SVD++ та Co-Clustering. Найбільш неповні та неточні результати видає Baseline predictors.

Таблиця 2.1 Порівняльна таблиця результатів роботи алгоритмів

Алгоритм роботи рекомендаційної системи	Середня квадратична помилка	Середня абсолютна помилка	Час роботи алгоритму, сек	Релевантність позицій у списку рекомендацій (точність системи)	Повнота системи
Baseline predictors	0,9551	0,7562	0,14	0,6587	0,5982
SVD	0,94235	0,74365	7,63	0,7145	0,66548
SVD++	0,912548	0,7203	411,25	0,7896	0,68854
KNN	0,92564	0,73658	7,698	0,7845	0,69851
Slope One	0,945687	0,74589	0,99	0,7236	0,65234
Co-Clustering	0,96521	0,75684	1,758	0,7836	0,68743

Ґрунтуючись на отриманих проміжних результатах, було запропоновано створити гібридну систему [12], [13], поєднавши алгоритми ко-кластеризації та найближчих сусідів. Обидва алгоритми базуються на різних підходах для побудови моделі: ко-кластеризація – на латентній моделі, а найближчі сусіди – на кореляційній. Використання різних підходів дозволяє частково вирішити недоліки кожної розробленої моделі.

Так, наприклад, для вирішення проблеми холодного старту запропоновано об'єднати item-based KNN з базовими предикторами.

Кореляція товарів дозволяє частково вирішити проблему для нових користувачів. У якості другого алгоритму розглядалися кандидати з часом навчання нижче, ніж у основного, таким чином було враховано проблему масштабованості. Кращим кандидатом визначено алгоритм ко-кластеризації, тому що за інших переваг він також не схильний до впливу розрідженості даних, оскільки заснований на моделі, а не на кореляції.

В алгоритмі KNN рекомендації продукту будь-якому користувачеві вибираються з продуктів, що сподобалися користувачам з найближчими оцінками до аналізованого. Для цього порівнюватимемо вектор-фактори продуктів – рядки матриці R. Залишимо лише ті продукти, оцінки користувачів яких нам відомі, та виділимо ті елементи r_{ia} матриці R, які нам відомі спочатку в обох векторах. Тепер обчислимо схожість цих двох векторів. Як міра подібності ω_{ab} обраний коефіцієнт кореляції Пірсона, оскільки він враховує тенденцію РС прогнозувати рейтинги вище тих товарів, яким у середньому користувачі визначають вищий рейтинг:

$$\omega_{ab} = \frac{(r_{ia}-\mu_a)(r_{ib}-\mu_b)+\cdots+(r_{Na}-\mu_a)(r_{Nb}-\mu_b)}{\sqrt{(r_{ia}-\mu_a)^2+\cdots+(r_{Na}-\mu_a)^2}\sqrt{(r_{ib}-\mu_b)^2+\cdots+(r_{Nb}-\mu_b)^2}}\ , \qquad (2.3)$$

Побудуємо прогноз нових рейтингів r'_{ia}. З цією метою для найбільш схожих товарів обчислюємо виважене середнє із продуктів, які вже оцінив користувач. Далі KNN (a) – безліч k найближчих об'єктів до a:

$$r'_{ia} = \sum_{b\in KNN(a)} r_{ib}\omega_{ab} / \sum_{b\in KNN(a)} |\omega_{ab}|\ , \qquad (2.4)$$

Віднімемо з r_{ib} базові предиктори z_{ib} та врахуємо їх при формуванні прогнозу:

$$r'_{ia} = z_{ib} + \sum_{b \in KNN(a)} (r_{ib} - z_{ib})\omega_{ab} / \sum_{b \in KNN(a)} |\omega_{ab}| \qquad (2.5)$$

Для кластеризації визначимо функцію втрат як квадратичну помилку, а також уточнимо клас наших низькорівневих структур. Знайдемо оптимальну кластеризацію для користувачів та товарів. Представимо кластеризацію у вигляді відображення безлічі об'єктів на безліч кластерів: $\rho: U \to U'$ для користувачів і $\gamma: V \to V'$ для товарів, де $U' = (u'_i)_{i=1}^{K}$ безліч K кластерів користувачів, $V' = (v'_a)_{a=1}^{L}$ – безліч L кластерів товарів. Область ю визначення ρ і γ є число Стерлінгу першого роду, оскільки оптимальна кластеризація вибирається з усіх можливих розбиття множин U і V.

Побудуємо наступну апроксимацію рейтингу товару a для користувача i:

$$r'_{ia} = \overline{C_{gh}} + (\mu_i - \overline{C_g}) + (\mu_a - \overline{C_h}) \qquad (2.6)$$

Введемо функції кластеризації $g = \rho(i); h = \gamma(a)$ – кластери аналізованих користувача і товару відповідно, де $\rho(i)$ – деяке розбиття безлічі користувачів на K кластерів з усіх можливих розбиттів, $\gamma(a)$ – аналогічно для безлічі товарів; $\overline{C_{gh}}$, $\overline{C_g}$, $\overline{C_h}$ – середні рейтинги користувача відповідно ко-кластера, кластера користувачів та кластера товарів, μ_a і μ_i середні рейтинги користувача i та товару a. Позначимо W_{ia} індикатор оцінки, який приймає значення 0, якщо $r_{ia} = 0$, і 1 в іншому випадку:

$$\overline{C_{gh}} = \sum_{(f|\rho(i)=g)} \sum_{(a|\gamma(a)=h)} r_{ia} / \sum_{(f|\rho(i)=g)} \sum_{(a|\gamma(a)=h)} W_{ia},$$

$$\overline{C_g} = \sum_{(f|\rho(i)=g)} \sum_{a=1}^{M} r_{ia} / \sum_{(f|\rho(i)=g)} \sum_{a=1}^{M} W_{ia},$$

$$\overline{C_h} = \sum_{f=1} \sum_{(a|\gamma(a)=h)}^{N} r_{ia} / \sum_{f=1} \sum_{(a|\gamma(a)=h)}^{N} W_{ia}, \qquad (2.7)$$

$$\mu_i = \sum_{a=1}^{M} r_{ia} / \sum_{a=1}^{M} W_{ia},$$

$$\mu_a = \sum_{i=1}^{N} r_{ia} / \sum_{i=1}^{N} W_{ia}.$$

Використовуючи отримані рейтинги r'_{ia}, знаходимо оптимальні функції кластеризації для користувачів та товарів через мінімізацію помилки щодо невідомих. Таким чином ми вибираємо найкраще розбиття множин N та M на кластери. Тут r'_{ia} залежить від розбиття, тобто від вибору таких функцій ρ і γ, для яких сумарне квадратичне відхилення мінімально:

$$\min_{\rho \to \gamma} \sum_{i=1}^{N} \sum_{a=1}^{M} (r_{ia} - r'_{ia})^2. \qquad (2.8)$$

Підставимо отримані значення формулу (2.3) і сумістимо її з першою моделлю (2.2). Регуляція моделей здійснюється за допомогою коефіцієнта α. Коефіцієнт α задає ваги при голосуванні для кластерної моделі та моделі, яка використовує базові предиктори. Вага при голосуванні

дозволяють враховувати те, що різні моделі мають різну цінність. Підсумкова модель виглядає так:

$$r'_{ia} = \alpha\left(z_{ia} + \sum_{b \in KNN(a)} (r_{ib} - z_{ib})\omega_{ab} / \sum_{b \in KNN(a)} |\omega_{ab}|\right) \qquad (2.9)$$
$$+ \quad +(1-\alpha)\left(\overline{C_{gh}} + \left(\mu_i - \overline{C_g}\right) + \left(\mu_a - \overline{C_h}\right)\right).$$

Таким чином, розроблено гібридний метод у ситуації холодного старту на основі проведеного порівняльного аналізу алгоритмів колаборативної фільтрації. Для цього використовувалися такі критерії: точність, повнота, швидкість навчання алгоритму, а також середньоквадратична та абсолютна помилки. Також було проаналізовано та частково вирішено проблеми та слабкі сторони кожного з підходів колаборативної фільтрації. Отримана система має помітне поліпшення точності характеристик порівняно з окремими алгоритмами колаборативної фільтрації, що використовуються, при відносноневеликому збільшенні часу навчання.

2.2 Технологія формування персоналізованого переліку товарів та послуг у рекомендаційній системі на основі знань

Взаємодія між користувачем і рекомендувачем може мати форму розмовних систем, систем на основі пошуку або навігаційних систем. Такі різні форми керівництва можуть існувати як ізольовано, так і в комбінації, і вони визначаються таким чином:

1. Розмовні системи: у цьому випадку переваги користувача визначаються в контексті циклу зворотного зв’язку. Основною причиною цього є те, що предметна область є складною, і переваги користувача можна визначити лише в контексті ітераційної розмовної системи[18].

2. Системи на основі пошуку: у системах на основі пошуку переваги користувача визначаються за допомогою попередньо встановленої послідовності запитань, таких як

наступне: «Ви віддаєте перевагу будинку в передмісті чи в межах міста?»

3. Рекомендація на основі навігації: у рекомендації на основі навігації користувач вказує кількість запитів на зміну елемента, який наразі рекомендований. За допомогою ітеративного набору запитів на зміну можна отримати бажаний елемент. Приклад запиту на зміну, указаний користувачем, коли рекомендувався конкретний будинок, виглядає наступним чином: «Я хотів би подібний будинок приблизно в 5 милях на захід від поточного рекомендованого будинку». Такі рекомендаційні системи також називають системами критичних рекомендацій [18].

Ці різні форми керівництва добре підходять для різних типів систем рекомендацій. Наприклад, системи критики, природно, розроблені для тих, хто рекомендує на основі конкретних випадків, тому що кожен критикує конкретний випадок, щоб досягти бажаного результату. З іншого

З боку, систему, засновану на пошуку, можна використовувати для встановлення вимог користувача до рекомендацій на основі обмежень. Деякі форми керівництва можна використовувати як із системами на основі обмежень, так і з системами на основі випадків. Крім того, різні форми керівництва також можуть використовуватися в комбінації в системі, заснованій на знаннях. Не існує строгих правил щодо того, як можна проектувати інтерфейс для системи, заснованої на знаннях. Мета завжди полягає в тому, щоб провести користувача через комплекс простіру продукту[5].

Типові приклади інтерактивного процесу в рекомендаціях на основі обмежень і рекомендаціях на основі випадків. (Рисунок 2.1-2.2).

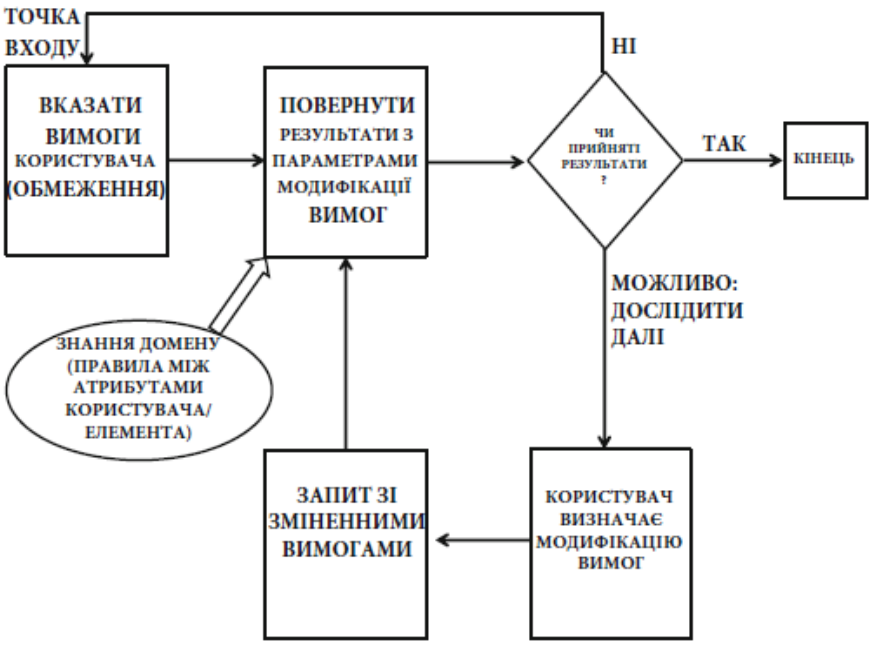


Рисунок 2.1 – Огляд інтерактивного процесу в рекомендаціях взаємодії на основі обмежень

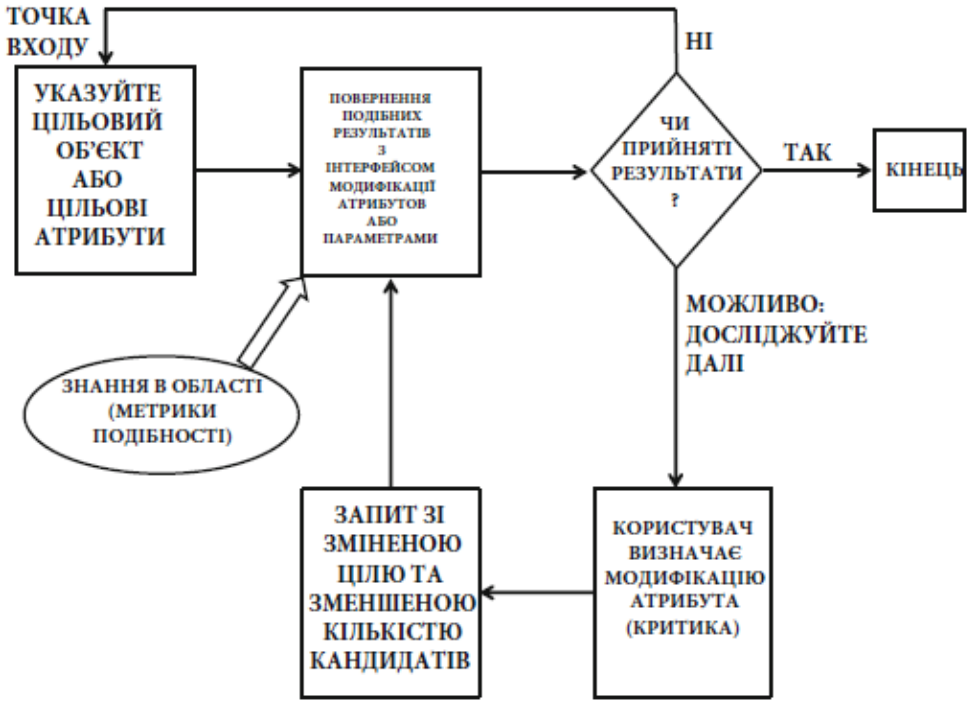


Рисунок 2.2 – Огляд інтерактивного процесу в рекомендаціях взаємодії на основі прецендентів

Загальний інтерактивний підхід досить схожий. Основна відмінність у двох випадках полягає в термінах як користувач вказує запити та взаємодіє з системою для подальшого уточнення. У системах, заснованих на обмеженнях, конкретні вимоги (або обмеження) вказуються користувачем, тоді як у системах, заснованих на регістрах, вказуються конкретні цілі (або випадки). Відповідно, у двох системах використовуються різні типи інтерактивних процесів і предметних знань[19]. У системах на основі обмежень вихідний запит змінюється шляхом додавання, видалення, модифікації, або послаблення вихідного набору вимог користувача.

У системах, заснованих на регістрі, або ціль змінюється за допомогою взаємодії з користувачем, або результати пошуку скорочуються за допомогою використання спрямована критика. У такій критиці користувач просто вказує, чи потрібно певним чином збільшити, зменшити чи змінити певний атрибут у результатах пошуку. Такий підхід представляє більш розмовний стиль, ніж просто зміна цілі. В обох цих типах систем загальна мотивація полягає в тому, що користувачі часто не в змозі точно сформулювати свої вимоги в складній області продукту. У системах, заснованих на обмеженнях, ця проблема частково вирішується за допомогою бази знань правил, які відображають вимоги користувача до атрибутів продукту.

У системах, заснованих на прецедентах, ця проблема вирішується за допомогою розмовного стилю критики. Інтерактивний аспект є спільним для обох систем, і він має вирішальне значення для того, щоб допомогти користувачам виявити, як елементи в складній області продукту відповідають їхнім потребам[3].

Слід зазначити, що більшість форм систем рекомендацій, заснованих на знаннях, значною мірою залежать від описів елементів у формі реляційних атрибутів, а не розглядають їх як текстові ключові слова, як системи, засновані на контенті.

Це природний наслідок складності, притаманної рекомендаціям, заснованим на знаннях, які є предметно-спеціальними знання можна легше закодувати за допомогою реляційних атрибутів. Наприклад, атрибути для набору будинків у програмі нерухомості проілюстровано у (рисунок 2.3).

Item-Id	Beds.	Baths.	Locality	Type	Floor Area	Price
1	3	2	Bronx	Townhouse	1600	220,000
2	5	2.5	Chappaqua	Split-level	3600	973,000
3	4	2	Yorktown	Ranch	2600	630,000
4	2	1.5	Yorktown	Condo	1500	220,000
5	4	2	Ossining	Colonial	2700	430,000

Рисунок 2.3: Приклади атрибутів у заявці з рекомендаціями щодо купівлі будинків.

У рекомендаціях на основі регістру показники подібності визначаються в термінах цих атрибутів, щоб забезпечити подібні відповідності цільовим будинкам, наданим користувачем. Зауважте, що кожен реляційний атрибут матиме різне значення та вагу в процесі зіставлення залежно від критеріїв домену. У системах, заснованих на обмеженнях, запити вказуються у формі вимог щодо цих атрибутів, таких як максимальна ціна на будинок або конкретна місцевість.

Метод побудови рекомендацій на основі адаптації обмежень:

Етап 1 – класифікація від користувача обмеження.

Етап 2: Пошук уточнюючих обмежень.

Крок 2.1: Пошук уточнюючих обмежень для визначеного класу об’єктів.

Крок 2.2: Пошук за минулим вибором користувача.

Етап 3 – Перевірка необхідності додаткового уточнення обмежень. Якщо необхідно то повторити Етапи 1,2.

Етап 4 – Відбір підмножини пропозиції товарів згідно с пропонованими обмеженнями.

Етап 5 – формування рекомендованого переліку товарів.

3 ПОНЯТТЯ ПЕРСОНАЛІЗОВАНОГО ПЕРЕЛІКУ ТОВАРІВ ТА ПОСЛУГ. ОБРАННЯ МОВ ПРОГРАМУВАННЯ, ІНСТРУМЕНТІВ РОЗРОБКИ. ПОБУДОВА ВЕБ-ДОДАТКУ

3.1 Поняття персоналізованого переліку товарів та послуг

Для просування бізнесу дедалі значимішою стає сфера інформаційно-комунікаційних технологій. В даний час важко переоцінити її вплив на взаємодію брендів та споживачів. Спостерігається дедалі активніша інтеграція різних бізнес-процесів в інтернет-простір, а іміджкомпанії все більше залежить від її позиціонування в соціальних мережах та засобах масової інформації в мережі Інтернет. За даними Асоціації комунікаційних агентств України сумарна частка реклами у засобах її розповсюдження за 2021 рік збільшилася на 13%, при цьому динаміка у друкованих засобах масової інформації (ЗМІ) знизилася в середньому на 15%, а в інтернеті – збільшилася на 24% порівняно з минулим роком [10].

Поряд з цим можна спостерігати іншу тенденцію: ринок став більш гнучким та привітним до індивідуальних покупців. Поступово зникають постачальники: практично кожен може замовити певний товар чи послугу безпосередньо, минаючи довгий ланцюжок постачальників і, як наслідок, уникаючи збільшення вартості товару кілька разів. Це призвело до зміни загальної картини ринку: бренди стали більше розуміти своїх споживачів та те, як можна з ними взаємодіяти. Звернення до широкої аудиторії вже неефективне, набагато більше відгуку приносить індивідуальний підхід та пошук персональних шляхів взаємодії зі споживачем. Реклама стала більш «точковою», орієнтованою на конкретну аудиторію і відповідає за певну потребу. Не останню роль у цьому відіграє технологічний розвиток засобів комунікації, який сьогодні може надати фантастичні можливості.

Дослідження, що проводяться в даній галузі [13], [17], [18] мають один істотний недолік. У них визнається наявність персонального підходу до споживача, проте в даний час не сформульовані етапи та порядок реалізації технології персоналізації і даний підхід не розглядається в загальному контексті стратегії просування бренду, хоча персоналізація вже зараз є невід'ємною складовою її успіху.

У процесах розвитку бізнесу та просування брендів у 2022 р. виділяють два провідні тренди: локалізація та перетин різних сфер бізнесу та культури [20]. Якщо раніше фахівці відзначали тенденцію до глобалізації, то сьогодні деякі країни схиляються до ізоляції та розвитку власної ідентичності, і не останню роль у цьому відіграло питання кібербезпеки. Проте найперспективнішим вектором для бізнесу є баланс між патріотизмом та глобалізацією. Інший тренд – перетин секторів розвитку. На даний момент практично неможливо знайти проблему, яка пов'язана лише з однією сферою і не торкається інших. Інтернет-комунікації сприяють миттєвому поширенню інформації, розвитку протилежних точок зору, поширенню політичних, соціальних, економічних та культурних тенденцій. Події, що відбуваються в одній країні, моментально викликають реакцію по всьому світу, формуючи такий унікальний тренд, як міжсекційний футуризм – перетин соціальних питань, які можуть розглядатися тільки в багатоскладовому, складному контексті. Сучасним компаніям необхідно враховувати, який вплив мають їхні продукти та послуги на соціум, оцінювати взаємозв'язки бізнесу з навколишнім середовищем та суміжними сферами діяльності.

Поруч із глобальними трендами, які стосуються переважно великих компаній, виявлено й найбільш локальні тренди, на які впливають поведінка окремих груп людей та їх розвиток у персональному ключі. На думку World Economic Forum, світ рухається до четвертої індустріальної революції, коли першорядне значення матиме не капітал, а талант. Ці зміни можна відзначити вже зараз: все більше фахівців здобуває освіту

віддалено та самостійно розвиває професійні навички. Дедалі більше людей вважають за краще заробляти на ідеях, контенті та особистій власності, а роботодавці відходять від традиційних форм організації робочого процесу, експериментуючи з форматами та залучаючи віддалених фахівців. Інтерес представляє дослідження Edelman Trust Barometer, згідно з яким 8 із 10 людей чекають, що бізнес вирішуватиме соціальні проблеми суспільства. Переробка відходів, екологія та інші проблеми все частіше стають об'єктом уваги стартапів та великихкомпаній. За даними Consumer News and Business Channel (CNBC) знижується вплив моди: більшість сезонних речей залишаються незатребуваними, а споживачі роблять вибір не на користь того, що модно,а на користь індивідуального стилю.

Отже, вищеперелічені дослідження разом дають уявлення про загальну картину взаємодії бізнесу з цільовою аудиторією. Від виробників товарів та послуг чекають уваги до соціальних проблем, оперативної реакції на події, що відбуваються у світі, а також персональної уваги до самих споживачів та обліку індивідуальних потреб.

Виробники, у свою чергу, все частіше здійснюють вибір на користь якості, а не кількості, акцентують увагу на таргетованій рекламі та адресних рекламних кампаніях і звертаються до суміжних сфер при формуванні стратегій розвитку. Величезну роль у цьому грає розвитоктехнологій: нові винаходи регулярно впливають на всі сфери діяльності, вносячи корективи, а найчастіше кардинально змінюючи напрямок розвитку. Проведений аналіз показав, що основною проблемою в галузі просування бренду за допомогою сучасних інтернет-технологій є надто динамічна зміна середовища. Очевидно, що інформація досить швидковтрачає актуальність, а методи, які застосовуються сьогодні, можуть у найближчому майбутньому виявитися неефективними.

Однак, коли йдеться про персоналізацію, ситуація виглядає дещо інакше. Цей тренд довговічніший, оскільки пов'язаний з особистим

ставленням споживача, з його власним вибором. На відміну від світових глобальних тенденцій, які мають властивість змінюватися, можливість підлаштовувати ситуацію під себе і робити вибір, ґрунтуючись наособистих уподобаннях – це фактично природна потреба людини. При такому стрімкому розвитку ринку та різноманітності інформації, найвірніший напрямок розвитку – персональне звернення до цільової аудиторії. Сучасній аудиторії досить важко нав'язати товар чи послугу, яка їй не потрібна, скоріше навпаки: у наш час закон «попит визначає пропозицію» отримує найнаочніше та конкретне підтвердження та діє з максимальною ефективністю. Це підводить до поняття персоналізації як таргетованого надання товарів та послуг з урахуванням особистих потреб, цілей, особливостей та способу життя конкретного споживача чи сегмента цільової аудиторії.

Суть персоналізованого підходу полягає в тому, що спочатку необхідно зібрати якнайбільше даних про користувачів, а далі, аналізуючи отриману інформацію, скласти психологічні портрети кожного з них і сформулювати їм таргетовану пропозицію. Найбільшої популярності цей підхід, що становить основу стратегій просування багатьох брендів, отримав завдяки корпорації Cambridge Analytica, яка активнозастосовувала його під час передвиборчої кампанії президента США. Політичні висловлювання були сформовані з урахуванням статі, національності, району проживання та переваг, що допомогло сформулювати як найпривабливіші для електорату гасла.

Подібний підхід широко використовується не тільки у великомасштабних акціях та кампаніях: в даний час можливості для збирання та аналізу персональних даних користувачів надають практично всі соціальні мережі, які є одним із найважливіших майданчиків для просування. Наприклад, Facebook збирає та зберігає інформацію про користувачів, дозволяючи рекламодавцям створювати найефективніші пропозиції, адресовані конкретним користувачам – мамам з маленькими

дітьми, людьми, які живуть із сусідами або у відносинах на відстані. Однак навіть таких докладних даних недостатньо, щоб сформулювати грамотну персоналізовану пропозицію. Ефективність та наповненість націленої реклами надає споживчий інсайт. Це справжній мотив споживача, який спонукає його до покупки [24]. Правильний інсайт означає справжні проблеми та мотиви споживача та є невід'ємною частиною його життя. Він відповідає питанням: як думає споживач? про що думає споживач? чому він поводиться саме так? що він по-справжньому відчуває? при цьому грамотно сформульований інсайт завжди є відкриттям, що дозволяє поглянути на речі під новим кутом, стимулює переоцінку існуючих знань та досвіду.

3.2 Визначення принципів формування поняття персоналізованого переліку товарів та послуг

Технологію персоналізації товарів та послуг умовно пропонується розділити на три етапи: збирання даних, їх аналіз, формування пропозиції з урахуванням інсайту. Розглянемо кожен етап докладніше.

Перший етап – збір даних – підготовчий. На цій стадії проводяться різні опитування, анкетування, персональні інтерв'ю з клієнтами бренду та глибинні дослідження. Етап збору даних може займати від кількох місяців до кількох років – залежно від цілей та завдань, які ставить компанія.Однак, щоб зібрати об'єктивніші дані, необхідно провести низку досліджень, охоплюючи найбільший сегмент цільової аудиторії. Високу ефективність показують глибинні інтерв'ю, які проводяться з метою якнайкраще вивчити психологію клієнта та визначити його проблеми та мотиви. Збору даних передує етап формування запиту, під час якого формулюються цілі та завдання досліджень та складаються основні питання, відповіді на які ці дослідження мають дати.

Після збору даних слід розпочати їх аналіз. Етап аналізу даних може тривати кілька тижнів, визначальним критерієм тут служить кількість аналізованих параметрів, мета і завдання. Насамперед безпосередньо оцінюються кількісні та якісні показники, отримані за допомогою досліджень, а потім аналізується можливість їх застосування для виконання поставленої мети, а також проводиться більш глибокий аналіз для з'ясування глибинних причин і мотивів, якими керується споживач. Ці глибинні мотиви (або інсайт) – прихована інформація, яку практично неможливо одержати методом опитування чи анкетування [26]. На цьому етапі вивчення споживачів рекомендується підключити фахівців (психологів, соціологів), щоб розглянути отримані дані максимально ефективно. Зібрана інформація систематизується для подальшого використання розробки пропозиції та персоналізації товарів тапослуг. Саме цьому етапі формулюється інсайт, що є основою стратегії просування.

Заключний етап – формування пропозиції з урахуванням зібраних та проаналізованих даних. При його розробці враховується не тільки інформація про споживача, але й цілі та завдання самої компанії. Отримане рішення має бути синтезом того, що хоче споживач, і того, що виробник може запропонувати йому.

Механізм персоналізації товарів та послуг схожий з традиційними маркетинговими дослідженнями, які зазвичай проводяться на початковому етапі розробки стратегії просування, проте має суттєву відмінність, тому що його мета зрештою не просто дати уявлення про ринок та аудиторію, ай сформулювати глибинні мотиви її безпосередніх учасників. Якщо маркетингові дослідження проводяться для самої компанії, дослідження з метою персоналізації товарів та послуг повинні відповідати ключовим потребам клієнта [25]. Якщо предметом маркетингових досліджень найчастіше є ринок, то під час проведення у разі персоналізації товарів та послуг – цим предметом є сам клієнт. Отже, метод персоналізації значно

підвищує ефективність стратегії просування. По-перше, пропозиція товару та послуги стає адресною, що дозволяє швидше досягти кінцевої мети – привернення уваги цільової аудиторії. По-друге, персоналізація товарів та послуг, незважаючи на трудомісткість та затратність на першому етапі – зборі даних та їх аналізі, надалі дозволяє уникнути зайвих витрат на нецільову рекламу. Нарешті, персоналізація товарів та послуг сприяє глибшому вивченню цільової аудиторії бренду, що дозволяє надалі формувати додаткові пропозиції, які отримають відгук, та ефективніше оцінювати нішу ринку, в якій бренд існує.

Ця технологія апробована на прикладі британського бренду Bronte by Moon, який відкрив представництво на території України у 2020 році. Цей бренд займається виготовленням виробів з натуральної шерсті в британському стилі, все виробництво сконцентровано на території Великобританії, на фабриці повного циклу в Йоркширі. Бренд має свої ключові цінності, засновані на менталітеті та культурі жителів Великобританії: вірність традиціям, цінність сім'ї та родинних зв'язків, спадкоємність, якість та довговічність виробів, локальність бренду (виробництво на території Великобританії), екологічність.

При розробці стратегії просування бренду на українському ринку враховано дослідження, що проводяться у Великій Британії. Однак різниця менталітету, клімату та національні особливості виявили необхідність проведення аналогічних досліджень на території України з метою персоналізації товарів та послуг. Основними споживачами продукції Bronte by Moon у Великій Британії є сім'ї, які, у тому числі мають аристократичне походження, з рівнем доходу вище середнього, а також індивідуальні покупці, зацікавлені у купівлі якісних виробів у британському стилі. Крім того, одним із напрямків діяльності фабрикиBronte by Moon у Йоркширі є виготовлення тканини та виробів із вовни для відомих британських брендів. Україні ж представлено виключно

напрямок, пов'язаний із продажем готових аксесуарів преміум-класу, отже, цільова аудиторія та її потреби разюче відрізняються.

Перший етап – збір даних – зайняв близько 6 місяців. За цей час було проведено 17 опитувань, які торкалися основних моментів взаємодії покупця з брендом та 96 глибинних інтерв'ю, що розкривають справжні мотиви та проблеми цільової аудиторії. Етап обробки та аналізу даних зайняв 3 місяці, за цей час сформульовано основний інсайт, виявлено ключові потреби споживачів та особливості цільової аудиторії. За результатами досліджень було складено портрет персони-моделі, на яку орієнтується бренд: жінка віку 30+, з рівнем доходу вищим за середній, зацікавлена у створенні власного, унікального стилю. Було визначено ключові цінності: свобода, унікальність, самовдосконалення та самореалізація. Серед основних показників, пов'язаних зі сферою діяльності цільової аудиторії, було виявлено дві категорії: робота та подорожі. Саме ці сфери діяльності становлять клієнтам бренду найбільшу цінність, нарівні з особистісними цілями: самореалізація і свобода вибору.

Спочатку позиціонування бренду не мало під собою чітко сформульованої бази, тому було розмитим та несфокусованим. У цільову аудиторію включалися споживачі по всій території України, як і раніше, що представництво бренду розташовується на Далекому Сході. Вироби рекламувалися за аналогією з іншими марками: виключно за рахунок опису позитивних властивостей продукту, для роботи з клієнтами та їх залучення використовувалися стандартні методи та прийоми (реклама в інтернет-ЗМІ, участь у виставках-продажах, розміщення рекламних буклетів у партнерів, участь у заходах) з можливістю представлення бренду тощо). Відсутні два найважливіші чинники: ціннісна база та персоналізація товарів та послуг.

Проведені дослідження надають можливість сформулювати основні цінності, важливі клієнта: унікальність, самореалізація, статусність; а також розділити всю цільову аудиторію на дрібніші категорії, щоб

сформулювати всередині кожної з них свої ключові цінності та проблеми та сформувати індивідуальну пропозицію. Це дозволяє подивитися на потенційних клієнтів під іншим кутом та включити до стратегії просування додаткові заходи, що сприяють підвищенню лояльності клієнтів до бренду. Наприклад, організація кількох клубів за інтересами, що дали клієнтам можливість реалізовувати власні потреби в особистісному зростанні, спілкуванні та обміні досвідом. Крім того, сформульовано персональні пропозиції для ключових груп споживачів, які підвищили продаж за рахунок адресної реклами та персонального звернення до споживача. Дані персональні пропозиції спрямовані на вирішення конкретних проблем: готові рішення для гардеробу, складання простих та універсальних образів за допомогою виробів, фокусування на корисності певних властивостей продукту (довговічність, екологічність, натуральність, гігроскопічність, естетичні характеристики). Збільшення обсягу досліджень для персоналізації товарів та послуг до традиційної стратегії просування послуг дозволило підвищити ефективність рекламних пропозицій на 17%,а також збільшити конверсію у соціальних мережах бренду на 24%.

Таким чином, застосування технології персоналізації дозволяє покращити традиційну схему продажу товарів та послуг, за якої основною метою є безпосередньо реалізація товару та отримання вигоди.

Технологія персоналізації дозволяє змінити існуючу схему на більш привабливу для споживача систему, де за товаром стоять певні цінності, які є відповіддю на їх невисловлені потреби. У мінливих умовах ринку цей підхід є більш ефективним, оскільки саме персоналізація є точкою перетину інформаційно-комунікаційних технологій та клієнторієнтованих стратегій розвитку. В умовах, коли споживач сам визначає необхідність вибору товарів та послуг, адресний поступ стає все більш актуальним. Отже, запропонована технологія персоналізації має стати невід'ємною частиною більшості стратегій просування бренду.

3.3 Визначення вимог до системи

Метою дипломного проекту є розробка знання-орієнтованих методів побудови персоналізованого переліку товарів та послуг в рекомендаційних системах.

Тема сайту – продукти телебачення (фільми, мультфільми, програми передач, кліпи, тощо). Відвідувачам сайту повинна надаватися можливість знаходити телевізійні продукти, що їх цікавлять, використовуючи пошук за фільтрами, переглядати стислий зміст, та рейтинговий бал, і на основі рейтингу отримувати рекомендації до перегляду. Таким чином, створюваний сайт дозволить користувачам дізнаватися про раніше невідомі їм телевізійні проекти, які повинні їм сподобатися.

Для досягнення поставленої мети необхідно вирішити такі завдання:

– вивчити та проаналізувати існуючі методи побудови рекомендаційних систем та платформи для розробки сайтів;
– спроектувати клієнт-серверну архітектуру програми;
– розробити алгоритм рекомендаційної системи;
– реалізувати додаток та перевірити його працездатність.

3.4 Аналіз алгоритму та систем

Останнім часом стає все більш популярним створення систем, які здатні вгадувати переваги та потреби користувачів та на основі цього пропонувати відповідні рішення. Такі системи, які називаються рекомендаційними, мають безліч застосувань. Великі компанії, такі як Amazon, Apple, eBay, Pandora і т.д., використовують рекомендаційні системи у складі своїх сервісів. Що важливо, рекомендації формуються для конкретного користувача, виходячи з його особистих даних та інтересів. Такий підхід спрощує користувачеві роботу з великим обсягом даних, допомагає скоротити час пошуку потрібного рішення,

забезпечуючи релевантність інформації. Проте найчастіше рекомендаційні системи розробляються на вирішення завдань конкретної області. Наприклад, Amazon спеціалізується на продуктах, що продаються ним, Netflix на фільмах і т.д. [21], [22].

Великі компанії мають необхідні кошти та кваліфікованих фахівців для того, щоб розробити власну рекомендаційну систему. Компанії ж, які тільки починають розвиватися і мають необхідність створення сервісу рекомендацій, часто не мають необхідних ресурсів і часу для цього. Розробка універсальної рекомендаційної системи, що адаптується до різних областей, може вирішити цю проблему. Можна відзначити, що при розробці рекомендаційних систем застосовуються загальноприйняті походи, що базуються на реалізації таких методів машинного навчання як колаборативна фільтрація та фільтрація на основі змісту [3] та ін., що дозволяє поставити завдання реалізації універсального рішення, що базується на використанні цих методів. Налаштування на конкретні предметні області може бути виконане на основі моделей, що створюються з використанням даних, що надаються компаніями- клієнтами (користувачами).

Сервіс рекомендацій, що налаштовується на декілька областей застосування, повинен мати в основі не тільки модель рекомендаційної системи, її предметної області та засоби збору даних про користувачів системи, а й включати додаткові модулі її підтримки, зокрема засоби аналізу даних. Дані, що використовуються для вироблення рекомендацій, повинні відповідати певним вимогам: актуальність, правильність, унікальність, повнота, структурованість тощо. Тільки за наявності якісних даних можливе прийняття рішень, які найбільш повно відповідають потребам користувачів. Як показав аналіз, на даний момент не існує рішень, які б врахувати всі вимоги до системи.

Останнім часом популярність рекомендаційних систем стала дуже великою. Це пояснюється прагненням компаній спростити вибір

користувача у тій чи іншій ситуації та отримати від цього прибуток. З огляду на велику затребуваність систем, які автоматично формують пропозиції користувачам з урахуванням їх інтересів, робляться спроби розробки універсальних рекомендаційних систем.

На даний момент існують такі рішення, пов'язані зі створенням рекомендаційних систем.

По-перше, це послуги з модулем рекомендацій для певної предметної області. Такий підхід використовується, наприклад, у Apple, Netflix, Amazon та інших компаніях.

По-друге, це дослідження, створені задля створення універсальної рекомендаційної системи, що забезпечує роботу з різними предметними областями (Unresyst); різні бібліотеки, у яких реалізовані методи, які можна використовуватиме створення власної рекомендаційної системи.

Третім рішенням є універсальні рекомендаційні системи (Apache Prediction IO), які є закінченим програмним продуктом.

Для порівняння даних рішень було виділено такі критерії:

– налаштування системи на будь-яку предметну область (універсальність);

– можливість інтеграції системи в продукт сторонніх розробників (використання системи як вбудованого модуля у власний сервіс);

– наявність компонентів збору даних для подальшогоформування оцінок;

– наявність модуля аналізу для забезпечення якості даних та рекомендацій;

– наявність готового програмного продукту над ринком програмного забезпечення.

Порівняння існуючих рішень наведено у таблиці 3.1.

Використовується така шкала для оцінювання рішень:

– 0 – рішення не виконує зазначеної функції;
– 1 – рішення частково виконує вказану функцію;
– 2 – рішення повністю виконує вказану функцію.

Таблиця 3.1 – Порівняння існуючих рішень

Критерії	Рекомендаційні системи у складі готових сервісів	Netflix (дослідний прототип)	Бібліотеки (Crab, LensKit, RankSys)	Prediction IO
Універсальність системи	0	2	1	2
Можливість інтеграції	0	0	0	1
Наявність компонентів збору даних	2	0	0	0
Наявність на ринку	0	0	2	2

Універсальність забезпечується системою Netflix, бібліотеками (платформами) для розробки рекомендаційних систем, Prediction IO. Однак Netflix не є готовим програмним продуктом, має лише розроблений універсальний метод формування рекомендацій без збирання та аналізу даних. Бібліотеки не є модулями, що вбудовуються, потрібно писати код для того, щоб вони могли приймати дані; крім того, бібліотеки не є масштабованими. Порівняння показує, що Prediction IO найбільше задовольняє поставленим вимогам: його можна використовувати як модуль, що вбудовується в системи сторонніх розробників, проте він не інтегрується повністю з основним сервісом. Крім того, у ньому відсутні модулі збору даних та аналізу, які необхідні при створенні рекомендаційної системи (готові сервіси, що мають модуль рекомендацій, обов'язково мають ці кошти).

Таким чином, виявлені у існуючих рішень обмеження вказують на актуальність розробки універсальної рекомендаційної системи, яка здатна

вбудовуватися в сервіс клієнта, має компоненти збору та аналізу даних для забезпечення якості даних і результатів, що використовуються.

3.5 Обґрунтування мови програмування

Ключові веб-технології, що використовувались: HTML, CSS, PHP та JavaScript.

Мова гіпертекстової розмітки (HTML). Використовується для опису структури веб-сторінки. HTML дозволяє вказати, де на веб-сайті розміщуються текст, зображення та інші форми вмісту.

Каскадні таблиці стилів або CSS описують, як елементи мають відображатися на веб-сторінці. CSS дозволяє визначати такі стилі, як колір елементів на сторінці, їх розташування та спосіб відображення тексту на веб-сайті.

JavaScript. Ця мова сценаріїв гарантує більш динамічний веб-сайт. Будь-яка інтерактивна веб-сторінка, яку бачить користувач, ймовірно, певною мірою працює на JavaScript. Наприклад, контент, який оновлюється автоматично, та веб-сайти з картами використовують JavaScript для візуалізації контенту у зовнішньому інтерфейсі.

Бібліотека JavaScript, jQuery. Бібліотека JavaScript – це розширення мови JavaScript. Всі ці бібліотеки мають певний набір функцій, які мають допомогти розробникам створювати веб-сайти більш ефективно.

Візьмемо для прикладу jQuery. JQuery було розроблено, щоб спростити реалізацію JavaScript на веб-сайтах. JQuery абстрагує багато функцій, які можуть зустрітися в JavaScript, на користь більш простого синтаксису.

Використання jQuery дозволяє ефективно виконувати різноманітні завдання. Існує просте керування CSS на веб-сторінці та зміна елементів HTML. Також є додавання ефектів та анімації на сайт і потокова передача даних на веб-сторінку з використанням AJAX.

Системи контролю версій (VCS) використовуються для керування змінами у програмному проекті. Приклади цих систем включають Git та Mercurial.

VCS є важливими, тому що вони дозволяють розробникам бачити, як розвивається проект. Вони відстежують кожну зміну, внесену до кожного файлу в проекті. Поряд із кожним записом про зміну файлу міститься інформація про те, хто і коли його змінив.

Наявність цього запису означає, що легко побачити, як проект з'явився у певний момент історії. Це також спрощує повернення проекту до попередньої версії, якщо розробник припустився помилки.

Фреймворк переднього плану – це каркас, який постачається із попередньо написаним кодом, за допомогою якого є можливість створювати програму. До поширених фреймворків JavaScript відносяться React та Next.js.

Наприклад, React спрощує створення інтерактивної веб-програми. Він має функції, які дають змогу відображати потрібні частини веб-сторінки при зміні даних на веб-сайті. React також дозволяє розділити проект на компоненти, щоб користувач мав змогу зменшити повторення в кодовій базі.

Використання React може скоротити час роботи над проектом, тому що він готовий використовуватись прямо з коробки. Він також надає ряд функцій, які можна використовувати в процесі розробки програми.

Використання мови PHP при web-розробці дозволяє створити простий, ефективний, багатофункціональний, безпечний і найголовніше динамічний сайт.

Дизайн сайту легко реалізується за допомогою каскадних таблиць стилів – CSS, а поділ сторінок сайту на блоки у багато разів спрощує і економить час на його зміну.

3.6 Етапи розробки системи

Основні етапи розробки програмного забезпечення:

- аналіз вимог;
- проектування;
- розробка та програмування;
- тестування.

Аналіз вимог

Життєвий цикл розробки програмного забезпечення починається з фази аналізу, під час якої учасники обговорюють вимоги до кінцевого продукту. Метою цього етапу є визначення детальних вимог системи. Крім того, необхідно переконатися, що всі учасники правильно розуміють цілі та те, як кожна вимога може бути реалізована на практиці.

Залежно від обраного режиму розвитку метод визначення моменту переходу від однієї фази до іншої може відрізнятися. Наприклад, у Cascade або V-моделі етапи аналізу вимог фіксуються в документі – Специфікації вимог до програмного забезпечення (SRS), проектування якого необхідно завершити перед переходом до наступного етапу.

Таким чином, цей етап включає збір вимог до програмного забезпечення, що розробляється, систематизацію, документацію, аналіз, а також виявлення та вирішення конфліктів.

Проектування. Під час фази проектування (також відомої як фаза проектування та архітектури) проектується система високого рівня на основі вимог.

На цьому етапі спрощення процесу візуалізації використовуються як так звані:

– Блок-схеми;

– ER-діаграми;

– UML-діаграми.

Розробка та програмування. Після того як вимоги і дизайн продукту затверджені, відбувається перехід до наступної стадії життєвого циклу – безпосередньої розробки. Тут починається написання програмістами коду програми відповідно до визначених раніше вимог.

Програмування передбачає чотири основні стадії:

– розробка алгоритмів – фактично, створення логіки роботи програми;

– написання джерела коду;

– компіляція – перетворення в машинний код;

– тестування та відладка – мова йде, головним чином, про юніт-тестування.

Тестування. Під час тестування відбувається пошук дефектів у програмному забезпеченні і порівнюють описану у вимогах поведінку системи з реальною. Тестування повторюється до тих пір, поки не будуть досягнуті критерії його закінчення.

3.7 Структура проекту

Для побудови ефективної і надійної системи, яка буде зручна в супроводі, підтримуванні і тестуванні, необхідно розуміти, що така система повинна володіти внутрішніми характеристиками, грамотно спроектованими для підтримки у подальшій розробці, і зовнішніми характеристиками, які орієнтовані на кінцевих користувачів і які повинні представляти максимальну простоту використання продукту.

Основою методики створення програмного модулю персоналізованого переліку товарів та послуг в рекомендаційних системах є Model-view-controller. Схема патерна наведена на рисунку 3.1. Для вирішення завдання персоналізованого переліку товарів та послуг в рекомендаційних системах пропонується використовувати схему (рисунок 3.2).

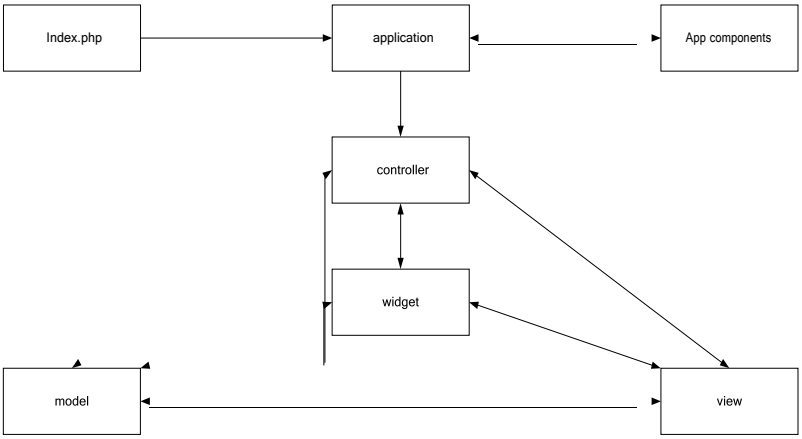


Рисунок 3.1 – Схема Model-view-controller

Рисунок 3.2 – Схема реалізації персоналізованого переліку товарів та послуг в рекомендаційних системах

Попередню модель програмного модулю персоналізованого переліку товарів та послуг в рекомендаційних системах можна представити у вигляді функціональних схем претендентів.

Головна діаграма прецедентів, що описує зовнішній кордон програмного модулю персоналізованого переліку товарів та послуг в рекомендаційних системах (рисунок 3.3).

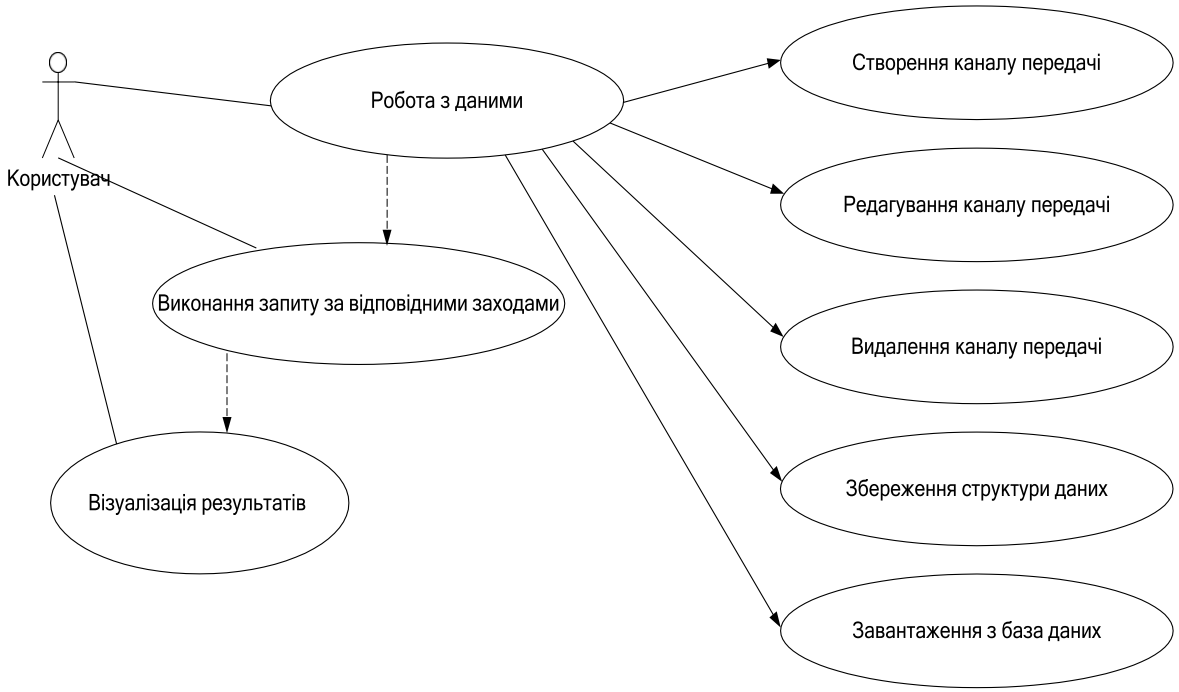


Рисунок 3.3 – Діаграма прецедентів програмного модулю персоналізованого переліку товарів та послуг в рекомендаційних системах

3.8 Ризики проекту

Будь-який проект з розробки додатка пов'язаний із певними ризиками. Ризики можуть змінюватись від проекту до проекту, їх завжди слід враховувати при розробці, але в цілому їх можна розділити на шість основних.

Неправильна оцінка. Коли складається оцінка проекту, буває, щовона не виправдовує очікувань. Команда може вибрати тривалість ітерації проекту, стек технологій та інші фактори. Між клієнтом та командою часто виникають розбіжності, що призводить до збільшення тривалості завдання, витрат, через які у клієнта закінчуються гроші, і він не може завершити проект.

Як мінімізувати цей ризик:

- виконувати лише найважливіші завдання;
- додати час розробникам для вивчення та зниження ризиків у частинах нового проекту;
- додати передбачуваний період для команди розробників протягом тижня для завдання, що виходить за межі проекту;
- в управлінні проектами конус невизначеності визначає розвиток невизначеності у разі під час проекту. На початку проектів, мало що відомо про продукт або результати роботи, тому оцінки схильні до великої невизначеності.

Зміни. Зміни відбуваються, якщо масштаб ітерації змінюється після затвердження дедлайну. Замовники часто хочуть змінити обсяг проекту і це створює великий ризик, не дозволяючи розробникам слідувати початковому ТЗ та графіку.

Як мінімізувати цей ризик:

- невеликі, прості в управлінні ітерації з гнучкої методології Agile дозволяють аналізувати та змінювати розмір проекту;
- при оцінці проекту взяти кілька додаткових тижнів для непередбачених ситуацій.

Залучення користувачів. Варто враховувати відгуки користувачів. Ці стратегії спрощують використання з допомогою гнучкої розробки.

Як мінімізувати цей ризик:

- користувальницьке тестування та зворотний зв'язок;
- фокус-групи;

- часті релізи;
- бета-тестування.

Код низької якості. Низька якість коду – одна з найпоширеніших проблем у розробці та одна з найбільших проблем клієнта. Найчастіше замовник не розуміє код і не може визначити його якість. До завершення розробки може з'ясуватися, що програма має неякісно розроблений код, проблеми в роботі та відсутність грамотного тестування.

Як мінімізувати цей ризик:

- розробникам важливо дотримуватися розроблених стандартів коду;
- клієнт може найняти менеджера проекту або технічного директора, який зможе перевіряти якість коду та контролювати команду розробників;
- дотримання системи стандартів якості коду (англ. Clear Coding Standards and Guidelines);
- тестування після кожної ітерації коду;
- перш ніж розпочати роботу з компанією-розробником, клієнт може запросити аналогічний проект та ознайомитися зі стандартами коду компанії.

Низька взаємодія з клієнтом. Щоб команда розробників закінчила вчасно проект, необхідно досить часто спілкуватися з клієнтом або менеджером проекту для уточнення деталей проекту. Якщо цього не робити, то є ризик непорозуміння з обох сторін та збільшення тривалості ітерацій.

Як мінімізувати цей ризик:

- необхідно чітко узгодити, коли клієнт та замовник зможуть провести Користувальницьке Тестування (User Acceptance Testing);
- важливо обговорити допустимий час відповіді, якщо будь-яка із сторін має проблему або питання щодо проекту;
- чіткий вибір цілей та пріоритетів проекту.

Недостатня кількість людей. Різних причин членам команди доводиться залишати проект. Через це робота припиняється доти, доки не буде знайдено нового члена команди, що збільшує терміни виконання проекту.

Як мінімізувати цей ризик:

- документування всього процесу розробки;
- хороша внутрішня кадрова система, що дозволяє у разі виникнення такої ситуації швидко замінити члена команди;
- менеджер проекту повинен часто контролювати графік робочого навантаження своєї команди, щоб оперативно зреагувати зміни.

Менеджер проекту відповідає за всі завдання, пов'язані із проектом. Бізнес-аналітик, що займається визначенням та аналізом вимог проекту. Разом вони можуть допомогти створювати успішні продукти та мінімізувати ризики розробки програмного забезпечення.

Головне завдання менеджера проекту під час розробки довести ідею клієнта до результату, використовуючи наявні ресурси. Йому необхідно створити план розвитку, організувати команду, налагодити процес роботи над проектом, дати зворотний зв'язок, ліквідувати перешкоди для команд, контролювати якість та своєчасність виконання завдань:

- розробка нового продукту чи нових функцій. Менеджер проекту організовує зустріч із технічним архітектором та розробниками, обговорює з ними майбутні завдання, допомагає у складання ТЗ;
- планування. Важливо враховувати всі фактори, що впливають на хід розробки, включаючи кваліфікацію співробітників та пов'язані зними ризики, залежність від сторонніх сервісів та виправлення помилок;
- контроль. Менеджеру проекту необхідно щодня відстежувати проектні ітерації та контролювати весь процес розробки, щоб оперативно реагувати на проблеми, що виникають;

- спілкування із замовником. Це важливий пункт при розробці проекту, оскільки проблеми можуть відбуватися ситуативно та важливо підтримувати комунікацію з клієнтом, повідомляти про зміни на проекті.

Бізнес-аналітик – це саме той фахівець на проекті, який має визначити бізнес-проблеми клієнта та знайти найефективніше рішення. На всіх етапах розробки програмного забезпечення бізнес-аналітик аналізує вимоги і є сполучною ланкою між командою розробників та клієнтом.Нижче наведено кілька основних функцій, які виконує бізнес-аналітик на проекті:

- виявлення вимог замовника до проекту;
- документування вимог до майбутнього проекту;
- створення прототипів вимог клієнтів, мозковий штурм з клієнтом та командою розробників, проведення тестів та анкетування для кращого розуміння та аналізу вимог;
- виявлення слабких сторін проекту з урахуванням власного аналізу. Пропозиція способів оптимізації процесів та вирішення можливих проблем щодо проекту;
- написання технічного завдання;
- оптимізація вимог до проекту;
- передача оптимізованих вимог від замовника до команди розробників.

Головний ризик розробки ПЗ. Для компанії – перевитрата бюджету та людських ресурсів. Отримання більшої кількості додаткових вимог від клієнта в процесі розробки та порушення специфікації вимог до програмного забезпечення.

Для клієнтів:

- вихід за межі бюджету;
- одержання низькоякісного продукту.

3.9 Практична реалізація рекомендаційної системи персоналізованого переліку товарів та послуг

Загальна схема роботи з рекомендаційною системою в рамках пропонованого підходу (рисунок 3.4). Пропонується інтегрувати систему з основним сервісом компанії, що пропонує товари чи послуги, налаштувавши її на відповідну область через створення моделі предметної області на основі даних, одержуваних від основного сервісу певному форматі.

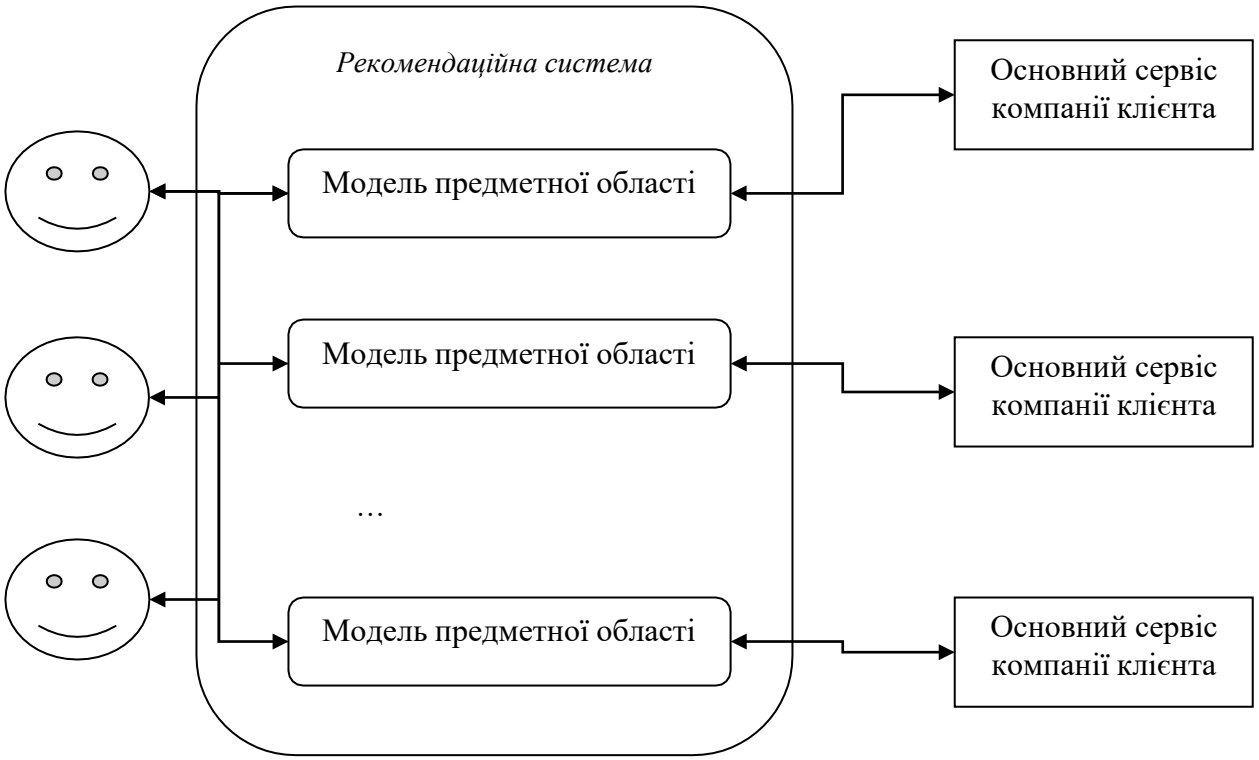


Рисунок 3.4 – Схема інтеграції рекомендаційної системи з основними сервісами компаній-клієнтів

Універсальна рекомендаційна система включає дві підсистеми, що взаємодіють з основним сервісом: власне рекомендаційна система, яка забезпечує збір даних та вироблення рекомендацій; модуль обробки та інтелектуального аналізу даних (рисунок 3.5).

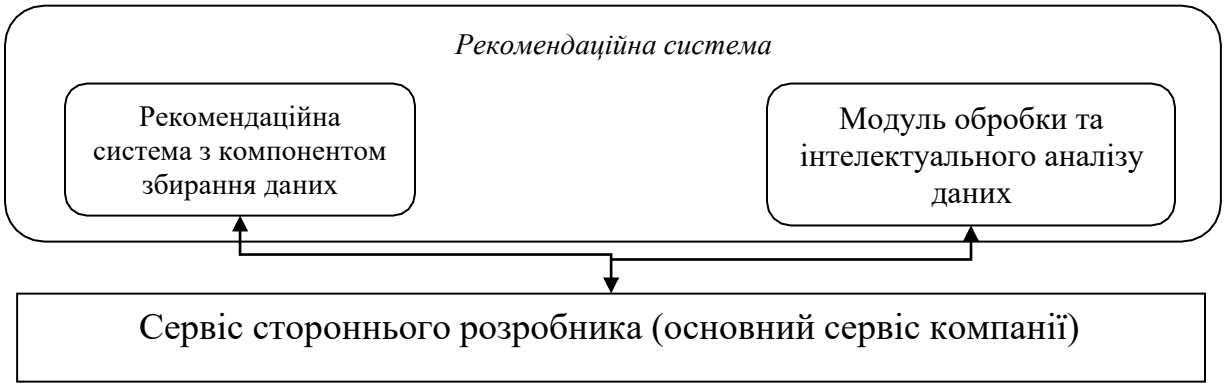


Рисунок 3.5 – Підсистеми універсальної рекомендаційної системи

Ці дві підсистеми можуть працювати незалежно одна від одної. Однак доповнення рекомендаційної системи засобами аналізу даних суттєво підвищує якість її роботи.

Під час проектування архітектури системи, яка взаємодіє з сервісом та користувачами, було обрано мікросервісний підхід. Мікросервісний підхід – це особливий спосіб розробки програмних систем, який знаходить широке застосування останніми роками. Мікросервісна архітектура передбачає розробку програмних додатків як набору незалежнихневеликих сервісів, що розгортаються, кожен з яких запускає унікальний процес і забезпечує комунікацію за допомогою чітко визначеного, легкого механізму. Завдяки гнучкості та масштабованості ці архітектурні рішення є найбільш підходящими, коли потрібно включити підтримку для цілого ряду платформ та пристроїв різних типів.

Основні переваги застосування мікросервісів:

– архітектура надає розробникам свободу самостійно розробляти та розгортати служби;

– мікросервіс може бути розроблений невеликою командою;

– код для різних служб може бути написаний різними мовами;

– проста інтеграція та автоматичне розгортання (з використанням інструментів безперервної інтеграції з відкритим вихідним кодом, таких як Jenkins, Hudson тощо);

– легкість розуміння та модифікації коду;

– можливість використання розробниками нових технологій;

– простота масштабування та інтеграції зі сторонніми сервісами.

Але цей підхід має свої недоліки – розробникам доводиться мати справу зі збільшенням складності робіт при проектуванні, реалізації та супроводі розподіленої системи, зокрема:

– розподілена архітектура привносить додаткову складність, оскільки розробникам доводиться балансувати навантаження, підтримувати стійкість до відмови і т.д.;

– розробники повинні докласти додаткових зусиль для впровадження та підтримки механізмів взаємодії між сервісами, обробки різних форматів повідомлень, реалізації більш дорогих віддалених викликів, повинні використовувати грубіші віддалені API-інтерфейси;

– завдяки розподіленому розгортанню тестування може стати складним та стомлюючим;

– при «перерозподілі обов'язків» між компонентами можуть виникнути проблеми, неузгоджена робота може призвести до дублювання зусиль;

– обробка функцій, що охоплюють більше однієї служби, потребує взаємодії та співробітництва між різними командами;

– збільшення кількості послуг може призвести до інформаційних бар'єрів;

– коли кількість послуг збільшується, інтеграція та управління системою загалом може стати складним;

– архітектура зазвичай призводить до збільшення споживання пам'яті та ін.

Таким чином, етап проектування архітектури, вибору технологій для реалізації компонентів системи стає найскладнішим танайвідповідальнішим етапом розробки.

У системі виділені такі завдання, реалізовані як окремі мікросервіси: збір даних; формування оцінки об'єктів; видача рекомендацій.

Для службових обчислень обрано архітектуру, засновану на кластерах, що забезпечують прозорість системи. В результаті цього підвищується надійність, збалансованість та продуктивність платформи.

Загальна архітектура рекомендаційної системи (рисунок 3.6).

У якості основного кластеру для зберігання та обробки даних використовується програмна платформа Hadoop – система, що складається з безлічі різних компонентів, що у сукупності створюють єдину платформу. Ця платформа має відкритий вихідний код і призначена для зберігання та обробки даних, які надто великі для одного конкретного пристрою або сервера. Сила Hadoop полягає в її здатності масштабуватись по тисячах товарних серверів, які не поділяють пам'ять або дисковий простір. Hadoop делегує завдання через робочі сервери, які називаються «робочими вузлами» або «підлеглими вузлами», суттєво використовуючи можливості кожного пристрою та запускаючи їх одночасно. Для цілей роботи використовується розподілена файлова система Hadoop (HDFS) – масштабована файлова система, яка розподіляє та зберігає дані на всіх машинах у кластері Hadoop (групі серверів), що дозволяє зберігати величезні файли.

На кластері Hadoop як база даних використовується Apache Hbase. Дана база даних є колонковою, працює на HDFS. Вона масштабується зберігання великих обсягів розріджених даних. У схемі Big Data вона відповідає за категорії сховища і є альтернативним або додатковим варіантом сховища х.

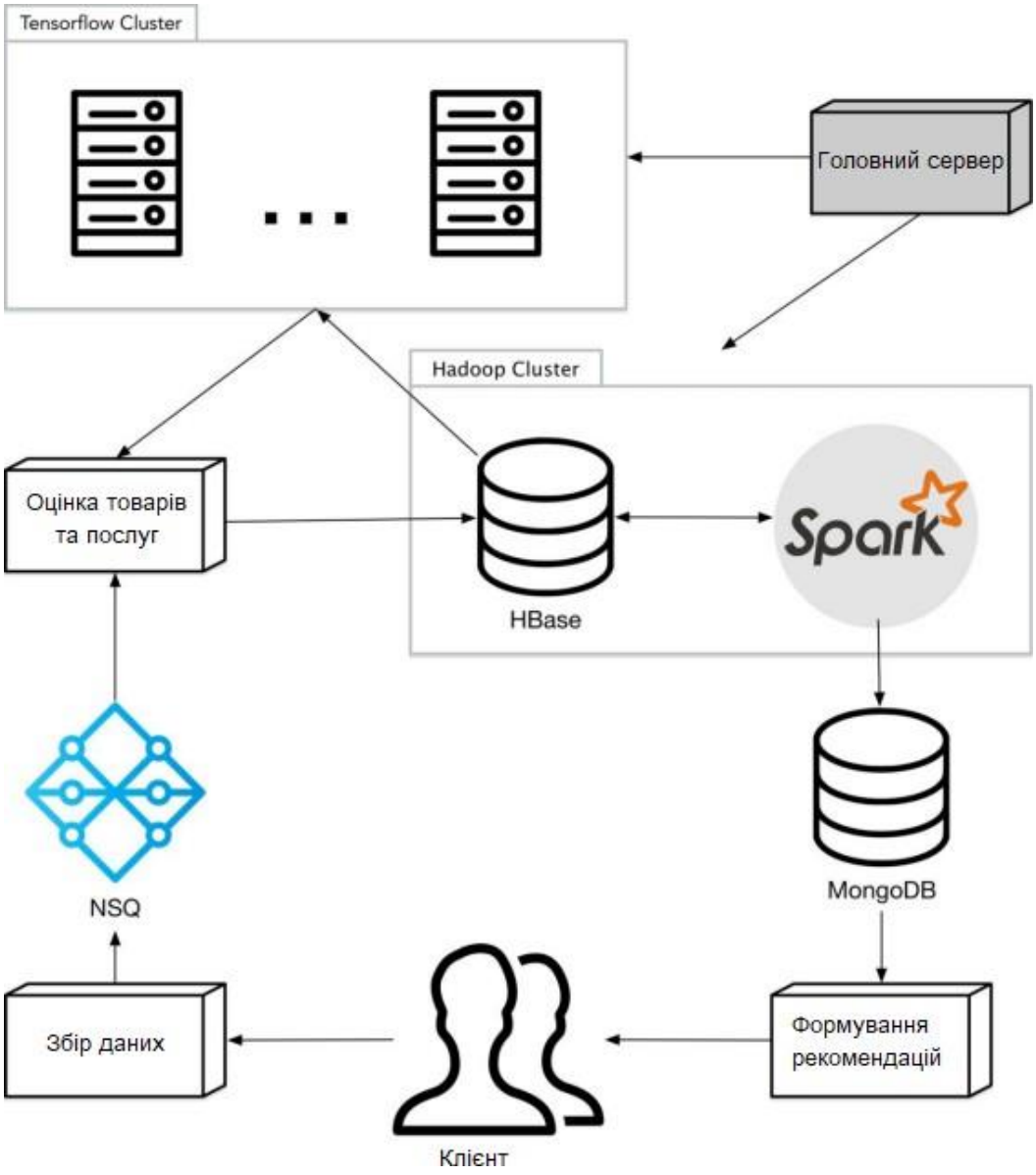


Рисунок 3.6 – Спрощена архітектура рекомендаційної системи

Для обробки даних на кластері використовується Apache Spark. Spark – це середовище виконання, яке працює з файловою системою, щоб розподіляти дані по кластеру та обробляти ці дані паралельно. Spark також приймає набір інструкцій із програми, написаної розробником. Це середовище підтримує різні мови програмування (Java, Python та Scala). Цей кластер використовує бібліотеку MLlib.

Для навчання нейронної мережі, що формує оцінки об'єктів за даними користувачів, використовується кластер TensorFlow. TensorFlow –

одна з найпопулярніших бібліотек для машинного навчання. Ця бібліотека використовує тензори та графи як основу. Кластер TensorFlow – це кілька комп'ютерів, на яких запущено сервер TensorFlow, об'єднаних майстер-сервером.

Первинне завдання сервісу збору даних – отримати дані про переглянуті об'єкти від користувачів. Для очікування даних від клієнта потрібний окремий сервіс. Після надсилання даних користувача сервер збору даних ці дані потрапляють на сервер NSQ – розподіленої платформи обміну повідомленнями у часі. Ця платформа підтримує розподілені та децентралізовані топології, забезпечуючи відмовостійкість та високу доступність у поєднанні з надійною гарантією доставки повідомлень, а також горизонтально масштабується. Вбудоване виявлення полегшує додавання вузлів у кластер, підтримує доставку повідомлень. Від NSQ дані розподіляються на мікросервіси формування оцінки об'єктів, де є навчена модель нейронної мережі, яка формує оцінку, якщо вона відсутня, і накопичує пакет повідомлень, щоб надіслати інформацію на сервер бази даних.

Архітектура сервісу формування пропозицій включає трикомпоненти: головний сервер, кластер Hadoop та компонент видачірекомендацій. Сервіс формування пропозицій починає свою роботу у Hadoop-кластері, де за сигналом головного сервера відбувається формування пропозицій для користувачів на платформі Apache Spark. Сформовані пропозиції передаються на сервер баз даних MongoDB, де відбувається реплікація даних інші сервери MongoDB до роботи з мікросервісом видачі рекомендацій. Реплікація – процес копіювання та синхронізації даних на кількох серверах – забезпечує надмірність підвищення доступності даних, відмовостійкості.

При необхідності клієнт (додаток основного сервісу) надсилає запит на отримання пропозицій для користувача. Логіка обробки наданих рекомендацій та видача їх кінцевому користувачеві реалізується

розробниками програми самостійно. Таким чином, система працює з двома СУБД: для збору даних – HBase; для надання рекомендацій – MongoDB. HBase забезпечує швидкий доступ до всієї таблиці для їх обробки. MongoDB має досить велике значення RPS (request per second), що дозволяє обробити велику кількість запитів, а також легко масштабується для потреб мікросервісів.

Обмін даними між сервером та клієнтом необхідний у двох випадках: при зборі даних про переглянуті об'єкти та при видачі рекомендацій. Весь обмін даними передбачає використання REST API.

Архітектура модуля аналізу даних включає мікросервіси роботи з вихідними даними, для пошуку дублікатів, для групування об'єктів та пошуку асоціативних правил (рисунок. 3.7). Кожен мікросервіс має доступ лише до бази даних. Оскільки мікросервісам, які реалізують основні завдання, необхідний доступ до даних, завантажених у систему клієнтом, вводиться сервіс для завантаження та доступу до загальних даних. Мікросервіси пошуку дублікатів, угруповання та пошуку асоціативних правил на запит отримують необхідні дані від мікросервісу для роботи з даними для їх подальшої обробки. Існує також комунікація між мікросервісом де-дуплікації та мікросервісом угруповання, оскільки процес пошуку дублікатів відбувається на попередньо кластеризованих даних. Мікросервіс угруповання на запит повертає згруповані дані сервісу пошуку дублікатів. Комунікація між усіма мікросервісами здійснюється за допомогою виклику віддалених процедур (RPC). В даному випадку цей метод реалізується за допомогою технології gRPC від Google, одним із застосувань якої є забезпечення зв'язку між мікросервісами. Як інструмент опису типів даних та серіалізації gRPC використовує Protobuf, відмінною особливістю якого є продуктивність навіть при великих обсягах даних. Дана технологія підтримує роботу з великою кількістю мов програмування, а також працює швидше ніж REST, тому що як протокол передачі структурованих даних використовує Protobuf.

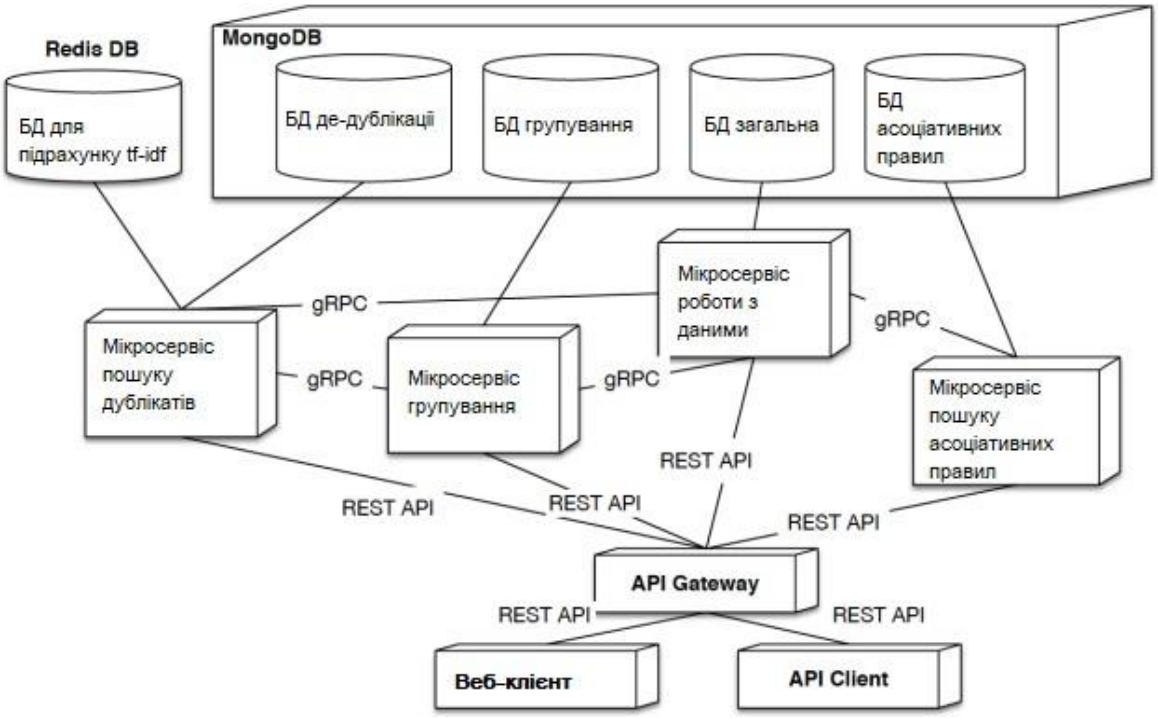


Рисунок 3.7 – Спрощена архітектура підсистеми аналізу даних

Кожен мікросервіс керує своєю базою даних: сервіс де-дуплікаціїмає доступ до бази даних Redis для зберігання даних за підрахунком tf-idf (необхідний для порівняння рядкових полів) та до бази даних де- дуплікації, де зберігаються результати роботи алгоритмів пошуку дублікатів; сервіси угруповання, пошук ка асоціативних правил також має свої бази даних, де зберігаються результати їх роботи, і на запит від клієнта вони видають необхідну інформацію, ґрунтуючись на цих даних. Мікросервіс для роботи з даними має доступ лише до загальної бази даних,куди він заносить дані на запит від клієнта, або забирає на запит від мікросервісів, що виступають у ролі клієнтів. У загальній базі даних зберігаються вихідні дані клієнта, які помістив у систему для аналізу. Таким чином, реалізовані всі основні умови мікросервісної архітектури: кожен сервіс виконує своє завдання, є окремим вузлом, має власний простір даних.

Для забезпечення взаємодії даних модулів із клієнтами необхідний сервер API Gateway, який перенаправляє запити клієнтів на необхідні мікросервіси. Це дозволяє створити єдиний API для доступу до мікросервісів. Клієнти не використовують адреси всіх вузлів мікросервісів безпосередньо, оскільки це знижує масштабованість. Для реалізації цього проксі-сервера вибрано платформу nginx, яка дозволяє також балансувати навантаження між вузлами. Проксі-сервер взаємодіє із мікросервісами за допомогою REST API.

Система має веб-клієнт, також надає можливість створення власного клієнта (має API для звернення до функцій системи з програмного коду). Клієнти зв'язуються із проксі-сервером за допомогою REST API.

Автоматичне угруповання об'єктів – завдання, яке вирішується методом кластеризації в машинному навчанні. Оскільки в рамках даної роботи розробляється дослідницький прототип модуля аналізу, для угруповання об'єктів вибрано найпростіший і найдоступніший алгоритм k-means. Мінус алгоритму k-means у тому, що необхідно підбирати кількість кластерів вручну, проте є підходи, які можуть допомогти оцінити кількість кластерів. У цій роботі застосовано метод Affinity propagation [24], також реалізований у бібліотеці scikit learn. Для отримання більш точних результатів надалі планується включити інші, ефективніші методи кластеризації задля забезпечення ще більшої адаптованості системи.

Однією з основних умов «правильності» рекомендацій є забезпечення унікальності даних. Кожна ліквідована копія об'єкта може сприяти збільшенню відсотка точності для формування пропозицій. У цій роботі пропонується такий алгоритм пошуку дублікатів, який з урахуванням отриманої функції подібності між парою об'єктів знаходить кластери дублікатів. Функція подібності повинна визначати ймовірність того, що два об'єкти є дублікатами, одержуючи на вхід різницю між ними. Для створення функції подібності можна використовувати методи машинного навчання з учителем. Найбільш універсальним методом, який

зарекомендував себе практично, є gradient boosting decision trees [25]. Використання цього в даній роботі не відкидає інші, можливо, ефективніші рішення. Для того щоб модуль аналізу був якомога універсальнішим і міг працювати з різного виду даними, надалі необхідно додавати інші методи вирішення задачі пошуку функції подібності.

Пошук закономірностей між подіями у базі даних (наприклад аналіз купівельних кошиків та пошук продуктів, що купуються спільно) в аналізі даних ведеться за допомогою асоціативних правил [26]. Пошук об'єктів, що доповнюють (часто зустрічаються комбінацій у даних) також можна здійснити за допомогою асоціативних правил. В основі асоціативних правил лежить поняття транзакції – багато подій, що відбулося одночасно. Масштабований алгоритм асоціативних правил є алгоритм Apriori [27], [28]. На основі реалізації цих засобів можна формувати пропозиції при виборі користувачем чергового об'єкта при складанні ним деяких комбінацій об'єктів (товарів, послуг тощо), а також проводити аналіз та вирішувати інші завдання, які специфічні для кожної області.

3.10 Результат роботи рекомендаційної системи

На основі локального серверу проведемо тестування розробленої системи, результати наведемо на рисунках 3.8-3.12.

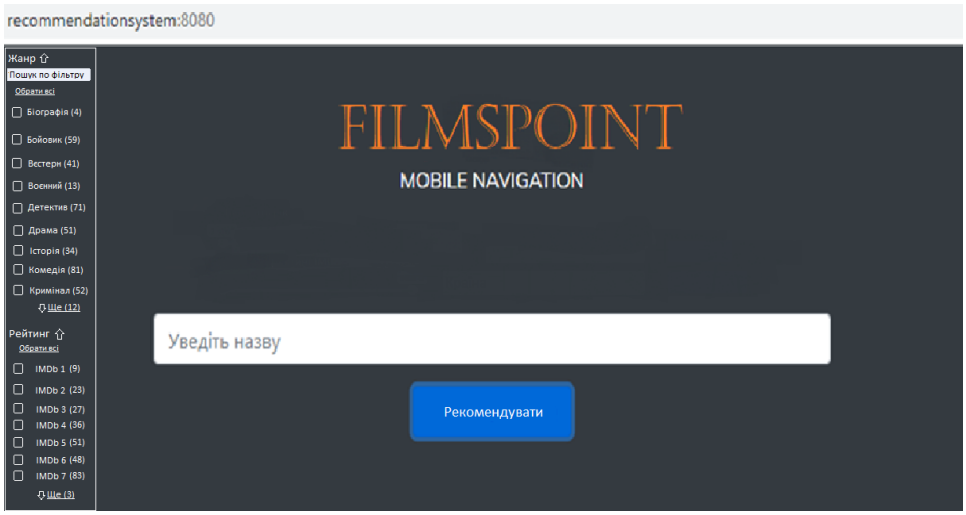


Рисунок 3.8 – Результат роботи рекомендаційної системи

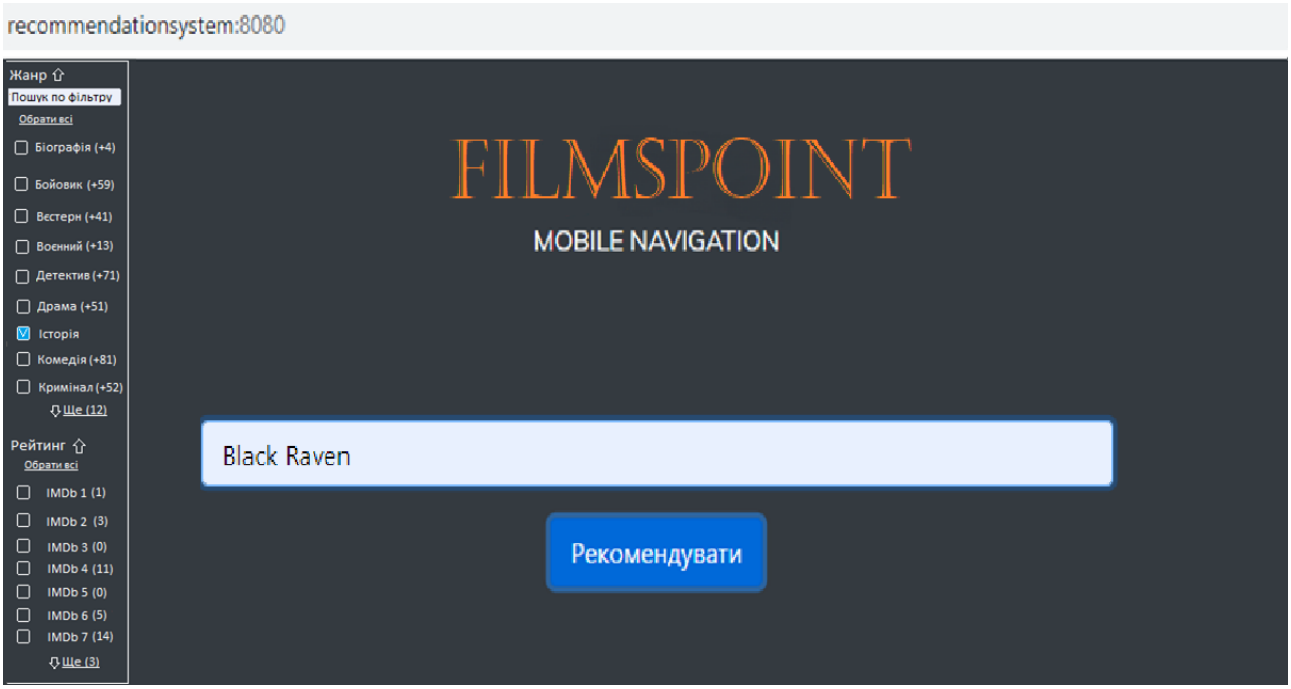


Рисунок 3.9 – Результат роботи рекомендаційної системи

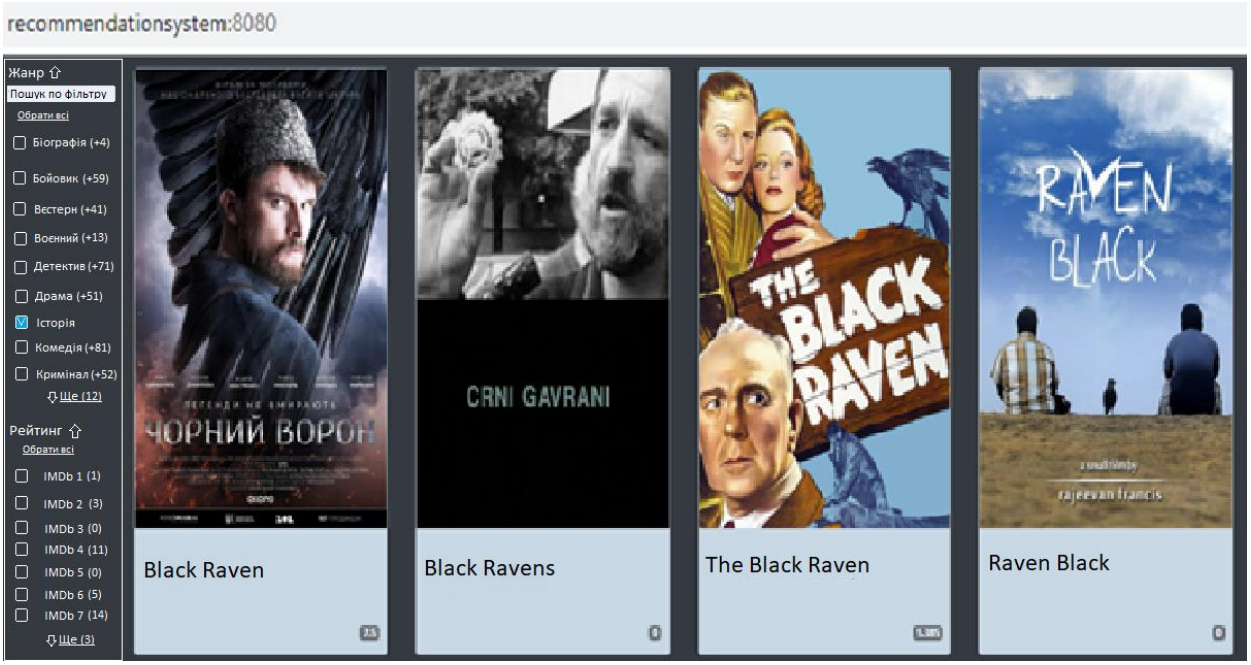


Рисунок 3.10 – Результат роботи рекомендаційної системи

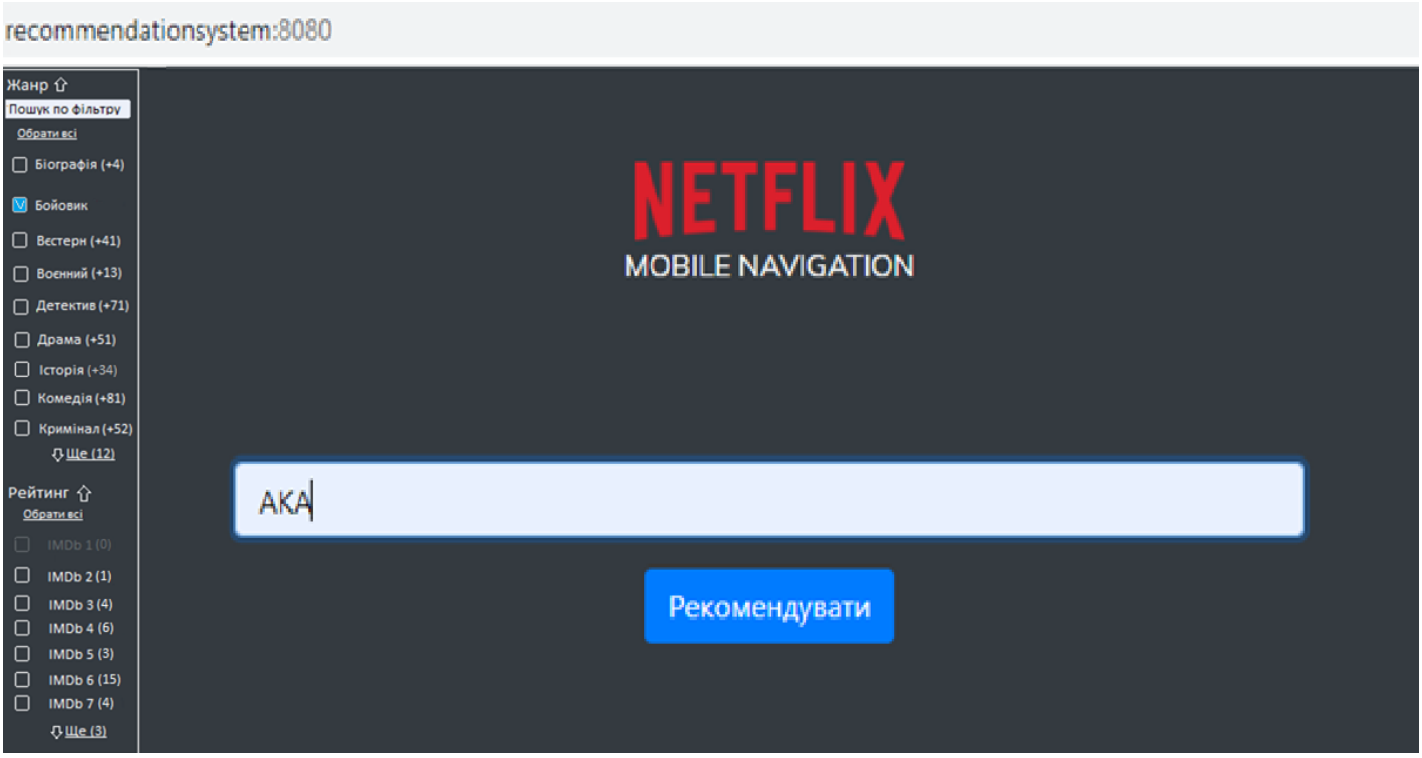


Рисунок 3.11 – Результат роботи рекомендаційної системи

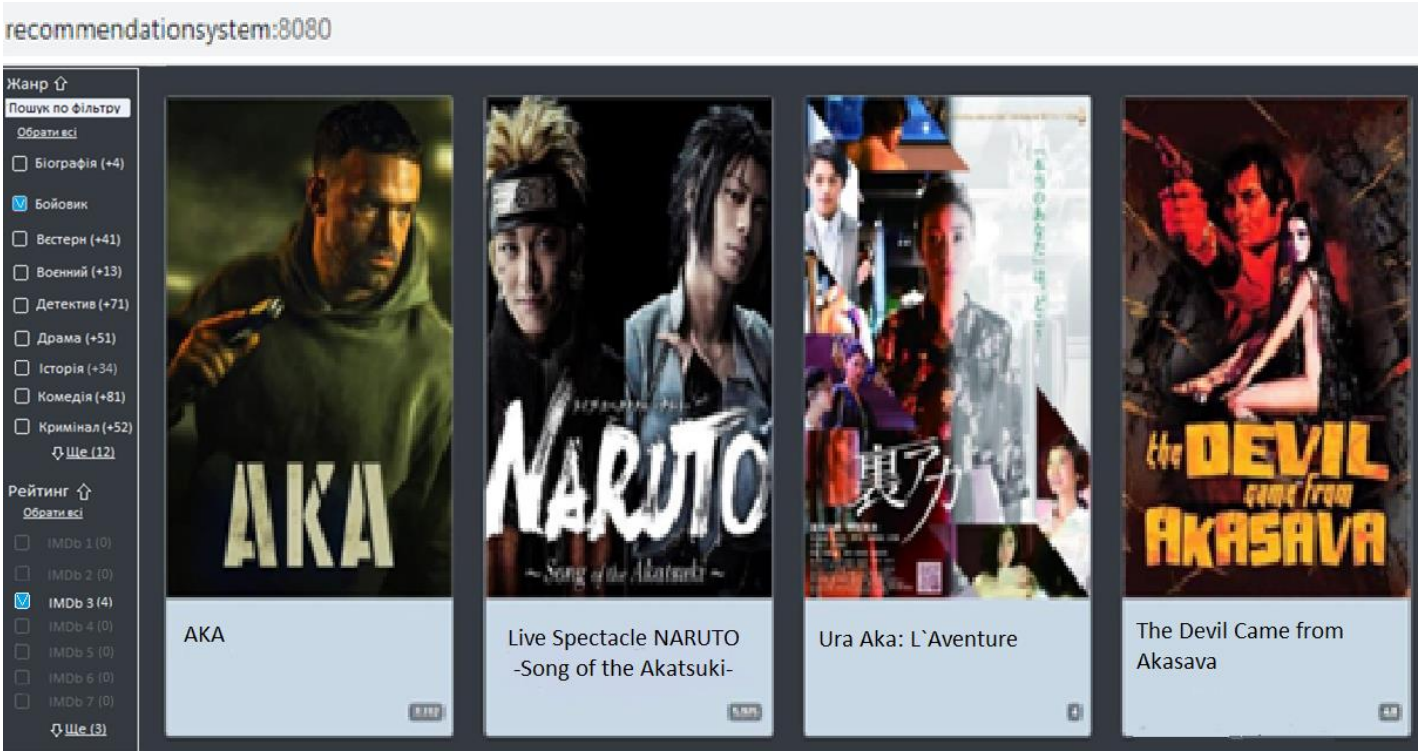


Рисунок 3.12 – Результат роботи рекомендаційної системи

У вибраних даних представлено 100000 оцінок за 1682 фільмамивід 943 різних користувачів. Причому кожен користувач цих данихпоставив оцінки не менше, ніж 20 фільмам.

Дані є 100000 рядків виду: user id | item id | rating | timestamp.

Тобто в кожному рядку записано: номер користувача, номер фільму, оцінка, яку користувач поставив фільму, і час, коли це сталося (рисунок 3.19).

Точність та повнота. Для порівняння методів рекомендацій спочатку було прийнято рішення скористатися стандартними заходами точності (precision) та повноти (recall):

$$precision = \frac{\left|\{relevant\ movies\} \cap \{retrieved\ movies\}\right|}{\left|\{retrieved\ movies\}\right|} \quad , (3.1)$$

$$recall = \frac{\left|\{relevant\ movies\} \cap \{retrieved\ movies\}\right|}{\left|\{relevant\ movies\}\right|} \quad . (3.2)$$

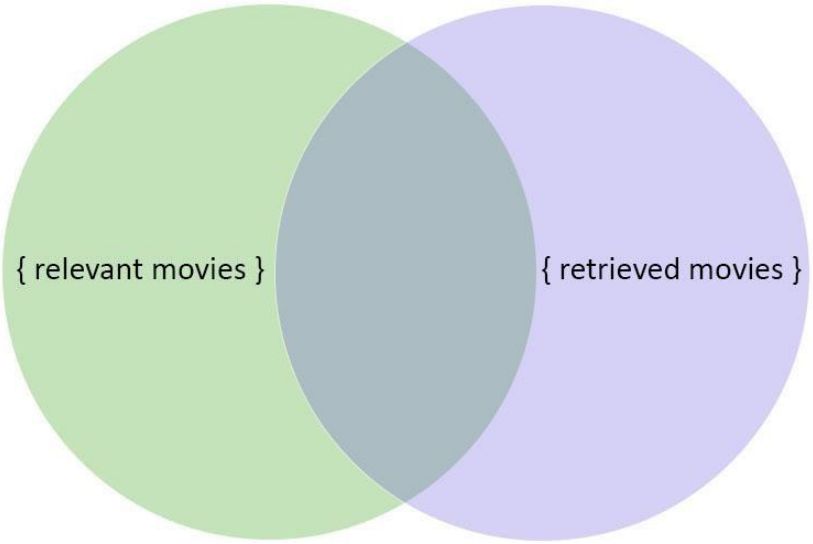


Рисунок 3.15 – Безліч рекомендованих та релевантних фільмів

Але вже в ході перших проведених експериментів стало видно, що точність методу запропонованого у даній кваліфікаційній роботі виходить у порівнянні зі Slope One набагато меншою. Причина була в тому, що запропонований гібридний метод кожному користувачеві в середньому видавав на порядок більше рекомендацій, ніж Slope One, і більшість цих рекомендованих фільмів користувач не дивився, і отже не оцінював.

Стало зрозуміло, що так оцінювати методи не зовсім коректно. Справді, якщо метод порекомендував фільм, а користувач його ще не оцінив, ми не можемо точно знати, сподобається він йому чи ні насправді.

Тоді для порівняння методів рекомендацій Slope One та запропонованого у даній кваліфікаційній роботі була розроблена модифікація точності (precision) та повноти (recall), які будемо називати скоригованими точністю та повнотою.

Для кожного конкретного користувача безліч переглянутих ним фільмів розбивається на дві множини: безліч фільмів, на яких навчаються алгоритми (train movies), та безліч контрольних фільмів (test movies). Перша множина складалася з 80% переглянутих фільмів, а друга відповідно з 20%. Причому фільми з першої множини були оцінені

користувачем раніше, ніж фільми з другої множини. Для такого розбиття всі оцінки користувача були впорядковані за часом проставлення і потім поділені.

Отже, було прийнято рішення скориговану точність та повноту розраховувати за формулами, зазначеними нижче:

$$precision = \frac{\left|\left\{\begin{matrix} relevant \\ movies \end{matrix}\right\} \cap \left\{\begin{matrix} retrieved \\ movies \end{matrix}\right\} \cap \left\{\begin{matrix} test \\ movies \end{matrix}\right\}\right|}{\left|\left\{\begin{matrix} retrieved \\ movies \end{matrix}\right\} \cap \left\{\begin{matrix} test \\ movies \end{matrix}\right\}\right|}, \quad (3.3)$$

$$recall = \frac{\left|\left\{\begin{matrix} relevant \\ movies \end{matrix}\right\} \cap \left\{\begin{matrix} retrieved \\ movies \end{matrix}\right\} \cap \left\{\begin{matrix} test \\ movies \end{matrix}\right\}\right|}{\left|\left\{\begin{matrix} relevant \\ movies \end{matrix}\right\} \cap \left\{\begin{matrix} test \\ movies \end{matrix}\right\}\right|}. \quad (3.4)$$

Такі оцінки точності та повноти дозволили уникнути описану вище проблему невизначеності, оскільки ми не знаємо, як саме користувач оцінить рекомендований фільм. У реальному житті ми могли б запитати користувача, чи правильна рекомендація, але в нашому випадку ми цього дозволити не можемо. Іншими словами, при оцінці точності та повноти даним методом ми вважаємо, що в даний конкретний момент для заданого користувача фільмів крім train movies і test movies просто не існує (рисунок 3.16).

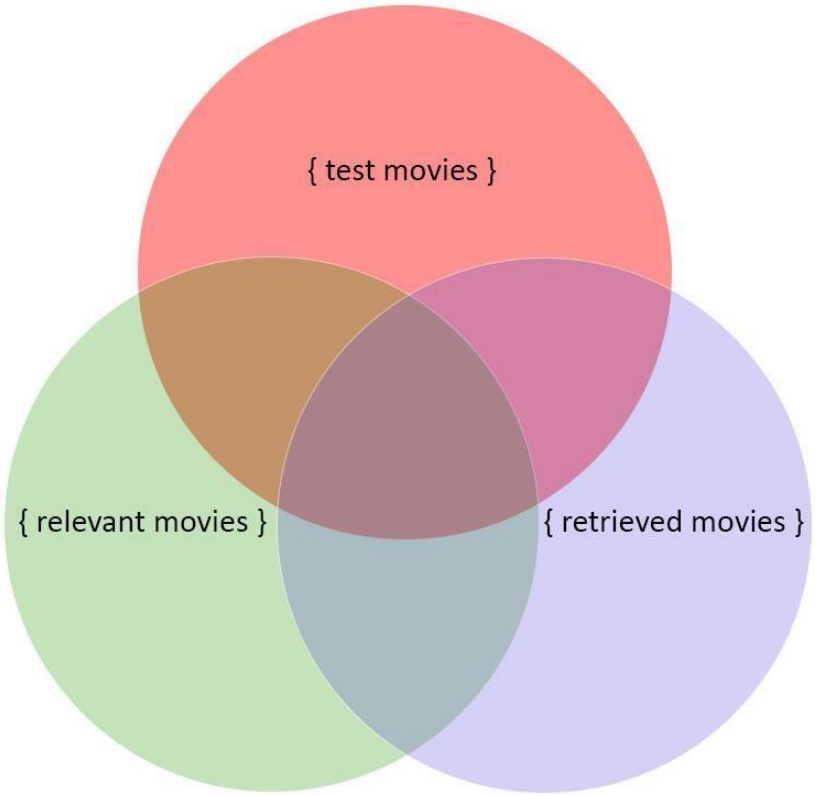


Рисунок 3.16 – Безліч рекомендованих, релевантних та контрольних фільмів

Програмні засоби. Для виконання порівняльного аналізу методів Slope One та запропонованого у даній кваліфікаційній роботі мовою програмування MATLAB було написано кілька програм.

data_load.m – ця невелика програма завантажує вихідні дані і з них створює дві розріджені (sparse) матриці: матрицю оцінок і матрицю часу. Ці матриці зберігаються як глобальні змінні та використовуються під час запуску інших програм.

main.m – ця програма розбиває випадковим чином вихідну кількість користувачів на два: train_users (80%) та test_users (20%).

Потім для кожного користувача з test_users виконуються такі кроки: безліч переглянутих ним фільмів розбивається на безліч фільмів, на яких навчатимуться алгоритми (train movies), і безліч контрольних фільмів (test

movies) таким чином, як описано в роботі. Далі запускаються рекомендаційні алгоритми запропонований у даній кваліфікаційній роботі та Slope One з різними параметрами. І наприкінці обчислюються точність і повнота як і, як описано вище за формулами (3.3) і (3.4).

raps.m – ця функція є реалізацією алгоритму запропонованого у даній кваліфікаційній роботі.

slope_one.m – ця функція є реалізацією алгоритму Slope One.

precision_and_recall.m – ця функція обчислює скориговану точність та повноту за формулами за формулами (3.3) та (3.4).

Необхідно зауважити, що перед усіма написаними програмами ставилося суто дослідницьке завдання. Надалі вони будуть оптимізовані та вдосконалені.

3.11 Експерименти та оцінка результатів

Як говорилося вище, для порівняння алгоритмів запропонованого у даній кваліфікаційній роботі і Slope One скористалися формулами точності.

Всі користувачі були випадково поділені на дві множини: на 80% користувачів відбувалося навчання алгоритмів, а на 20% – тестування. Причому, для кожного користувача з контрольної вибірки безліч переглянутих ним фільмів також поділялося на два: 80% раніше переглянутих фільмів – для навчання алгоритмів, та 20% останніх переглянутих – для тестування.

Було проведено три серії тестів.

Коли гарною оцінкою вважалася лише оцінка 5 (інтервал [5], [5]).

Оцінка вважалася гарною, якщо вона лежала в інтервалі [4], [5].

Хорошою оцінкою фільму для рекомендації вважалася будь-яка оцінка в інтервалі [3], [5].

Всі тести проводилися на комп'ютері під керуванням OS Windows із процесором Intel Core i7 2.3 ГГц та 8 ГБ оперативної пам'яті. Версія MATLAB – R2013a.

Середні значення результатів наведені у таблиці 3.2.

Таблиця 3.2 – Результати експериментів

	Середній час, сек	Середня точність, %	Середня повнота, %
Запропонований метод на основі занань [5], [5]	**3.62**	**19.42**	**50.52**
Slope One[5], [5]	18.37	1.57	23.41
Запропонований метод на основі знань [4], [5]	**18.23**	**55.61**	**63.33**
Slope One[4], [5]	18.90	53.99	30.39
Запропонований метод на основі знань[3], [5]	32.98	80.11	**83.65**
Slope One[3], [5]	**18.51**	**83.81**	81.88

У якості критеріїв порівняння методів використовувалися: середній час роботи алгоритму в секундах, середня точність та середня повнота.

З таблиці з результатами виразно видно, що алгоритм запропонований у даній кваліфікаційній роботі сильно випереджає Slope One за всіма критеріями оцінювання початкових параметрів [5], [5].

Для параметрів [4], [5] обидва підходи мають схожий середній час роботи та точність, але Slope One видає вдвічі менші рекомендації.

Для параметрів [3], [5] алгоритми мають схожі середні значення точності та повноти, але запропонований у середньому виконує роботу у півтора рази повільніше.

З цього можна зробити висновок, що розроблений метод рекомендацій має право на існування.

3.12 Висновки до розділу

У межах третього розділу даної кваліфікаційної роботи здійснено обґрунтування, проектування та розробку знання-орієнтованого методу побудови персоналізованого переліку товарів та послуг в рекомендаційних системах. Описано веб-додаток для реалізації рекомендаційної системи, проведено тестування.

ВИСНОВКИ

У ході виконання даної роботи проведено дослідження та розробка знання-орієнтованих методів побудови персоналізованого переліку товарів та послуг в рекомендаційних системах.

На основі вищевикладеного варто зробити наступний висновок:

Рекомендаційні системи становлять одну з основних частин екосистеми електронної комерції, що дозволяє користувачам відсіювати великі обсяги інформації та продуктів. Дослідження у цій галузі дозволяють зрозуміти, як технологія рекомендацій може застосовуватися уконкретних галузях.

Розроблено гібридний метод у ситуації холодного старту на основі проведеного порівняльного аналізу алгоритмів колаборативної фільтрації. Для цього використовувалися такі критерії: точність, повнота, швидкість навчання алгоритму, а також середньоквадратична та абсолютна помилки. Також було проаналізовано та частково вирішено проблеми та слабкі сторони кожного з підходів колаборативної фільтрації. Отримана система має помітне поліпшення точності характеристик порівняно з окремими алгоритмами колаборативної фільтрації, що використовуються, при відносно невеликому збільшенні часу навчання.

Теоретична значущість роботи полягає у розробці універсального підходу до формування рекомендацій на основі даних із різних предметних областей. Для підвищення якості рекомендацій запропоновано використовувати композицію методів аналізу даних, яка дозволяє ефективно вирішувати завдання кластеризації, пошуку дублікатів та асоціацій у рекомендаційних системах кількох спектрів областей. Цей підхід може використовуватися для розробки рекомендаційних систем для груп предметних областей зі специфічними вимогами, які складно врахувати під час розробки.

Виконаний аналіз методів побудови рекомендацій на основі знань показав, що при виборі товарів за їх властивостями користувач має враховувати множину технічних характеристик товарів, що суттєво ускладнює вибір. Тому важливою задачею є уточнення обмежень з використанням знань щодо взаємозв’язків між властивостями товарів, запропонованих в рекомендації.

Розроблено метод побудови рекомендацій на основі адаптації обмежень, який ітеративно уточнює задані користувачем обмеження з урахуванням знань щодо властивостей групи предметів, які вибирає користувач.

Виконано експериментальну перевірку розробленого методу .

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Abed, Lamees & Hamad, Murtadha & Aljaaf, Ahmed. (2023). A review of marketing recommendation systems. AIP Conference Proceedings. 2591. 030050. 10.1063/5.0119651.

2. Чалий С.Ф., Лещинський В.О., Лещинська І.О. Інтеграція локальних контекстів споживачів в рекомендаційних системах на основі відношень еквівалентності, схожості та сумісності. Process mining Materials of the VII International Scientific Conference «Information-Control System and Technologies» 17th-18th September, 2018, Odessa. С.142-144.

3. Noughabi, Havva & Behkamal, Behshid & Zarrinkalam, Fattane & Kahani, Mohsen. (2023). Post-hoc Explanation for Twitter ListRecommendation. 10.21203/rs.3.rs-2457669/v1.

4. Sasaki, So. (2020). The Tentative Definition of Environmental Service Trade: The Negative List Method Using Environmental Goods Trade.

5. Чалий С.Ф., Лещинський В.О., Лещинська І.О. Моделювання контексту в рекомендаційних системах. Науковий журнал «Проблеми інформаційних технологій», 2018, №. 1(023). С. 21-26..

6. Zhang, Yongfeng. (2016). Explainable Recommendation – Theory and Applications.

7. Kang, Guosheng & Liang, Bowen & Liu, Jianxun & Cao, Bu-Qing & Xu, Yu. (2022). QoS-Centric Web service Recommendation with Personalized Diversity. 10.21203/rs.3.rs-2230319/v1.

8. Pang, Nan. (2022). A Personalized Recommendation Algorithm for Semantic Classification of New Book Recommendation Services for University Libraries. Mathematical Problems in Engineering. 2022. 1-8. 10.1155/2022/8740207.

9. Han, Shutao & Liu, Qian & Xiao, Mingkun & You, Fadong & Li, Xuchu & Kim, Junmin. (2023). Personalized Internet Celebrity Check-In

Recommendation System Based on Hybrid Recommendation Algorithm. 10.1007/978-3-031-29097-8_41.

10. Han, Yufan. (2022). Personalized News Recommendation Algorithm for Event Network. Mathematical Problems in Engineering. 2022. 1- 10. 10.1155/2022/7813457.

11. Lei, Hao & Shan, Xinru & Jiang, Liwei. (2022). Personalized Item Recommendation Algorithm for Outdoor Sports. Computational Intelligence and Neuroscience. 2022. 1-10. 10.1155/2022/8282257.

12. Wu, Bing & Liu, Lixue. (2023). Personalized Hybrid Recommendation Algorithm for MOOCs Based on Learners' Dynamic Preferences and Multidimensional Capabilities. Applied Sciences. 13. 5548. 10.3390/app13095548.

13. Cao, Lijuan. (2022). Library Personalized Recommendation System Based on Collaborative Filtering Recommendation Algorithm. 10.1007/978-3-030-97874-7_61.

14. Li, Yiyang. (2023). Personalized Course Recommendation Model of University Environmental Design Based on Collaborative Filtering Algorithm. 10.1007/978-981-99-2287-1_49.

15. Wang, Yue & Qin, Zhaoxiang & Tang, Jun & Zhang, Wei. (2022). Optimization of Digital Recommendation Service System for Tourist Attractions Based on Personalized Recommendation Algorithm. Journal of Function Spaces. 2022. 1-9. 10.1155/2022/1191419.

16. Yun, Laiyan & Luo, Zhenrong. (2022). Multisource Information Fusion Algorithm for Personalized Tourism Destination Recommendation. Mathematical Problems in Engineering. 2022. 1-11. 10.1155/2022/3503548.

17. Geng, Baojun. (2023). Research on personalized recommendation algorithm for micromechanical sensors based on cloud model. Applied Mathematics and Nonlinear Sciences. 10.2478/amns.2023.1.00080.

18. Cheng, Bin & Chen, Ping & Zhang, Xin & Fang, Keyu & Qin, Xiaoli & Liu, Wei. (2023). Personalized Privacy Protection-Preserving

Collaborative Filtering Algorithm for Recommendation Systems. Applied Sciences. 13. 4600. 10.3390/app13074600.

19. Zhang, Liang & Zeng, Xiao & Lv, Ping. (2022). Higher Education-Oriented Recommendation Algorithm for Personalized Learning Resource. International Journal of Emerging Technologies in Learning (iJET). 17. 4-20. 10.3991/ijet.v17i16.33179.

20. Yang, Zhen. (2022). Research on Personalized Product Recommendation Algorithm for User Implicit Behavior Feedback. 10.1007/978-981-19-6901-0_149.

21. Lv, Guangping. (2022). Personalized Recommendation Algorithm of Literary Works Based on Annotated Corpus. Mobile Information Systems. 2022. 10.1155/2022/5198612.

22. Xueting, Liu. (2022). Personalized Recommendation Algorithm of Tourist Attractions Based on Transfer Learning. Mathematical Problems in Engineering. 2022. 1-7. 10.1155/2022/2520140.

23. Zeng, Fangqin & Tang, Rong & Wang, Yibai. (2022). User Personalized Recommendation Algorithm Based on GRU Network Model in Social Networks. Mobile Information Systems. 2022. 1-8.10.1155/2022/1487586.

24. Su, Xueping & He, Jiao & Ren, Jie & Peng, Jinye. (2022). Personalized Chinese Tourism Recommendation Algorithm Based on Knowledge Graph. Applied Sciences. 12. 10226. 10.3390/app122010226.

25. Liu, Feng & Guo, Weiwei. (2022). Personalized Recommendation Algorithm for Interactive Medical Image Using Deep Learning. Mathematical Problems in Engineering. 2022. 10.1155/2022/2876481.

26. Pang, Zhaojun & Wu, Wenbin & Fan, Xinxin & Liu, Zhixin. (2022). A New Online Education Personalized Recommendation Algorithm. Journal of Engineering Research. 10.36909/jer.ICCSCT.19485.

27. Qiu, Gang & Cheng, Jie. (2022). Network Resource Personalized Recommendation System Based on Collaborative Filtering Algorithm. 10.1007/978-3-030-94551-0_50.

28. Jawaheer, G. Modeling user preferences in recommender systems: a classification framework for explicit and implicit user feedback / G. Jawaheer, P. Weller, P. Kostkova // ACM Transactions on Interactive Intelligent Systems. 2014. Vol. 4. №2. P. 1-26.

29. Jones, K.S. A statistical interpretation of term specificity and its application in retrieval // Journal of Documentation. 2004. Vol. 60. №5. P. 493-502.

30. Recommender Systems Handbook / F. Ricci, L. Rokach, B. Shapira [et al.]. – New-York : Springer Science+Bussiness Media, 2011. 842 p.

31. Kaminskas, M. Product recommendation for small-scale retailers / M. Kaminskas, D. Bridge, F. Foping, D. Roche // In International Conference on Electronic Commerce and Web Technologies. 2015. P. 17- 29.

32. Recommender systems: an introduction / D. Jannach, M. Zanker, A. Felferning [et al.]. – New-York : Cambridge University Press, 2011. 352 p.

33. Kaminskas, M. Product-Seeded and Basket-Seeded Recommendations for Small-Scale Retailers / M. Kaminskas, D. Bridge, F. Foping, D. Roche // Journal on Data Semantics. 2016. Vol. 6. №1. P. 3-14.

34. Manning, C. D. An Introduction to Information Retrieval Draft / C. D. Manning, P. Raghavan, H. Schütze // Information Retrieval Journal. 2010. Vol. 13. № 2. P. 192-195.

35. Ming, L. Grocery shopping recommendations based on basket-sensitive random walk / L. Ming, B. M. Dias // In Proceedings of the 15th International Conference on Knowledge Discovery and Data mining. 2009. P. 1215–1224.

ДОДАТОК А

Відомість кваліфікаційної роботи

Позначення	Найменування	Дод. відомості
	Текстові документи	
1.	Пояснювальна записка	77 с.
	Інші документи	
2.	Презентаційні матеріали	___плакатів

.	.	Прізвище та ініціали.	Підп.	Дата			
Розробив		Чмутов Д.С.			Дослідження та розробка знання-орієнтованих методів побудови персоналізованого переліку товарів та послуг в рекомендаційних системах	Шифр групи	Код напр./спец.
Перевірив		Чалий С.Ф.				СШІм-21-2	122
Н.контр.		Малєєва І.А.				ХНУРЕ кафедра ШІ	
Затв.		Філатов В.О.					