

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
Кафедра Медіасистеми та технології
Рівень вищої освіти другий (магістерський)
Спеціальність 186 Видавництво та поліграфія
Тип програми Освітньо-професійна
Освітня програма Комп'ютерні технології
та системи видавничо-поліграфічних виробництв
(шифр і назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри МСТ _____
(підпис)
« 30 » жовтня 2023 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

студентові Короля Андрія Леонідовича
(прізвище, ім'я, по батькові)

1. Тема роботи Підтримка прийняття рішень для агрегації новин про друковані видання
затверджена наказом по університету від 27 жовтня 2023р. № 1249 Ст

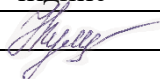
2. Термін подання студентом роботи до екзаменаційної комісії 21 січня 2024 р.

3. Вихідні дані до роботи
Додатки, що призначені для вибору нових друкованих видань; електронна документація програмних засобів, тестові набори даних

4. Перелік питань, що потрібно опрацювати в роботі
Аналіз процесу прийняття рішень про покупки та використання рекомендаційних систем у ньому; Аналіз процесу прийняття рішення про покупку книжкової новинки; Основні підходи до реалізації рекомендаційних систем; Експериментальна частина; Економічна частина; Висновки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій
Мета, об'єкт, предмет та часткові завдання дослідження; Процес прийняття рішень покупцем; Інформаційний пошук під час прийняття рішення покупцем; Процес оцінки альтернатив перед покупкою; Модель процесу прийняття рішення покупцем; Вихідні дані для рекомендаційної системи; Принцип дії алгоритму к найближчих сусідів; Асинхронний режим дії системи агрегації новин про книжкові видання; Схема процесу агрегації новин про книжкові видання; Економічна частина; Висновки.

6. Консультанти розділів роботи

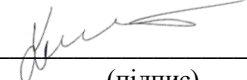
Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Основна частина	проф. Кулішова Н.Є.		19.01.24
Економічна частина	ас. Помогалова Н.В.		20.01.24

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз процесу прийняття рішень про покупки та використання рекомендаційних систем у ньому	10.10.2023	виконано
2	Аналіз процесу прийняття рішення про покупку книжкової новинки	25.10.2023	виконано
3	Основні підходи до реалізації рекомендаційних систем	15.11.2023	
4	Експериментальна частина	25.11.2023	виконано
5	Економічна частина	29.12.2023	виконано
6	Оформлення пояснювальної записки	10.01.2024	виконано
7	Оформлення графічної частини	15.01.2024	виконано
8	Попередній захист кваліфікаційної роботи	17.01.2024	виконано

Дата видачі завдання 30 жовтня 2023 р.

Студент


(підпис)

Король А.Л.

Керівник роботи


(підпис)

проф. Кулішова Н.Є.
(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 82 с., 7 табл., 22 рис., 2 дод., 17 джерел.

РІШЕННЯ ПРО ПОКУПКУ, КУПІВЛЯ КНИЖОК, РЕКОМЕНДАЦІЙНА СИСТЕМА, ПРИЙНЯТТЯ РІШЕННЯ.

Метою роботи є підвищення якості програмних застосунків для вибору або купівлі книжок.

Об'єктом дослідження є процес прийняття рішення споживача про придбання нових книжок в інтернет-магазині.

У ході виконання кваліфікаційної роботи було розглянуто процес прийняття рішення про покупку книжкових новинок, яке приймає споживач, розглянути місце рекомендаційних систем в цьому процесі; проаналізовано основні принципи реалізації рекомендаційних систем, зокрема, в умовах відсутності початкових даних про книгу або користувача. Було розроблено експериментальний зразок автоматичної системи вибору книжкової новинки для нового користувача та перевірено його ефективність. Проведено економічне обґрунтування дослідження.

ABSTRACT

Explanatory note of qualification work: 82 p., 7 tab., 22 pic., 2 app., 17 sources.

PURCHASE DECISION, PURCHASE OF BOOKS, RECOMMENDATION SYSTEM, DECISION MAKING.

The purpose of the work is to improve the quality of software applications for choosing or buying books.

The object of the study is the process of making a consumer's decision to purchase new books in an online store.

In the course of the qualification work, the process of making a decision on the purchase of book novelties, which is taken by the consumer, was considered, the place of recommendation systems in this process was considered; the basic principles of implementation of recommender systems are analyzed, in particular, in the absence of initial data about the book or the user. An experimental sample of an automatic book selection system for a new user was developed and its effectiveness was tested. The economic justification of the research was carried out.

ЗМІСТ

	С.
ВСТУП.....	7
1 АНАЛІЗ ПРОЦЕСУ ПРИЙНЯТТЯ РІШЕНЬ ПРО ПОКУПКИ ТА ВИКОРИСТАННЯ РЕКОМЕНДАЦІЙНИХ СИСТЕМ У НЬОМУ	9
1.1 Поведінка споживача та прийняття рішень про покупки	9
1.2 Системи рекомендацій для вибору книжок	13
2 АНАЛІЗ ПРОЦЕСУ ПРИЙНЯТТЯ РІШЕННЯ ПРО ПОКУПКУ КНИЖКОВОЇ НОВИНКИ.....	17
3 ОСНОВНІ ПІДХОДИ ДО РЕАЛІЗАЦІЇ РЕКОМЕНДАЦІЙНИХ СИСТЕМ	30
3.1 Проблема «холодного старту».....	34
3.2 Алгоритми факторизації	43
3.3 Оцінка якості системи	51
3.4 Складнощі моделей	55
4 ЕКСПЕРИМЕНТАЛЬНА ЧАСТИНА	57
4.1 Вибір програмного середовища.....	57
4.2 Реалізація системи підбору книжкових новинок.....	59
4.3 Оцінювання якості запропонованої системи агрегації новин про книжкові видання	68
5 ЕКОНОМІЧНА ЧАСТИНА	72
5.1 Характеристика науково-дослідного рішення	72
5.2 Етапи виконання НДР, їх трудомісткість та заробітна плата	72
5.3 Розрахунок одноразових витрат на розробку НДР.....	75
5.4 Оцінка результатів науково-дослідної роботи.....	77
5.5 Визначення економічної ефективності результатів НДР	79
ВИСНОВКИ	80
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	81
ДОДАТОК А Програмний код для визначення групи користувача	83
ДОДАТОК Б Програмний код для визначення рейтингу нової книжки	85

ВСТУП

Система рекомендацій є одним із найпопулярніших застосувань науки про дані. Кожній інтернет-компанії, що займається споживачами, потрібна система рекомендацій, як у Netflix, Youtube, стрічка новин тощо. Те, що компанія хотіла б показати користувачеві з величезного асортименту товарів, – це знаходить система рекомендацій. Системи рекомендацій мають на меті передбачити інтереси користувачів і показати продукти, які цілком імовірно їх цікавлять. Вони є одними з найпотужніших систем машинного навчання, які онлайн-продавці впроваджують для стимулювання продажів. За допомогою зібраної інформації про вподобання користувачів, алгоритмів машинного навчання та штучного інтелекту рекомендаційні системи дозволяють пропонувати клієнтам товари та послуги згідно персональним запитам, досвіду споживання. У найзагальнішому сенсі рекомендації – це пропозиція користувачеві релевантних об'єктів, персоналізація на противагу уніфікації, яка не враховує інтересів конкретного користувача. Історично рекомендаційні системи народилися як відгалуження завдань пошуку. У пошуку є запит, є контекст (вподобання користувача та його характеристики) і є елементи, які потрібно ранжувати. У рекомендаціях все аналогічно, але запитом виступає сам користувач, його переваги.

Продуктом, споживання якого суттєвим чином залежить від особистих вподобань, життєвого досвіду, культурного рівня та планів на майбутній розвиток, для людей є книга. Купуючи книжки, люди найчастіше спираються на те, що вони знають про автора, про видавця, про історію створення книжки, а також на відгуки знайомих або літературних критиків, на рецензії в інших виданнях. І все одно, іноді доводиться додатково ознайомитися з фрагментами змісту, перегортаючи сторінки у книгарні, або користуючись попереднім переглядом в інтернет-магазині. Так чи інакше, очевидно, що вибір книжки – справа важлива і непростя.

За останні кілька років українським видавцям довелося пройти найскладніші випробування: спочатку пандемія COVID-19, а потім – нова фаза російсько-української війни. В таких умовах розвиток окремих видавництв, усієї книжкової галузі України залежить від того, як складається попит читачів, наскільки він змінюється, які теми більше хвилюють людей, що купують книжки.

Метою роботи є підвищення якості програмних застосунків для вибору або купівлі книжок.

1 АНАЛІЗ ПРОЦЕСУ ПРИЙНЯТТЯ РІШЕНЬ ПРО ПОКУПКИ ТА ВИКОРИСТАННЯ РЕКОМЕНДАЦІЙНИХ СИСТЕМ У НЬОМУ

1.1 Поведінка споживача та прийняття рішень про покупки

У роки війни український книжковий ринок не лише існував, а й знаходив можливості для розвитку. Десятки видавців передрукували заплановані випуски, які були відкладені з кінця лютого, а також підготували та випустили нові випуски. За роки війни було випущено 290 нових назв. З лютого 2022 року кількість виданих книг в Україні скоротилася майже вдвічі. На кінець вересня вийшло 5,5 тис. книжок [1]. Наклади зменшилися на 35 відсотків. Водночас продажі аудіокниг та електронних книг стрімко зростають. Наприклад, з 2022 р. у період з лютого по червень в книжковому онлайн-магазині Yaкабоо було продано 52 000 електронних копій.

Ці дані підтверджують основні тенденції і в світовій торгівлі книжками. У 2022 році обсяг продажів перевищив рівень 2021 року і збільшився більш ніж на 10% у деяких групах. Індекс світового книговидавничого ринку зріс на 5,4%. За оцінками експертів, обсяг світового ринку поліграфії зросте в середньому на 1%, тобто до 2020-2025 років досягне 18,69 млрд доларів. За даними Publishers Weekly, продажі книг, виданих у 2022 році, зросли на 7,9%. Позитивна динаміка відмічена в категорії художня література та нон-фікшн, серед книжок для дітей та юнацтва; також на 5-6% подорожчали книги для дорослих [1].

У 2022 році продажі електронних книг на світовому ринку зростали, але меншими темпами, ніж торік. На розвинених ринках (США та Великобританія) електронні книги приносили більшу частку доходу, ніж друковані книги. Зростання продажів склало 10,7% (11,88% у 2021 році). Світовий ринок електронних книг у 2022 році оцінювався приблизно в 20 млрд. доларів. Світовий попит на аудіокниги зростає, але меншими темпами:

34,7% у 2018 році порівняно з 16,4% у 2019 році; у 2020 році - понад 15%; до кінця 2021 року - 17%; у 2022 році попит на аудіокниги встановлено на рівні 18% [1].

Найбільші книжкові ринки 2022 року були у Великобританії, Німеччині, Франції, Іспанії та Італії. Загальна ринкова вартість оцінюється приблизно в 40 мільярдів євро. За даними Statista, половина всіх книжок, проданих у світі минулого року, прийшла з Китаю та США. У 2022 році бібліотеки збільшують запас електронних книг приблизно на 20% порівняно з попередніми роками. Згідно з дослідженням Ithaca S+R, опублікованим у грудні 2022 року, тенденція бібліотечних інвестицій у цифрові ресурси та послуги посилюється та прискорюється. Продажі книг зросли на 8% порівняно з 2021 роком. Більшість книжкового бізнесу належить Amazon, продажі якої зросли майже на 40% [1].

Всі ці дані свідчать про зростання цікавості до книжок, у друкованому чи електронному вигляді. Кількість найменувань та авторів збільшується також через те, що в інтернет-магазині немає складнощів із зберіганням книжок, виданих кілька років тому, навіть якщо з тиражу непроданими залишилися кілька примірників. Це явище навіть отримало власну назву «довгих хвостів». В таких умовах актуальним є завдання прийняття рішення – що придбати, щоб прочитати.

Розглянемо більш детально це завдання. По-перше, необхідно відзначити високий інтерес споживача до персоналізації рекомендації, по-друге, наявна зацікавленість споживача в активній участі у віртуальних спільнотах транзакцій, розташованих у середовищі онлайн-покупок, яку він/вона реалізує, надсилаючи оцінки та коментарі щодо продукту. По-третє, саме під час вибору та покупки книжки проявляється інтерес споживача до думок інших споживачів у віртуальних спільнотах щодо продукту. Також, важливими є такі чинники:

– книги є однією з найпоширеніших категорій продуктів, що продаються через Інтернет у всьому світі та в Україні. Отже, імовірніше, що

учасники опитування щодо рекомендацій книг мають досвід у цьому контексті, що робить результати опитування більшими надійний;

– книги – це клас продуктів, де суб'єктивні смаки та вподобання мають велике значення для прийняття рішення про покупку. Як наслідок, поради спільноти є важливими аспектами у виборі цієї категорії продуктів;

– продаж книг через Інтернет дозволяє впровадити автоматизовані системи прийняття рішення, засновані на використанні методів машинного навчання.

Під час вибору книжок користувач стикається з проблемою, яка відноситься до класу так званих «проблем преференційного вибору» [2], тобто йому потрібно прийняти рішення з початкового великого набору потенційних альтернатив. У цьому контексті процеси прийняття рішень зазвичай моделюють як феномен «обмеженої раціональності» [2]. Обмежена раціональність полягає в тому, що в складних середовищах прийняття рішень люди часто не можуть глибоко оцінити всі доступні альтернативи, перш ніж зробити свій вибір, через когнітивні обмеження свого розуму і кінцевий проміжок часу, який вони мають для прийняття рішення; отже, вони прагнуть досягти задовільного, хоча й не обов'язково оптимального рівня досягнення, застосовуючи свою раціональність лише після значного спрощення набору доступних варіантів.

Можливо, когнітивні зусилля можна зменшити за допомогою багатоетапного процесу прийняття рішень, у якому глибина обробки інформації змінюється залежно від етапу [2]. Спочатку люди переглядають повний простір рішення (наприклад, набір усіх книг, доступних в інтернет-магазині), щоб ідентифікувати (великий) набір потенційних альтернатив або набір пошуку (наприклад, перелік книжок певного жанру або певного автора); потім вони шукають цей набір і визначають підмножину перспективних кандидатів (набір для розгляду). Згодом вони отримують детальну інформацію про вибрані альтернативи, які необхідно серйозно розглянути (набір поглибленого порівняння), оцінюють і порівнюють їх більш детально, і,

нарешті, роблять конкретний вибір. Хоча деякі з наведених вище дій можна повторювати, цей процес за своєю суттю є лінійним і, ймовірно, закінчиться тим, що користувач зробить конкретний вибір і придбає послугу.

Добре відомо, що в будь-якому контексті спроба RS зменшити зусилля користувача щодо прийняття рішень ризикує створити ефект, коли RS мають тенденцію показувати лише інформацію, яка узгоджується з попередніми точками зору користувачів, фактично ізолюючи користувача в бульбашку, яка має тенденцію виключати нові та випадкові елементи [2]. В такому випадку процес прийняття рішення має бути повторений, можливо, кілька разів. Така ситуація може призвести до збільшення зусиль користувачів щодо прийняття рішень і часу для прийняття рішень, а отже, до зниження якості процесу прийняття рішень. Це, у свою чергу, потенційно негативно вплине на сприйняття користувачами щодо довіри, корисності та задоволеності RS. Що ще гірше, користувач може здатися, не завершивши процес прийняття рішення, залишити поточну службу онлайн-бронювання та спробувати іншу, що матиме очевидні наслідки для постачальника послуг з точки зору довіри клієнтів і фактичних бізнес-результатів.

Щоб подолати ці проблеми, користувачі повинні отримувати нові та випадкові рекомендації [2]. Це парадигматична зміна ролі RS у процесі прийняття рішень: від інструменту, який допомагає користувачам звужити пошуковий набір і набір розглядів у випадку необмежених ресурсів, до інструменту, який розширює глибинний набір у випадку обмежених ресурсів.

Визначення стратегій проектування РС, які враховують можливість обмежених ресурсів, є складним питанням. Наведемо деякі вимоги, які необхідно взяти до уваги.

1. Підтримка процесів прийняття рішень, які є сильно повторюваними, максимізуючи зручність виконання та повторного виконання попередніх кроків, зокрема у повторному визначенні уподобань, коли користувач усвідомлює відсутність доступних елементів, які відповідають його вимогам.

2. Потрібно підтримувати довіру користувачів і підтримувати участь користувача в процесі прийняття рішень, незважаючи на початкові невдачі, які потенційно можуть виникнути через брак ресурсів. У цьому відношенні слід визначити конкретні стратегії пояснення, які не тільки покращують прозорість і пояснюють, як генеруються рекомендації, але також повідомляють користувачеві про брак ресурсів.

3. Здатність діяти одночасно як фільтр, який обмежує набір цінних альтернатив, і як мультиплікатор, який допомагає користувачеві розширити свій кругозір, рекомендуючи випадкові альтернативи.

Нарешті, концепції цілей користувача (різких або м'яких) і обмежених ресурсів мають вплив на моделі оцінювання, методології та емпіричні дослідження щодо RS як інструментів підтримки прийняття рішень.

1.2 Системи рекомендацій для вибору книжок

Механізм рекомендацій – це клас машинного навчання, який пропонує відповідні пропозиції клієнту. До появи системи рекомендацій основною тенденцією купувати було прийняття пропозицій друзів. Але тепер Google знає, які новини буде читати споживач, Youtube знає, які типи відео він буде дивитися на основі історії пошуку, історії переглядів або історії покупок.

Система рекомендацій допомагає організації створювати лояльних клієнтів і зміцнювати довіру до бажаних продуктів і послуг. Сучасна система рекомендацій має бути настільки потужною, щоб впоратися з новими клієнтами, які відвідали сайт уперше. Вони рекомендують продукти, які на даний момент є трендовими або високо оціненими, а також можуть рекомендувати продукти, які приносять максимальний прибуток компанії.

Система рекомендацій книг – це тип системи рекомендацій, коли ми маємо рекомендувати схожі книги читачеві на основі його інтересу. Система рекомендацій книг використовується онлайн-сайтами, які надають електронні книги.

В прикладі [3] пропонується РС для вибору книжок, яка вживає матричну факторизацію для побудови поради. Але цей підхід буде корисним лише для постійних користувачів. Система обмежує кількість постійних користувачів, що оцінили щонайменше 200 книжок, або розглядає книжки, що отримали щонайменше 50 оцінок. Для оновлення рекомендацій згідно оновленню даних про книжки в системі вживається метод к-найближчих сусідів, який розв’язує завдання кластеризації на основі евклідової відстані. На жаль, цей метод чутливий до великої розмірності масиву даних, а також система в цілому неспроможна надати корисні поради користувачам, які зайшли вперше, або рекомендувати нові книжки.

Інша система [4] навчена на основі набору даних про вибір книжок, кожна з яких має 100 відгуків за 5-бальною шкалою. В цій системі (рис. 1.1) кластеризація даних реалізована за допомогою штучної нейронної мережі і в якості найбільш релевантних стовпців, які потрібні, щоб рекомендувати книгу користувачеві використовує такі: ‘id’, ‘book_id’, ‘isbn’, ‘authors’, ‘original_publication_year’, ‘title’, ‘average_rating’, ‘ratings_count’, ‘small_image_url’.

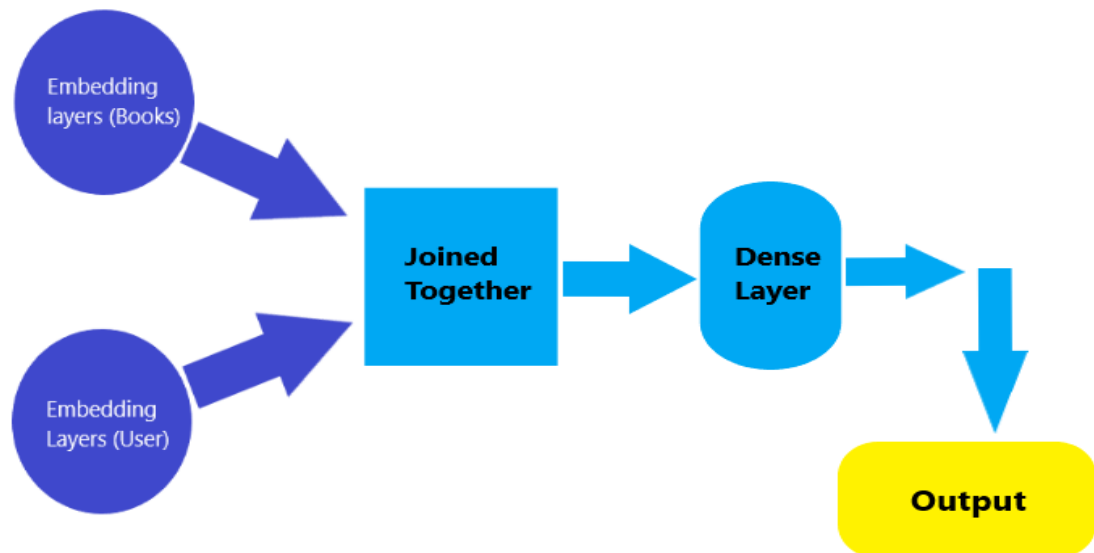


Рисунок 1.1 – Архітектура автоматичної рекомендаційної системи для вибору книжок, заснованої на використанні глибокої нейронної мережі

Система повертає 5 найпопулярніших книг для конкретного користувача. Але навчання штучних нейронних мереж вимагає великих наборів даних (обсягом щонайменше 100 000 записів), які не завжди є доступними, а також розглянута система не долає проблему рекомендацій для нових клієнтів або для нових книжок.

Ще одна система [5] рекомендує в режимі реального часу книжки на основі оцінок з набору даних Amazon Books. Дані, що надходять із цього потоку, можна запускати через Memgraph, який можна налаштувати за допомогою нової Memgraph Cloud або запустивши локальний екземпляр Memgraph DB. Остання версія Memgraph Lab, програми, яку можна використовувати для запитів і візуалізації даних, має нову функцію, яка допоможе користувачеві підключитися до потоку. Уся інформація, необхідна для підключення до потоку, доступна на веб-сайті Awesome Data Stream [6]. На жаль, користувачеві для цього доведеться переглядати документацію Cloud або посібник про підключення до потоків за допомогою Memgraph Lab. Система побудована на потоковому алгоритмі Online PageRank [7], який розраховує ймовірність випадкового потрапляння у певну вершину дерева пошуку. На жаль, для українських книжок ця система неприйнятна, як і інші розглянуті аналоги, оскільки не були навчені на наборах даних про видання, випущені в Україні. Крім того, в розглянутих рекомендаційних системах не вживаються підходи до створення поради про нові книжки, для яких відгуків спільноти ще недостатньо, або для нових користувачів сайту.

Тож, основною метою магістерської кваліфікаційної роботи є підвищення якості програмних застосунків для вибору або купівлі книжок.

Гіпотеза магістерської кваліфікаційної роботи: поєднання товарної колаборативної фільтрації та колаборативної фільтрації користувача дозволить знизити негативний вплив проблеми «холодного старту» на дію автоматичної системи підтримки прийняття рішень для агрегації новин про книжні видання.

Об'єктом дослідження є процес прийняття рішення споживача про придбання нових книжок в інтернет-магазині.

Предмет дослідження – методи машинного навчання, на основі яких розробляють рекомендаційні системи.

Для досягнення основної мети роботи потрібно розв'язати кілька часткових завдань:

- дослідити процес прийняття рішення про покупку книжкових новинок, яке приймає споживач, розглянути місце рекомендаційних систем в цьому процесі;

- вивчити основні принципи реалізації рекомендаційних систем, зокрема, в умовах відсутності початкових даних про книгу або користувача;

- розробити експериментальний зразок автоматичної системи вибору книжкової новинки для нового користувача;

- перевірити ефективність зразку системи під час експериментального дослідження.

2 АНАЛІЗ ПРОЦЕСУ ПРИЙНЯТТЯ РІШЕННЯ ПРО ПОКУПКУ КНИЖКОВОЇ НОВИНКИ

Гарне розуміння процесу прийняття рішення покупцем під час вибору книжок безумовно сприятиме задоволенню користувачів, що вдаються до Інтернет-покупки, та зростанню продажів і розвитку галузі в цілому.

Цілісні моделі намагаються об'єднати та взаємопов'язати всі фундаментальні конструкції поведінки споживача щодо процесу прийняття рішення [8]. На рис. 2.1 показано фазову модель цього процесу [8], яка включає фази: (1) визнання потреби, (2) пошук інформації, (3) оцінка альтернатив перед покупкою, (4) покупка, (5) споживання, (6) оцінка після споживання. Ця модель представляє дорожню карту міркувань споживачів.

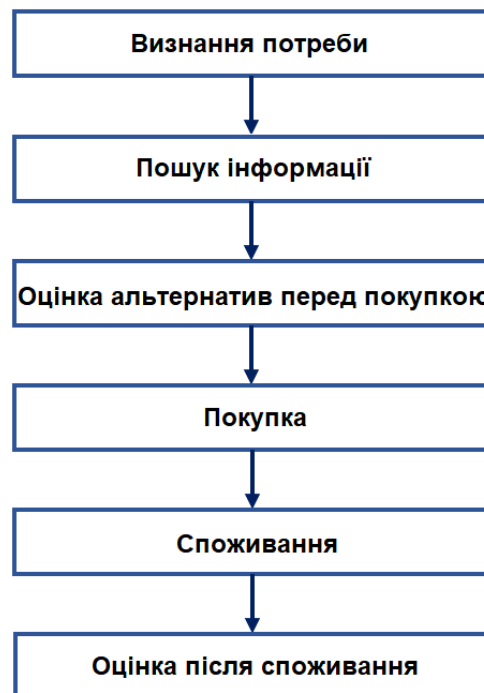


Рисунок 2.1 – Процес прийняття рішень покупцем

Визнання потреби відбувається, коли індивід відчуває різницю між тим, що він або вона сприймає як ідеальне, проти фактичного стану [8]. Розпізнавання потреби з'являється, коли споживач починає усвідомлювати,

що він або вона має потребу, яку, ймовірно, можна задовольнити через продукт або послугу. Визнання потреби може статися з причин, що не залежать від компанії, або може бути під зовнішнім впливом. Персоналізовані рекомендації, які надаються системами рекомендацій, можна розуміти як форму «реклами, налаштованої на особистість». Отже, системи рекомендацій можуть бути використані для більш ефективного створення або стимулювання цих потреб. На потреби клієнтів в основному впливають два фактори: (1) вплив навколишнього середовища (наприклад, культура, соціальний клас, особисті впливи) і (2) індивідуальні відмінності (наприклад, споживчі ресурси, мотивація, ставлення, знання) [8]. Якщо рекомендації враховують ці два фактори, вони можуть ефективно допомогти споживачеві визначити його потреби.

Пошук інформації є наступним кроком у моделі процесу прийняття рішень споживачем. Як тільки потреба визнана, споживачі починають шукати інформацію, щоб задовольнити незадоволені потреби. Цей пошук може бути внутрішнім або зовнішнім. Внутрішній пошук відноситься до отримання з пам'яті знань, важливих для прийняття рішень. Навпаки, зовнішній пошук відбувається, коли споживач збирає інформацію з ринку, аналогів або інших відповідних джерел інформації. Зовнішній пошук зазвичай відбувається після процесу внутрішнього пошуку: споживач вважає, що його знання недостатні для прийняття рішення про покупку, він чи вона, ймовірно, здійснить зовнішній пошук (рис. 2.2) [9].

Це може відбуватися пасивно (тобто споживач стає більш сприйнятливим до джерел інформації) або активно, коли споживач демонструє пошукову поведінку, наприклад перевірку споживчих публікацій, рекламних матеріалів, веб-сайтів або роздрібних точок продажу. Зовнішній пошук можна розділити на пошук перед покупкою та поточний пошук. Пошук перед покупкою мотивується майбутнім рішенням про покупку, тоді як поточний пошук відбувається регулярно, незалежно від спорадичних потреб у покупці [8, 10].



Рисунок 2.2 – Інформаційний пошук під час прийняття рішення покупцем

Системи рекомендацій можуть бути використані для допомоги споживачеві в обох категоріях зовнішнього пошуку. Наприклад, якщо книжковий ентузіаст регулярно отримує електронною поштою рекомендації нових видань у сфері його чи її інтересів, він чи вона отримує підтримку в процесі постійного пошуку.

Коли споживач застосовує зовнішній пошук, для обробки інформації [8, 10] використовуються наступні кроки:

- оприлюднення: на цій фазі інформація досягає споживача, за допомогою чого органи чуття споживачів активуються та починається попередня обробка;

- розуміння: споживачі аналізують повідомлення за категоріями значення, які вже зберігаються в пам'яті;

- прийняття: після досягнення розуміння повідомлення можна прийняти або відхилити як неприйнятне. Прийняття повідомлення є необхідною передумовою для модифікації або зміни існуючих поглядів або переконань споживачів;

- зберігання: збереження означає зберігання та прийняття повідомлення в пам'яті таким чином, щоб воно було доступним для подальшого використання.

Створюючи віртуальні спільноти та використовуючи системи рекомендацій, постачальники можуть використовувати цю інформацію для підвищення довіри. Наприклад, системи рекомендацій можуть узагальнювати критику спільноти та рекомендувати продукти з високими оцінками від членів віртуальної спільноти або експертів. Роблячи це, постачальники можуть допомогти споживачеві в процесі прийняття рішення, надаючи інформацію, яка не стосується маркетологів. Однак, щоб створити або зберегти довіру, дуже важливо чесно використовувати ці джерела інформації.

У цьому контексті виникає питання, наскільки широко споживачі здійснюють зовнішній пошук. Концепція «економіки інформації» [10], дає розуміння цієї проблеми з точки зору витрат і прибутку. Згідно з цією системою споживачі інформують себе про продукти та послуги на ринку до точки, коли граничні витрати на збір додаткової інформації дорівнюють або перевищують граничну віддачу (тобто вигоди від збору нової інформації). Серед чинників обсягу пошукових зусиль визначають передбачуваний ризик (тобто невпевненість споживачів щодо потенційних позитивних і негативних наслідків рішення про покупку), кількість досвіду роботи з класом продукту, зміст досвіду (тобто позитивний чи негативний) і вартість. Оскільки онлайн-системи рекомендацій можуть значно скоротити витрати на пошук, вони є цінним інструментом для споживачів щодо зовнішнього пошуку.

Оцінка альтернатив перед покупкою. На цій стадії процесу прийняття споживачем рішення основна увага приділяється способу, яким споживач оцінює альтернативи для покупки [8, 10]. Перш ніж прийняти рішення про покупку, споживачі зазвичай порівнюють різні продукти та послуги. Споживачі можуть використовувати вже існуючі або нові оцінки, збережені в пам'яті, щоб вибрати продукти та послуги, які, швидше за все, задовольнять їхні потреби. На те, як відбувається цей процес, впливають індивідуальні відмінності та вплив навколишнього середовища. У цьому процесі розрізняються основні та визначальні атрибути [8, 10]. Споживачі оцінюють основні параметри як найважливіші характеристики продукту чи послуги

(наприклад, ціна, швидкість процесора та розмір жорсткого диска персонального комп'ютера). Однак споживач застосовує визначальні атрибути для фактичного вибору певного продукту та послуги, особливо коли основні атрибути вважаються рівними між альтернативами. Вирішальні атрибути часто виявляються дуже суб'єктивними для особистого смаку споживача. На рис. 2.3 показано процес оцінки перед покупкою. Коли потрібно прийняти рішення, споживачі зазвичай не розглядають усі доступні варіанти. Насправді вони обмежують альтернативи підмножиною, яка називається «набір для розгляду».

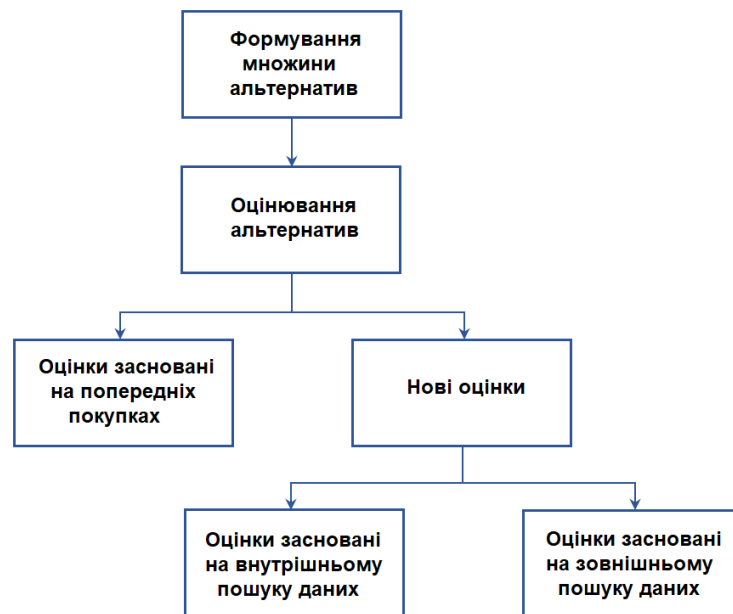


Рисунок 2.3 – Процес оцінки альтернатив перед покупкою

Коли споживачі оцінюють альтернативи, вони можуть:

- покладатися на вже існуючі оцінки, що зберігаються в пам'яті (у цьому випадку набір для розгляду називається «набір для пошуку»);
- вирішити побудувати нові оцінки на основі інформації, отриманої через внутрішній або зовнішній пошук [8, 10].

Попередні оцінки можуть базуватися на власному досвіді споживачів у минулих покупках і споживанні продукту чи послуги. В інших випадках - особливо коли у споживача бракує власного досвіду - непрямий або вторинний

досвід (наприклад, враження від друзів) може стати домінуючим для оцінки. Це ілюструє важливість усного спілкування в процесі прийняття рішень. Якщо споживачі не можуть (наприклад, відсутність попереднього досвіду) або не бажають (наприклад, зміни факторів навколишнього середовища) покладатися на вже існуючі оцінки, вони можуть вирішити створити нові оцінки. При цьому споживачі можуть застосовувати два основні процеси: процес категоризації або поетапний процес. Процес категоризації відноситься до оцінки альтернатив щодо певних ментальних категорій, до яких вони віднесені. Основне припущення полягає в тому, що люди природним чином поділяють світ об'єктів навколо них на категорії, що дозволяє ефективний спосіб обробки та розуміння середовища. Споживачі зазвичай приписують своїм розумовим категоріям певний ступінь симпатії чи антипатії. Крім того, оцінка, віднесена до певної категорії, може бути перенесена на будь-який новий об'єкт цієї конкретної категорії [8, 10]. Регулярно саме так споживачі формують оцінки нових продуктів і послуг. Якщо новим продуктам або послугам присвоєно певну категорію, вони будуть оцінюватися відповідно до ступеня вподобання цієї категорії. Цей процес отримання оцінок також можна назвати «впливом, керованим схемою», оскільки типова категорія «зразків» або «прототипів» функціонує як а. схема процесу оцінювання. «Зразки» – це добре відпрацьовані реальні приклади категорії, тоді як «прототипи» – це абстрактні вигадані образи категорії, які втілюють типові атрибути та характеристики, пов'язані з категорією.

Більш складний метод оцінки продуктів і послуг називається поетапним процесом [8, 10]. У цьому випадку продукти оцінюються за атрибутом--поатрибутна основа. Продукти сприймаються як набір окремих атрибутів, причому кожен атрибут має окрему суб'єктивну цінність або вагу. Поетапний процес можна розділити на три фази:

- визначення важливих критеріїв або розмірів продукту;
- судження про альтернативні рішення з огляду на кожен окремий атрибут;
- судження про загальну продуктивність альтернатив.

Коли рішення включають «непорівнянні» альтернативи (наприклад, споживач повинен вибирати між різними категоріями продукту), потрібно використовувати більш абстрактні критерії, оскільки альтернативи мають лише кілька спільних критеріїв, за якими можна проводити порівняння. Наприклад, якщо споживачеві доводиться вибирати між різними формами розваг (наприклад, покупка домашньої стереосистеми чи купівля ігрової консолі), для порівняння потрібно використовувати більш абстрактні критерії, такі як статус чи необхідність.

Наступний крок вимагає від споживача оцінити кожен продукт і послугу за критерієм, який раніше вважався важливим. Споживачі часто використовують так звані «відсічення» для спрощення прийняття рішень. Відсічення представляє заздалегідь визначений прийнятний рівень для атрибута.

Третім і останнім кроком у поетапному процесі є судження про загальну ефективність альтернатив у наборі для розгляду. Отже, це виходить з оцінки ефективності альтернатив щодо кожного атрибута. У принципі можна виділити компенсаційні та некомпенсаційні стратегії оцінювання.

Стратегії некомпенсаційного оцінювання стосуються процесу оцінювання, де слабкість продукту за одним атрибутом не може бути компенсована його високою продуктивністю за іншими атрибутами [8, 10]. Некомпенсаційні стратегії застосовуються в різних формах [8, 10].

1. Лексикографічна стратегія: відповідно до цієї стратегії продукти порівнюються за найважливішою ознакою. Вибирається продукт, який найкраще відповідає найважливішим характеристикам. Якщо альтернативи оцінюються як однаково хороші за найважливішою ознакою, вони оцінюються за другою найважливішою ознакою. Цей процес триває до тих пір, поки продукт не буде визнаний кращим порівняно з іншими.

2. Стратегія елімінації за аспектами: ця стратегія тісно пов'язана з лексикографічним підходом. Отже, продукцію оцінюють за найважливішою ознакою. Однак тепер споживач використовує обмеження для визначення

альтернатив. Якщо тільки одна альтернатива задовольняє обмеження за найважливішою ознакою, споживач вибирає цей продукт. Якщо кілька альтернатив відповідають межі, процес продовжується на другому найважливішому атрибуті і так далі. Якщо жоден із продуктів не задовольняє вимоги щодо вибраних порогів, споживач повинен переглянути пороги, застосувати іншу стратегію оцінки або відкласти рішення.

3. Кон'юнктивна стратегія: у цій стратегії споживачі також використовують обмеження для процесу прийняття рішень. Від споживача вимагається встановити мінімальні рівні відсічення для кожного основного атрибута. Продукти порівнюються один за одним із усім набором граничних значень. Вибирається продукт, який відповідає всім вимогам. Недотримання заданих граничних рівнів для будь-якого атрибута призводить до відхилення елемента. Як і у випадку стратегії виключення за аспектами, якщо жоден із продуктів не є прийнятним, споживач повинен змінити обмеження, використати іншу стратегію оцінювання або відкласти рішення.

Стратегії компенсаційної оцінки вживають, коли споживач визнає, що погані оцінки за деякими атрибутами можуть компенсуватися відмінними оцінками за інші атрибути. Отже, уявна слабкість атрибута (навіть найважливішого) може бути компенсована іншими атрибутами. Проста адитивна та зважена адитивна є видними формами компенсаційних оцінок стратегії [8, 10].

1. Проста добавка: споживач просто підраховує, скільки разів кожна альтернатива показує себе вигідно порівняно з іншими з точки зору основних характеристик. Вибирається альтернатива з найбільшою кількістю підрахунків. Споживачі застосовують цю стратегію, коли мотивація або можливості обробки обмежені.

2. Зважена добавка: це більш складна форма компенсаційної стратегії. У цьому випадку споживачі використовують ваги, які відображають важливість, яку надають кожному атрибуту. Отже, це вимагає від споживача більшої розумової обробки.

Розуміння цих стратегій є важливим питанням при розробці системи рекомендацій. Ці системи також потребують «стратегії оцінки», щоб визначити, наскільки споживачу сподобається певний продукт. Методи створення рекомендацій можуть варіюватися від дуже простих (наприклад, неперсоналізовані рекомендації на основі статистичних підсумків) до досить складних (наприклад, персоналізовані рекомендації). Наприклад, якщо персоналізовані рекомендації генеруються за допомогою фільтрації на основі атрибутів, стратегії оцінки споживачів тісно пов'язані з алгоритмом класифікації (тобто алгоритмом оцінки ступеня інтересу до продукту або обслуговування). Якщо обраний алгоритм класифікації успішно імітує стратегію оцінювання споживача та прозоро пояснює ці припущення, споживач, швидше за все, прийме рекомендацію.

Наступні два етапи в моделі процесу прийняття рішень споживачем – це покупка та споживання. На рис. 2.4 показано, як в процесі з першої по четверту стадії (тобто розпізнавання потреби, пошук інформації, оцінка перед покупкою та купівля) поєднуються разом.

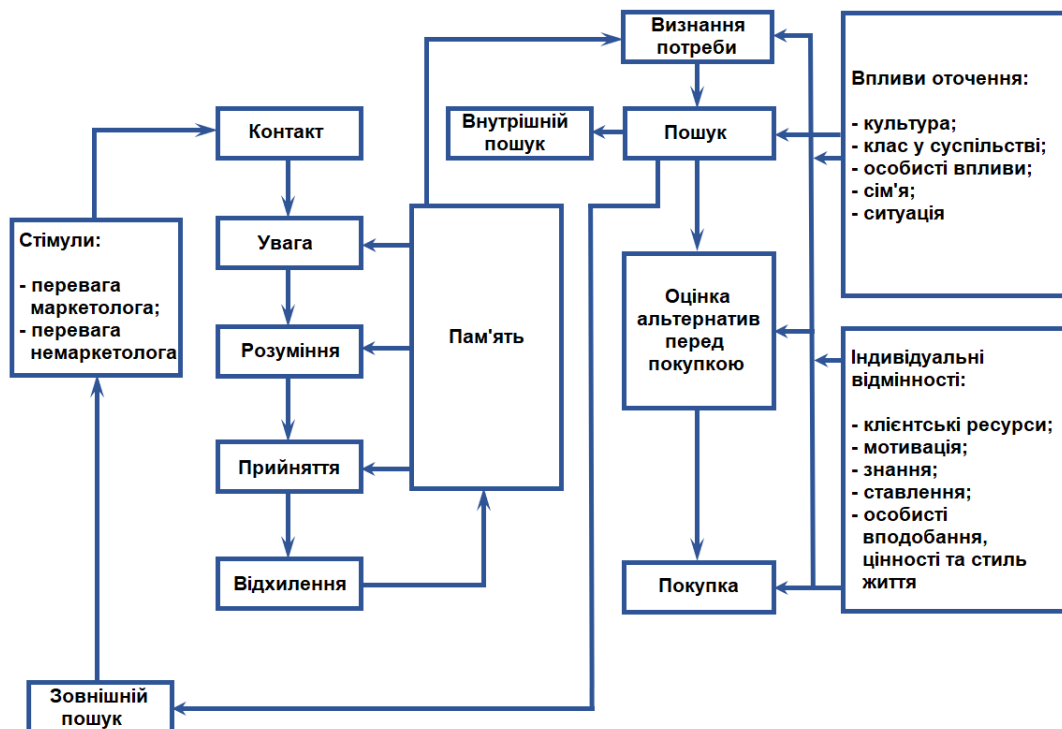


Рисунок 2.4 – Модель процесу прийняття рішення покупцем

У процесі прийняття рішення про покупку споживач вирішує: (1) чи купувати, (2) коли купувати, (3) що купувати, (4) де купувати і (5) як платити.

У цьому процесі рішення можуть прийматися в трьох різних формах [8, 10].

1. Повністю запланована покупка: покупка називається повністю запланованою, коли і продукт, і бренд вибираються заздалегідь [8, 10]. Отже, під час взаємодії з додатком електронної комерції споживач зосереджує свою увагу на конкретному продукті чи послугі. Рекомендаційні системи комерційних додатків можна використовувати як маркетингову тактику для відволікання уваги споживачів на інші бренди. Менш нав'язливим варіантом є демонстрація додаткових продуктів для збільшення перехресних продажів.

2. Частково запланована покупка: споживач знає, який вид продукту він хоче придбати, але конкретний вибір конкретного продукту чи бренду відкладено до покупки [8, 10]. Це типова модель застосування для систем рекомендацій. У цьому випадку рекомендаційні системи спрямовані на те, щоб надати огляд асортименту доступних продуктів і послуг і допомогти знайти відповідну альтернативу. Наприклад, список лідерів продажів можуть бути представлені власнику. Іншою можливістю є використання систем рекомендацій у формі так званих «пошуків продукту», щоб допомогти споживачеві в процесі прийняття рішення. Шукачі продуктів дозволяють споживачеві вказати певні атрибути (зазвичай складні) продукту, які повинен або не повинен мати. Товари, що не відповідають цим вимогам, відфільтровуються з асортименту. Користувач зазвичай визначає атрибути та їх значення самостійно. Це вимагає від споживача базових знань про значення атрибутів. Навпаки, системи фільтрації на основі атрибутів переслідують довгострокову стратегію персоналізації. Ці системи намагаються вивести відповідні атрибути та їхні значення, вивчаючи поведінку користувача. Таким чином, ці системи добре підходять для продуктів із моделями повторних покупок (наприклад, книг).

3. Незапланована покупка: незаплановані покупки відбуваються, коли як тип продукту, так і конкретний продукт або бренд вибираються

спонтанно [8, 10]. Системи рекомендацій також можуть використовуватися для підтримки цієї імпульсивної купівельної поведінки. Дослідження показують, що незаплановані покупки відіграють важливу роль у поїздках за покупками. Отже, системи рекомендацій можуть функціонувати як засіб отримання додаткових доходів у додатках електронної комерції, підтримуючи імпульсивну купівельну поведінку.

Процеси після споживання включають споживання, оцінку після споживання та відчуження. Споживання стосується використання придбаного продукту та послуги. Оцінка після споживання є ще однією фундаментальною частиною моделі процесу прийняття споживачем рішень. Під час і після споживання споживачі формують оцінки щодо продукту та досвіду споживання [8, 10].

Оцінка після споживання має велике значення для компаній. Вона впливає на регулярну купівельну поведінку та формує спілкування користувачів з вуст в уста.

Повторна купівельна поведінка зазвичай виникає, коли споживач задоволений продуктами або послугами. Отже, позитивні оцінки після споживання є вирішальними для утримання споживачів [8, 10]. Це дуже важливо для компаній, оскільки набагато дешевше утримувати старих клієнтів, ніж залучати нових. Системи рекомендацій можуть додатково сприяти утриманню клієнтів щодо електронних постачальників. Якщо рекомендації сприймаються як корисні, вони представляють послугу з доданою вартістю, яка веде до більшої задоволеності клієнтів. Особливо персоналізовані рекомендації, які базуються на довгострокових відносинах навчання, сприяють розвитку відносин між клієнтом і компанією.

Спілкування з вуст в уста є наступним наслідком оцінок після споживання. Споживачі обговорюють свій досвід споживання з іншими – це звичайна діяльність. Зазвичай спілкування з вуст в уста нагадує результат оцінки після споживання. Таким чином, сприятливість спілкування з вуст в уста безпосередньо пов'язана зі сприятливістю досвіду споживання. Здатність

компанії забезпечити задовільний досвід споживання вплине на її здатність утримувати поточних клієнтів, а також залучати нових [8, 10]. У програмах електронної комерції для цілей постачальника може використовуватися передача інформації з вуст в уста. У зв'язку з цим системи рекомендацій у поєднанні з віртуальними спільнотами пропонують спосіб використовувати сарафанне спілкування для цілей рекомендацій, для підвищення довіри та для того, щоб відрізнити постачальника від інших.

Наприклад, якщо надаються рекомендації, можуть відображатися коментарі та оцінки клієнтів, щоб збагатити опис продукту чи послуги постачальника. Це допомагає зміцнити довіру до електронного постачальника. Крім того, деякі методи рекомендацій (наприклад, спільна фільтрація) вимагають рейтингів клієнтів для створення рекомендацій.

Прийняття рішень за умов обмеженої раціональності відкриває двері для різних типів несвідомих впливів на поведінку користувача при прийнятті рішень. Теорії з психології прийняття рішень і когнітивної психології намагаються пояснити ці впливи, наприклад, ефекти приманки та замовчування можуть викликати значні зміни в ймовірності вибору предметів; у сценаріях прийняття групового рішення видимість уподобань інших членів групи може мати значний вплив на остаточне групове рішення.

Складність завдань прийняття рішень, обмежені когнітивні ресурси користувачів і прагнення максимально скоротити загальні зусилля щодо прийняття рішень призводять до феномену обмеженої раціональності, тобто користувачі використовують евристику прийняття рішень, а не намагаються прийняти оптимальне рішення. Крім того, уподобання користувачів, ймовірно, змінюватимуться протягом сеансу рекомендацій, тобто уподобання створюються в конкретному середовищі прийняття рішень, і користувачі не знають своїх уподобань заздалегідь.

Взаємодія з системою рекомендацій означає прийняття різних рішень, як-от вибір книжки зі списку рекомендацій, вибір певного значення функції (наприклад, формат, видавець) як критерію, вибір функцій зворотного зв'язку

для критики під час сеансу рекомендацій на основі критики або вибір доповнення пропозиції для непослідовних уподобань користувача під час взаємодії з експертом, який рекомендує на основі знань. У всіх цих сценаріях користувачі повинні вирішити завдання прийняття рішення. Розв'язати його здатні рекомендаційні системи, які створені за допомогою підходів машинного навчання.

3 ОСНОВНІ ПІДХОДИ ДО РЕАЛІЗАЦІЇ РЕКОМЕНДАЦІЙНИХ СИСТЕМ

Задача рекомендаційної системи – інформувати користувача про товари, які йому можуть бути найбільш цікавими в даний момент часу. Клієнт отримує інформацію, а сервіс заробляє на наданих якісних послугах. Сервіс також може заробляти на комісійних або просто підвищувати лояльність користувачів, яка потім виливається в рекламні та інші доходи.

З моменту популяризації веб-транзакцій стало дедалі легше збирати дані про купівельну поведінку користувачів. Ці дані включають інформацію про профілі користувачів, інтереси, поведінку веб-переглядача, купівельну поведінку та оцінки різних товарів. Цілком природно використовувати такі дані для надання рекомендацій клієнтам щодо можливих інтересів щодо купівлі. Персоналізація онлайн-маркетингу – очевидний тренд останнього десятиліття. За оцінками McKinsey, 35% виручки Amazon або 75% Netflix приходять саме на рекомендовані товари, і цей відсоток, ймовірно, буде рости. Рекомендаційні системи – це те, що пропонує клієнту, щоб зробити його щасливим.

Основні характеристики, за допомогою яких можна описати рекомендаційну систему.

1. Предмет рекомендації – що рекомендується – це можуть бути товари (Amazon), статті (Arxiv.org), новини (Surfingbird), зображення (500px), відео (YouTube, Netflix), люди (Linkedin, LonelyPlanet), музика (Last.fm, Pandora), плейлисти тощо.

2. Мета рекомендацій – навіщо рекомендується – покупка, інформування, навчання, заведення контактів.

3. Контекстні рекомендації – що робить користувач у цей момент – дивиться товари, слухає музику, спілкується з людьми.

4. Джерело рекомендації – хто рекомендує – аудиторія (середній рейтинг ресторану в TripAdvisor), схожі за інтересами користувачі, експертне

суспільство (буває, коли мова йде про складний товар, такий, як, наприклад, вино).

5. Ступінь персоналізації. Неперсональні рекомендації – коли клієнту рекомендують то саме, що й усім. Вони допускають таргетинг за регіоном або часом, але не враховують особисті переваги. Більш розвинений варіант – коли рекомендації використовують дані з поточної сесії клієнта. Він переглянув кілька товарів, і внизу сторінки йому пропонують подібні. Персональні рекомендації включають всю доступну інформацію про клієнта, в тому числі історію його покупок.

6. Прозорість. Люди більше довіряють рекомендації, якщо розуміють, як саме вона була отримана. Так менше ризику натрапити на «недобросовісні» системи, що провокують проплачений товар або ставлять більш дорогі товари вище в рейтингу. Крім того, гарна рекомендаційна система сама повинна вміти боротися з купленими відгуками і накрутками продавців.

7. Формат рекомендацій. Це може бути спливаюче віконце, що з'являється у визначеному розділі сайту, відсортований список, стрічка внизу екрана або що-небудь ще.

8. Алгоритми. Незважаючи на безліч існуючих алгоритмів, всі вони зводяться до кількох базових підходів. До найбільш класичних алгоритмів відносяться Summary-based (неперсональні), Content-based (моделі, засновані на описі товару), Collaborative Filtering (колаборативна фільтрація), Matrix Factorization (методи, засновані на матричному розкладенні матриці) та деякі інші.

У центрі рекомендаційної системи знаходиться так звана матриця переваг. Це матриця, на одній із осей якої відкладені всі клієнти сервісу (Користувачі), а на іншій – рекомендації об'єктів (Елементи) (рис. 3.1). На перетинанні пар (користувачів, елементів) дана матриця заповнена оцінками (Ratings) – це відомий показник зацікавленості користувача в цьому товарі, виражений за заданою шкалою (наприклад, від 1 до 5).

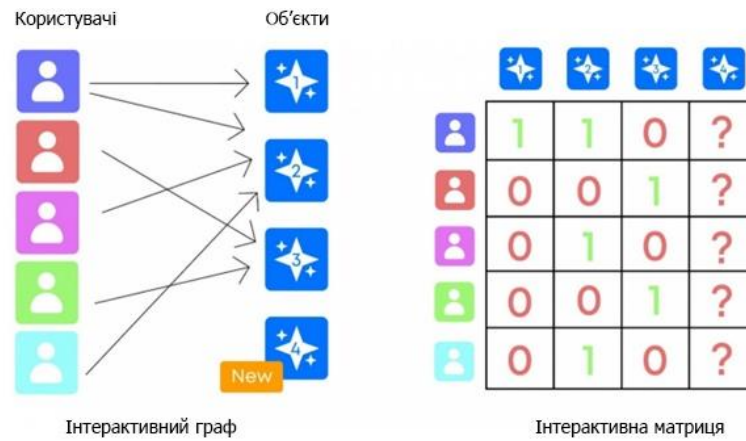


Рисунок 3.1 – Вихідні дані для рекомендаційної системи

Шаблони споживання у людей різні, і не обов'язково потрібно рекомендувати нові товари. Можна показати повторні позиції, наприклад, для поповнення запасу. За цим принципом виділяють дві групи товарів:

- ті, що повторюються, наприклад, шампуні або бритвені станки, які потрібні завжди;
- ті, що не повторюються - книги або фільми, які рідко придбають повторно.

Поняття «цікавості» теж суб'єктивне. Деяким користувачам потрібні речі тільки з їх улюбленої категорії (консервативні рекомендації), а хтось, навпаки, більше відкликається на нестандартні товари або групу товарів (ризиковані рекомендації). Наприклад, відеохостинг може рекомендувати користувачам тільки нові серії улюбленого серіалу, а може періодично пропонувати йому нові шоу або взагалі нові жанри. В ідеалі варто вибирати стратегію показу рекомендацій для кожного клієнта окремо, за допомогою моделювання категорії клієнта.

Користувацькі оцінки можна отримати двома способами:

- явно (явні оцінки) – користувач сам ставить рейтинг товару, залишає відгук, лайкає сторінку;
- неявно (імпліцитні оцінки) – користувач явно своє ставлення не виражає, але можна зробити непрямий висновок із його дій: купив товар – значить, він йому подобається, довго читав опис – значить, є інтерес і т.п.

Звичайно, явні переваги кращі – користувач сам говорить про те, що йому сподобалося. Однак на практиці далеко не всі сайти дають можливість явно виразити свій інтерес, так і не всі користувачі мають бажання це робити. Частіше використовують одразу обидва типи оцінок, які добре доповнюють одна одну.

Також важливо відрізнити терміни Prediction (передбачення ступеня інтересу) і власне Recommendation (показ рекомендації). Що і як показати – це окрема задача, яка використовує отримані на етапі прогнозування оцінки, але може бути реалізована по-різному. Іноді термін «рекомендація» використовується в більш широкому значенні для позначення будь-якої оптимізації, будь то вибір клієнтів для рекламної розсилки, визначення найкращої ціни пропозиції або просто вибір кращої стратегії комунікацій з клієнтом.

Неперсоналізовані рекомендації найбільш прості в реалізації. В них потенційний інтерес користувача визначається середнім рейтингом товару. За цим принципом працює більшість сервісів, коли користувач не авторизується в системі, наприклад, у TripAdvisor.

Показувати рекомендації можна по-різному – як банер з описом товару (Amazon), як результат запити, відсортований за певним параметром (TripAdvisor), або іншим чином.

Рейтинг товару також може відображатися різними способами. Це можуть бути зірочки поруч із товаром, кількістю лайків, різницею позитивних і негативних голосів (як зазвичай роблять на форумах), доля високих оцінок або взагалі гістограма оцінок. Гістограми – найбільш інформативний спосіб, але в них є один мінус – їх складно порівнювати між собою або сортувати, коли потрібно вивести товари списком.

3.1 Проблема «холодного старту»

«Холодний старт» – це типова ситуація, коли ще не накопичено достатню кількість даних для коректної роботи рекомендаційної системи (наприклад, коли товар новий або просто його дуже рідко купують). Якщо середній рейтинг розрахований за оцінками всього трьох користувачів (mykola92, xyz_111 і tandy_s), така оцінка явно не буде достовірною, і користувачі це розуміють. Часто в таких ситуаціях рейтинги штучно коректують.

Перший спосіб корекції – показати не середнє значення рейтингу, а згладжене середнє (Damped Mean): при невеликій кількості оцінок показаний рейтинг більше схиляється до безпечного «середнього» показника, а як тільки набирається достатня кількість нових оцінок, «середня» корекція перестає діяти.

Другий підхід – розраховувати по кожному рейтингу інтервали достовірності (Confidence Intervals): чим більше оцінок, тим менша варіація середнього і, значить, більше впевненості в його коректності. А в якості рейтингу можна вивести, наприклад, нижню межу інтервалу (Low CI Bound). При цьому зрозуміло, що така система буде досить консервативною, з тенденцією до зниження оцінок по новим товарам (якщо, звичайно, це не хіт).

Оскільки оцінки обмежені певною шкалою (наприклад, від 0 до 1), звичайний спосіб розрахунку інтервалу достовірності тут погано застосовний: через хвости розподілу, що знаходяться у нескінченності, і симетричність самого інтервалу. Є альтернативний і більш точний спосіб його розрахувати – довірчий інтервал Вільсона. При цьому отримують несиметричні інтервали (рис. 3.2).

Проблема «холодного старту» так само актуальна і для неперсоналізованих рекомендацій. Загальний підхід тут – замінити те, що в даний момент не може бути враховано, різними евристиками (наприклад, замінити середній рейтинг, використати простіший алгоритм, або взагалі не рекомендувати товар, поки не зберуться дані).

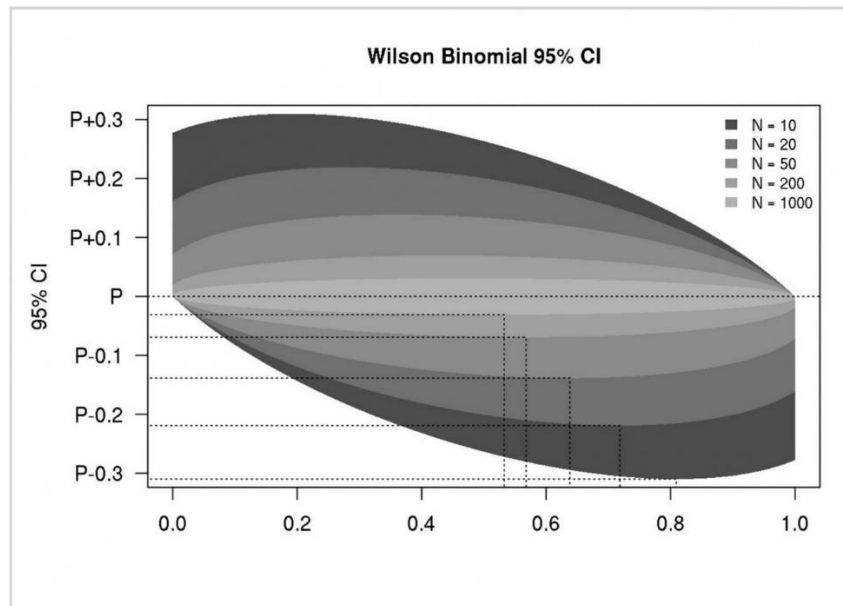


Рисунок 3.2 – Довірчі інтервали Вільсона (по горизонталі відкладена оцінка середнього значення рейтингу, по вертикалі – розкид навколо середнього значення; тоном виділені різні розміри вибірки – чим вибірка більша, тим менший інтервал достовірності)

В деяких випадках також важливо враховувати «свіжість» рекомендацій. Це особливо актуально для статей або повідомлень на форумах. Свіжі записи мають частіше потрапляти в топ. Для цього використовуються коефіцієнти корекції (фактори демпфування) [11, 12].

Приклад розрахунку рейтингу в журналі Hacker news:

$$Rank = \frac{(U - D - 1)^{0.8} \cdot P}{T^{1.8}}, \quad (3.1)$$

де U – голоси за;

D – голоси проти;

P (Penalty) – доповнювальна корекція для впровадження будь-яких бізнес-правил.

Розрахунок рейтингу в Reddit:

$$Rank = \log_{10} \left(\max(1, U - D) \right) - \frac{|U - D|T}{const}, \quad (3.2)$$

де U – кількість голосів «за»;

D – кількість голосів «проти»;

T – час запису.

Перший доданок оцінює «якість запису», а другий робить поправку на час.

Очевидно, що універсальної формули не існує, і кожен сервіс знаходить ту формулу, яка краще вирішує його задачу – перевіряється це емпірично.

Персональні рекомендації передбачають максимальне використання інформації про самого користувача, в першу чергу про його попередні покупки. Одним з перших з'явився підхід фільтрації на основі вмісту *content-based filtering*. В рамках даного підходу опис товару (контенту) співставляється з інтересами користувача, дослідженими з його попередніх оцінок. Чим більше товарів відповідає цим інтересам, тим вище оцінена потенційна зацікавленість користувача. Очевидна вимога тут – у всіх товарів в каталозі повинно бути опис.

Історично предметом змістовних рекомендацій частіше були товари з неструктурованим описом: фільми, книги, статті. Такими описами можуть бути, наприклад, анотації, рецензії, склад акторів тощо. Однак можна використовувати і звичайні чисельні або категоріальні ознаки.

Неструктуровані ознаки описуються типовим для тексту способом – векторами в просторі слів (модель *Vector-Space*). Кожен елемент такого вектора – ознака, яка потенційно характеризує інтерес користувача. Аналогічно, продукт – вектор в тому ж просторі.

У міру взаємодії користувача із системою (скажімо, він купує книжки), векторні описи придбаних товарів об'єднуються (сумуються та нормалізуються) в єдиний вектор i , таким чином, формується вектор його

інтересів. Далі достатньо знайти товар, який описано найбільше близько до вектору інтересів, тобто, розв'язати задачу пошуку n найближчих сусідів.

Не всі елементи опису однаково значимі: наприклад, союзні слова, очевидно, не зовсім корисні завантаження. Тому при визначенні числа елементів, що входять до двох векторів, необхідно попередньо зважувати всі виміри за їх значимістю. Дану задачу вирішує добре відоме в інтелектуальному аналізі тексту перетворення TF-IDF, яке призначає більшу вагу більш рідкісним інтересам. Співпадіння таких інтересів має більше значення при визначенні близькості двох векторів, ніж співпадіння популярних [11, 12]:

$$w_{x,y} = tf_{x,y} \times \log\left(\frac{N}{df_x}\right), \quad (3.3)$$

де $tf_{x,y}$ – частота слова x в описі товару y ;

df_x – кількість товарів, описи яких містять слово x ;

N – загальна кількість товарів.

Принцип TF-IDF так само застосовується до звичайних номінальних атрибутів, таких як, наприклад, жанр, письменник, мова книжки. TF – міра значущості атрибута для користувача, IDF – мера «рідкості» атрибута.

Існує ціле сімейство подібних перетворень (наприклад, BM25 і аналогічні), але змістовно всі вони повторюють ту ж логіку, що TF-IDF: рідкі атрибути повинні мати більшу вагу при сравненні товарів. Рис. 3.3 ілюструє, як саме залежить значення TF-IDF від показників TF і IDF.

Деякі моменти, які корисно врахувати у реалізації:

– при формуванні векторного простору представлення товару замість окремих слів можна використовувати шингли або n -грамми (послідовні пари слів, тройки і т.д.). Це робить модель більш деталізованою, однак вимагає більше даних для навчання;

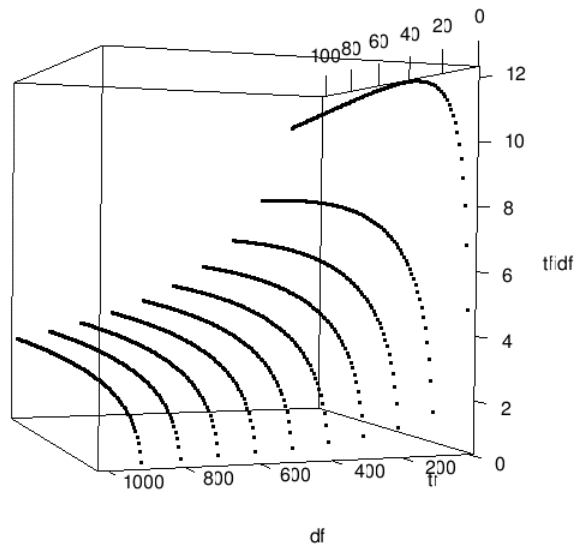


Рисунок 3.3 – Залежність значення TF-IDF від показників TF і IDF (ближня горизонтальна вісь – це DF: частота атрибута серед усіх товарів, дальня горизонтальна вісь – TF: логарифм частоти атрибута в користувача)

- у різних місцях опису товару можуть відрізнитися всі ключові слова (наприклад, опис книжки може складатися із заголовка, анотації та відгуків критиків);

- описи товарів від різних користувачів можна зважувати по-різному. Наприклад, можна дати більшу вагу активним користувачам, у яких багато оцінок;

- аналогічно можна зважувати і товари: чим більше середній рейтинг об'єкта, тим більше його вага (аналог PageRank);

- якщо опис товару допускає посилання на зовнішні джерела, то можна проаналізувати всю пов'язану з товаром зовнішню інформацію.

При додаванні нової оцінки вектор інтересів оновлюється поступово (тільки за тими елементами, які змінилися). При перерахунку має сенс давати новим оцінкам чути більше ваги, оскільки переваги можуть змінитися.

В рамках підходу колаборативної фільтрації рекомендації генеруються на основі інтересів інших схожих користувачів. Такі рекомендації є результатом «співробітництва» великої кількості користувачів. Класичний алгоритм реалізації заснований на принципі k найближчих сусідів (рис. 3.4).

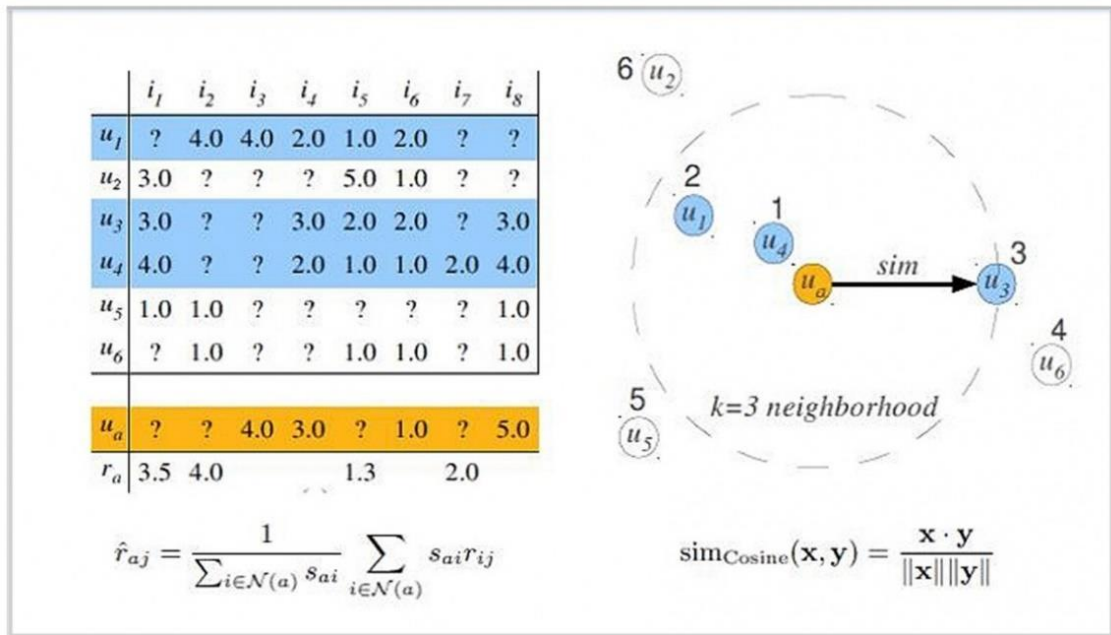


Рисунок 3.4 – Принцип дії алгоритму k найближчих сусідів

«Схожість» – в даному випадку є синонімом «кореляції» інтересів і може розраховуватися безліччю способів (помимо кореляції Пірсона, є ще косинусна відстань, відстань Жаккара, відстань Хеммінга та інші).

У класичному алгоритмі є один явний недолік – він погано застосовний на практиці через квадратичну складність. Фактично, метод вимагає розрахунку всіх попарних відмінностей між користувачами (а користувачів можуть бути мільйони, якщо не мільярди). Неважко порахувати, що складність розрахунку матриці розташувань буде $O(n^2m)$, де n – кількість користувачів, а m – кількість товарів. Для мільйона користувачів для зберігання матриць відмінностей, потрібно мінімум 4 ТБ.

Краще ввести корекцію в алгоритм:

- оновлювати відмінності не при кожній покупці, а батчами (наприклад, раз в день);
- не перераховувати матрицю відмінностей повністю, а оновлювати її поступово;
- обрати ітеративні і наближені алгоритми (наприклад, ALS).

Для того, щоб алгоритм був ефективним, важливо виконати кілька умов: смаки людей не змінюються з часом (або змінюються, але не для всіх однаково); якщо смаки людей співпадають, то вони співпадають у всьому.

Околицю користувача в просторі переваг (його сусідів), яку потрібно аналізувати для генерації нових рекомендацій, можна вибирати по-різному. Можна працювати взагалі з усіма користувачами системи, можна задати деякі пороги, можна вибрати кілька сусідів випадковим чином або вибрати n найбільш схожих сусідів (це найбільш популярний підхід). Автори з MovieLens як оптимальну кількість сусідів виводять цифри в 30-50 сусідів для фільмів і 25-100 для довільних рекомендацій.

Важливий етап підготовки даних – нормалізація оцінок. Оскільки всі користувачі оцінюють по-різному – хтось усім підряд п'ятірки ставить, а від когось четвірки не дочекається – перед розрахунком дані краще нормалізувати, тобто, привести до єдиної шкали, щоб алгоритм міг правильно порівнювати їх між собою. Звичайно, запропоновану оцінку потім потрібно буде перевести у вихідну шкалу зворотним перетворенням (і, якщо потрібно, округлити до найближчого цілого числа).

Нормалізувати можна кількома способами [11, 12]:

- центруванням (mean-centring) – з оцінки користувача вираховують його середню оцінку;
- стандартизацією (z-score) – окрім центрування оцінка поділяється на стандартне відхилення від користувача;
- подвійною стандартизацією – перший раз нормують оцінками користувача, другий раз – оцінками товару.

Схожість або кореляцію переваг двох користувачів можна обчислювати різними способами.

Кореляція Пірсона – класичний коефіцієнт, який цілком застосовний і при зіставленні векторів:

$$\rho = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2 (y_i - \bar{y})^2}}. \quad (3.4)$$

Основний його недолік – коли схожість за оцінками низька, кореляція може бути високою просто випадково.

Для боротьби з випадково підвищеною кореляцією можна домножити на коефіцієнт (Rating intersection) або будь-який інший фактор демпфування, вплив якого зменшується з ростом числа оцінок.

Кореляція Спірмана

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)}. \quad (3.5)$$

Основна відмінність – коефіцієнт ранговий, тобто, працює не з абсолютними значеннями рейтингів, а з їх порядковими номерами. В цілому дає результат дуже близький до кореляції Пірсона.

Косинусна відстань – ще один класичний коефіцієнт:

$$\text{similarity}(A, B) = \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}}. \quad (3.6)$$

Якщо дивитися, косинус кута між стандартизованими векторами – це і є кореляція Пірсона.

Цікавий розвиток колаборативного підходу – так звані рекомендації, засновані на довірі, в яких враховується не тільки близькість людей за інтересами, але також їх «соціальна» близькість і ступінь довіри між ними.

Якщо, наприклад, видно, що на Facebook дівчина періодично заходить на сторінку з аудіозаписами подруги, значить, довіряє її музичному смаку. Тому в рекомендаціях дівчині цілком можна додавати нові пісні з плейлиста подруги.

Важливо, щоб користувач довіряв рекомендаційній системі, а для цього вона має бути простою і зрозумілою. При необхідності завжди повинно бути доступне пояснення рекомендації. В рамках пояснення непогано показувати оцінку товару сусідами, за яким саме атрибутом (наприклад, назва або автор) було співпадіння, а також виводити впевненість системи в оцінці (confidence).

Колаборативна фільтрація користувача (user-based CF) шукає для кожного клієнта групу найбільш схожих на нього (у термінах попередніх покупок) клієнтів і усереднює їх переваги [11, 12]. Ці усереднені переваги і є рекомендаціями для користувача. У випадку з товарною колаборативною фільтрацією (item-based CF) найближчі сусіди шукаються на множині товарів – стовпців матриці переваг. І усереднення відбувається саме з них.

Переваги Item-based перед User-based:

- коли користувачів багато (майже завжди), завдання пошуку найближчого сусіда стає погано обчислюваним;
- оцінка близькості товарів набагато точніша, ніж оцінка близькості користувачів. Це прямий наслідок того, що користувачів зазвичай набагато більше, ніж товарів і, отже, стандартна помилка при розрахунку кореляції товарів там істотно менша;
- у user-based варіанті описи користувачів, як правило, сильно розріджені (товарів багато, мало оцінок). З одного боку, це допомагає оптимізувати розрахунок – перемножуються тільки ті елементи, де є перетин. Але з іншого боку – скільки сусідів не бери, список товарів, які можна порекомендувати, виходить дуже невеликим;
- вподобання користувача можуть змінюватися з часом, але опис товарів більш стійкіший.

Через те, що кореляція продуктів визначається на більшій кількості спостережень, не так критично перераховувати її після кожної нової оцінки і можна це робити періодично в батчевому режимі.

При використанні item-based підходу рекомендації мають тенденцію бути більш консервативними. Справді, розкид рекомендацій виходить меншим і отже меншою є можливість показати нестандартні товари.

Якщо в матриці переваг як рейтинг вживається перегляд опису товару, то рекомендовані товари швидше за все будуть аналогами – товарами, які часто дивляться разом. Якщо ж рейтинги в матриці переваг визначаються на підставі покупок, то, швидше за все, рекомендовані товари будуть аксесуарами – товарами, які часто купують разом.

3.2 Алгоритми факторизації

Було б корисно описати інтереси користувача узагальнено. Це вирішить проблему великої розмірності даних – адже інтереси будуть описуватися не вектором товарів, а значно меншим вектором переваг. Такі підходи називають спектральним розкладанням або високочастотною фільтрацією. Одне з найчастіше використовуваних розкладань називається SVD-розкладом (singular value decomposition).

Метод SVD застосовувався наприкінці 80-х для вибірки схожих за змістом, але не змісту сторінок, а потім став використовуватися і в завданнях рекомендацій. В основі методу – розкладання вихідної матриці рейтингів у добуток 3 матриць [11, 12]:

$$R = U \cdot D \cdot S, \quad (3.7)$$

де розміри матриць $(k, m) = (k, r) * (r, r) * (r, m)$,

r – ранг розкладання – параметр, що характеризує ступінь деталізації розкладання.

Застосовуючи цей розклад до нашої матриці переваг, ми отримуємо дві матриці факторів (скорочених описів):

- U – компактний опис уподобань користувача;
- S – компактний опис характеристик продукту.

Важливо, що при такому підході ми не знаємо, які характеристики відповідають факторам у зменшених описах, для нас вони закодовані якимись числами. Тому SVD є моделлю, що не інтерпретується.

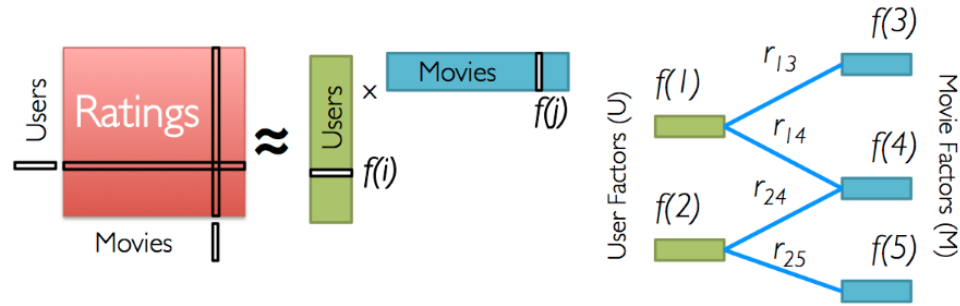
Для того, щоб отримати наближення матриці переваг, достатньо перемножити матриці факторів. Зробивши це, отримуємо оцінку рейтингу для всіх пар клієнт-продукт. Загальна родина подібних алгоритмів називається NMF (non-negative matrix factorization). Як правило, обчислення таких розкладів дуже трудомістке, тому на практиці часто вдаються до їх наближених ітеративних варіантів.

ALS (alternating least squares) – популярний ітеративний алгоритм розкладання матриці переваг на добуток 2 матриць: факторів користувачів (U) та факторів товарів (I). Працює за принципом мінімізації середньоквадратичної помилки на рейтингах. Оптимізація відбувається по черзі, спочатку за факторами користувачів, потім за факторами товарів. Також для обходу перенавчання до середньоквадратичної помилки додаються регуляризаційні коефіцієнти (рис. 3.5).

Якщо доповнити матрицю переваг новим виміром, що містить інформацію про користувача або товари, можна розкласти вже не матрицю переваг, а тензор. Таким чином, буде задіяно більше доступної інформації і, можливо, буде створено більш точну модель.

Асоціативні правила (Association Rules) [11, 12] зазвичай використовуються при аналізі продуктових кореляцій (Market Basket Analysis) і виглядають приблизно так «якщо в чеку клієнта є молоко, то у 80% випадків там буде хліб».

Low-Rank Matrix Factorization:



Iterate:

$$f[i] = \arg \min_{w \in \mathbb{R}^d} \sum_{j \in \text{Nbrs}(i)} (r_{ij} - w^T f[j])^2 + \lambda \|w\|_2^2$$

Taken from the BerkeleyX Course Big Data Analysis with Spark

Рисунок 3.5 – Схема дії алгоритму низькорангової матричної факторизації

Це не те саме, що аналіз рознесених у часі покупок, але якщо ми вважатимемо всю історію одним великим кошиком, то цілком можемо застосувати цей принцип і тут. Це може бути виправданим, коли ми, наприклад, продаємо дорогі разові товари (кредит, політ).

Обмежені машини Больцмана RBM (restricted Boltzman Machines) – відносно старий підхід, що ґрунтується на стохастичних рекурентних нейронних мережах [11, 12]. Він представляє собою модель з латентними змінними і в цьому схожий на SVD-розкладання. Тут також створюється найбільш компактний опис користувацьких переваг, що кодується за допомогою латентних змінних. Метод не був розроблений для пошуку рекомендацій, але він успішно використовувався у топових рішеннях Netflix Prize і досі застосовується у деяких завданнях.

Автоенкодери (autoencoders) – в основі лежить той самий принцип спектрального розкладання, тому такі штучні нейронні мережі ще називають denoising auto-encoders. Мережа спочатку згортає відомі їй дані користувача в деяке компактне подання, намагаючись залишити лише значну інформацію, та потім відновлює дані до вихідної розмірності. У результаті виходить певний

усереднений, очищений від шуму шаблон, яким можна оцінити інтерес до будь-якого продукту.

DSSM (deep semantic similarity models) – один із нових підходів (рис. 3.6). Той самий принцип вживання штучних нейронних мереж, але в ролі латентних змінних тут внутрішні тензорні описи вхідних даних (embeddings). Спочатку модель створювалася для матчингу запиту з документами (як і content-based рекомендації), але легко трансформується в завдання матчингу користувачів і товарів.

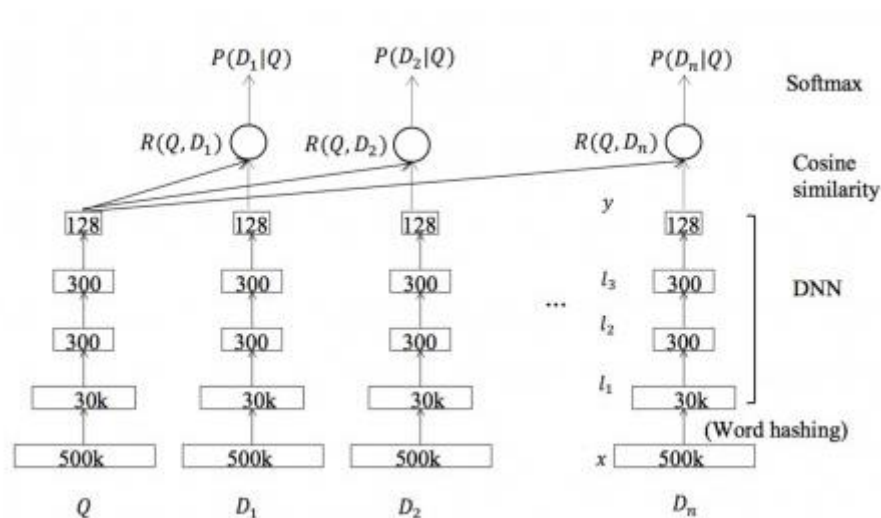


Рисунок 3.6 – Архітектура глибокої структурованої семантичної моделі DSSM

Різноманітність архітектур глибоких мереж безмежна, тому Deep Learning надає справді широке поле для експериментів у сфері рекомендаційних систем.

Є й інші підходи отримання ембедингів з розрідженої матриці взаємодій користувачів з об'єктами. Один із них запозичений з мовних NLP-моделей і зветься word2vec. Досить просто уявити, що об'єкти – це слова, а послідовності взаємодії з ними у рамках однієї сесії (або користувача) – це пропозиції. Порівняно з попередніми методами, NLP-моделі використовують ковзне вікно невеликого розміру, отже, дозволяють враховувати ще послідовність дій, але близькість одного до іншого в послідовності.

Існують два методи формування train-набору даних. Один називається skip-gram – це вгадування оточення навколо слова. Інша схема - це CBOW (continuous bag of words), вгадування слова щодо його оточення (слів навколо) (рис. 3.7).

Jay was hit **by a red bus in**...

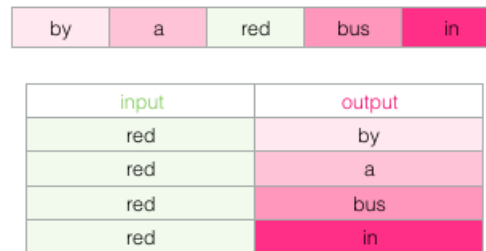


Рисунок 3.7 – Принцип дії суцільного мішка слів

Суть одна – виділити патерни слів, які часто зустрічаються разом в одному контексті. Виходить модель, яка бере вхідне та вихідне слова та обчислює ймовірність їхнього сусідства. Але потрібно ввести в набір даних негативні зразки – слова, які не є сусідами. Це називається Skip-gram із негативною вибіркою (SGNS).

Мета на етапі навчання – підібрати такі ембедінги для всіх слів словника, щоб скалярний добуток їх був мірою ймовірності для цих слів зустрітися по сусідству.

Ключові гіперпараметри word2vec: розмір вікна, кількість негативних зразків та розмірність ембедінгів. У Gensim (відомій бібліотеці для обробки природної мови, де реалізований алгоритм word2vec) розмір вікна за замовчуванням дорівнює 5, і 5 негативних зразків береться на кожен позитивний.

Загальна архітектура CBOW-мережі представлена малюнку. Вхідний шар може бути розширений залежно від кількості слів у контексті. Використовується розріджене one-hot encoding уявлення слів (рис. 3.8).

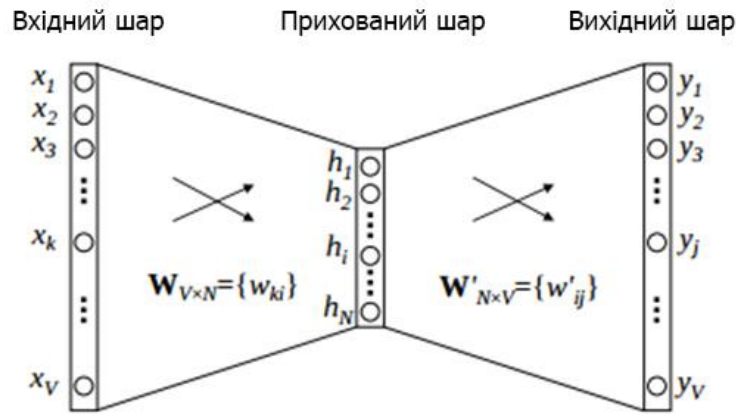


Рисунок 3.8 – Архітектура штучної нейронної SBOW-мережі

Розглянемо конкретне застосування у рекомендаціях з прикладу букингового агентства. У букинговій агенції використовується skip-gram model. Якщо формально, то функція L , яку максимізують, виглядає так:

$$L = \sum_{s \in S} \sum_{l_i \in s} \left(\sum_{-m \leq j \leq m, i \neq j} \log P(l_{i+j} | l_i) \right). \quad (3.8)$$

S – це сесії, кожна з яких складається з послідовності кліків по 1 лістингах (об'єктах розміщення), v – ембединги, що навчаються:

$$P(l_{i+j} | l_i) = \frac{\exp(\mathbf{v}_{l_i}^T \mathbf{v}'_{l_{i+j}})}{\sum_{l=1}^{|\mathbf{V}|} \exp(\mathbf{v}_{l_i}^T \mathbf{v}'_l)}. \quad (3.9)$$

Імовірність P – це можливість побачити лістинг l_{i+j} на відстані, яку вважають контекстом від лістингу l_i . Де m – це параметр, що відображає розмір контексту. v_l і v_l зі штрихом - це вхідний та вихідний вектор ембедингів лістингу l . За такого підходу лістинги, які у подібних контекстах, отримають схожі ембединги (як і зі словами). Щоб не брати до уваги градієнт по всьому

«словнику» лістингів, використовується підхід позитивного та негативного семплінгу пар таргет-контекст.

З цікавих рішень: щоб зробити додатковий акцент на сесіях, що закінчилися бронюванням, бронювання додається як глобальний (додатковий) контекст до всіх тренувальних прикладів із сесій із бронюванням незалежно від розміру вікна (рис. 3.9).

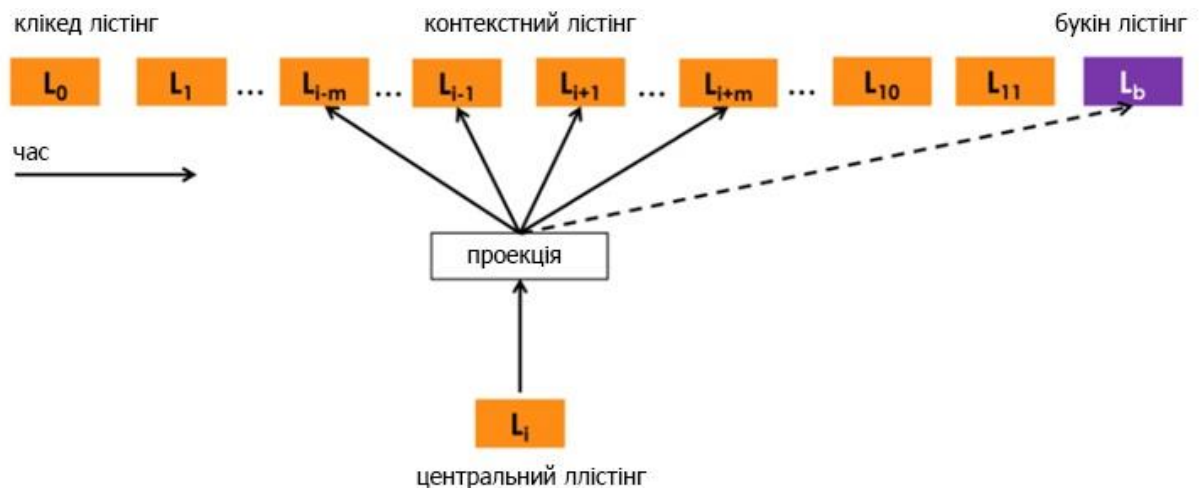


Рисунок 3.9 – Додавання контексту до тренувальних прикладів під час навчання моделі word2vec

Є ще одна цікава модифікація, яку використовують для букінгових агенцій. Замість word2vec тут використовується DeepWalk, який використовує ту ж ідею Skip-gram, але для складання ембедингів графових структур. Його ключова особливість - це випадкові блукання, які використовуються замість звичайного фіксованого контексту Word2vec (рис. 3.10).

Насправді, рідко використовується лише один підхід. Як правило, кілька алгоритмів комбінуються в один, щоб досягти максимального ефекту. Головні переваги об'єднання моделей – це збільшення точності та можливість більш гнучкого налаштування на різні групи клієнтів. Недоліки – менша інтерпретованість та більша складність реалізації та підтримки.

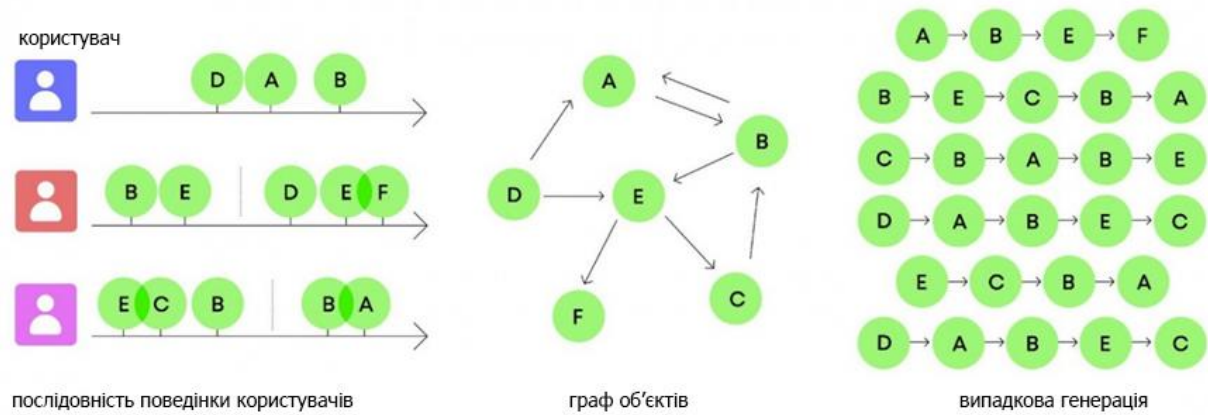


Рисунок 3.10 – Принцип дії моделі DeepWalk

Декілька стратегій об'єднання:

- **Weighting** – вважати середньозважений прогноз за декількома оцінками;
- **Stacking** – передбачення окремих моделей є входами іншого (мета)класифікатора, який навчається правильно зважувати проміжні оцінки;
- **Switching** – для різних продуктів/користувачів застосовувати різні алгоритми;
- **Mixing** – обчислюються рекомендації з різних алгоритмів, а потім просто об'єднуються в один список.

Наприклад, використовується content-based recommender, а як одна з фіч – результат колаборативної фільтрації.

Feature Weighted (Linear) Stacking:

$$P(u, i) = w_1 P_1(u, i) + w_2 P_2(u, i) + \dots + w_n P_n(u, i). \quad (3.10)$$

Ваги w_1, w_2, \dots, w_n навчаються на вибірці. Зазвичай, цього використовується логістична регресія.

Stacking у загальному вигляді (рис. 3.11):

$$P(u,i) = f_1(u,i)P_1(u,i) + f_2(u,i)P_2(u,i) + \dots + f_n(u,i)P_n(u,i). \quad (3.11)$$

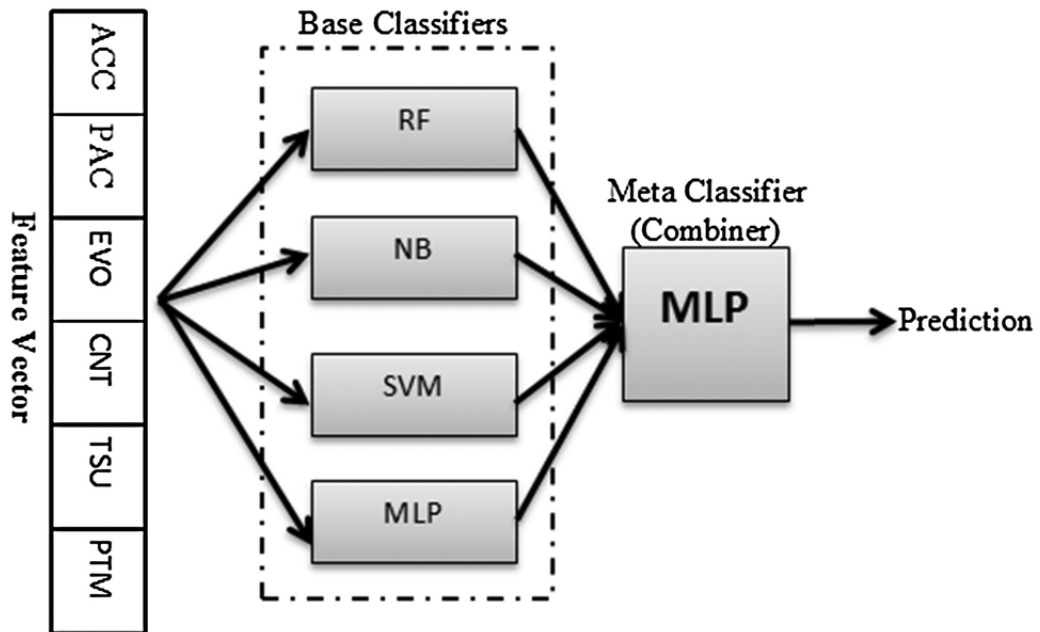


Рисунок 3.11 – Архітектура ансамблю класифікаторів для рекомендаційної системи

Незважаючи на широкий спектр сучасних алгоритмів машинного навчання, які підвищують точність та швидкодію рекомендаційних систем, основною проблемою для них залишається проблема «холодного старту», коли відомостей про вподобання користувача ще замало.

Особливо складною ця проблема стає для рекомендацій під час продажів книг, де відмінності між жанрами достатньо розпливчасті, і спиратися на схожість користувачів, наприклад, за територіальною ознакою, буде помилкою.

3.3 Оцінка якості системи

Усі метрики якості можна умовно поділити на три категорії [11, 12]:

- Prediction Accuracy – оцінюють точність прогнозованого рейтингу;
- Decision support – оцінюють релевантність рекомендацій;

– Rank Accuracy метрики – оцінюють якість ранжування рекомендацій, що видаються.

На жаль, не існує єдиної рекомендованої метрики на всі випадки життя і кожен, хто займається тестуванням рекомендаційної системи, підбирає її під цілі.

Коли рейтинги оцінюються за безперервною шкалою (0-10), зазвичай досить метрик класу Prediction Accuracy. Це середнє абсолютне відхилення (MAE – Mean Absolute Error); середньоквадратична помилка (MSE – Mean Squared Error); корінь із середньоквадратичної помилки (RMSE – Root Mean Squared Error).

Метрики класу Decision Support працюють з бінарними даними (0 та 1). Якщо в завданні рейтинги спочатку відкладаються за безперервною шкалою, їх можна перевести в бінарний формат, застосувавши вирішальне правило – скажімо, якщо оцінка менша за 3.5, вважаємо оцінку «0», а якщо більшою, то «1».

Частка рекомендацій, що сподобалися користувачеві [11, 12]:

$$Precision = \frac{TP}{TP + FP}; \quad (3.12)$$

частка цікавих для користувача товарів, яка показана

$$Recall = \frac{TP}{TP + FN}; \quad (3.13)$$

середнє гармонійне метрик Precision та Recall:

$$F1 - Measure = \frac{2PR}{P + R}. \quad (3.14)$$

ROC AUC – наскільки високою є концентрація цікавих товарів на початку списку рекомендацій;

Precision@N – метрика Precision, порахована на Top-N записях [11, 12];

$$Precision @ k = \frac{1}{k} \sum_{i=1}^k Relevance @ i, \quad (3.15)$$

Recall@N – метрика Recall, порахована на Top-N записях;

AverageP – середнє значення Precision на всьому списку рекомендацій.

$$AP @ N = \frac{1}{m} \sum_{i=1}^N (P(k) \text{ if } k^{\text{th}} \text{ item was relevant}) = \frac{1}{m} \sum_{i=1}^N P(k) \cdot rel(k). \quad (3.16)$$

Зазвичай, рекомендації виводяться списком із кількох позицій (спочатку топова, потім у порядку зменшення пріоритету). Метрики класу Rank Accuracy вимірюють наскільки правильним є порядок показу рекомендацій у відсортованому списку.

На якій позиції списку рекомендацій користувач знаходить першу корисну [11, 12]:

$$Mean Reciprocal Rank = E \left(\frac{1}{pos} \right). \quad (3.17)$$

Кореляція (Спірмена) реального та прогнозованого рангів рекомендацій

$$Spearman Correlation = E \left(|P - R|^2 \right). \quad (3.18)$$

Інформативність видачі з урахуванням ранжування рекомендацій [11, 12].

$$nDCG = \sum \frac{R(i)}{\max(1, \log(i))}. \quad (3.19)$$

Наскільки високою є концентрація цікавих товарів на початку списку рекомендацій

$$FractionOfConcordancePairs = P(X_R > X_P). \quad (3.20)$$

Як і в будь-якій моделі, спрямованій на мотивацію користувача до дії, слід оцінювати тільки інкрементальний приріст цільової дії. Тобто, наприклад, при підрахунку покупок за рекомендацією потрібно виключити ті, які користувач і так зробив би без моделі. Якщо цього не зробити, ефект від застосування моделі буде сильно завищений.

Поведінка користувача погано формалізована і жодна метрика повною мірою не опише розумові процеси у його голові під час виборів товару. На рішення впливає багато чинників. Перехід за посиланням з товаром, що рекомендується, ще не є його високою оцінкою або навіть інтересом. Частково зрозуміти логіку клієнта допомагає онлайн-тестування. Нижче наводиться пара сценаріїв такого тестування.

Перший і найочевидніший сценарій – аналіз подій сайту. Ми дивимося, що користувач робить на сайті, чи звертає увагу на рекомендації, чи переходить за ними, які фічі системи мають попит, які – ні, які товари краще рекомендуються, які гірші.

Другий сценарій – це отримання фідбеку від користувачів у вигляді опитувань та голосувань. Як правило, це загальні питання для розуміння, як клієнти користуються сервісом – що важливіше: релевантність або різноманітність, чи можна показувати продукти, що повторюються, або це занадто дратує. Перевага сценарію – він дає пряму відповідь на всі ці питання.

3.4 Складнощі моделей

Складність побудови моделей у необхідності одночасно задовольняти різні критерії якості, а також у складності підготовки та інтерпретації наявних даних. Неповний перелік конкретних проблем:

- різноманітність типів фідбека (тривалість, лайки, кліки, додавання до обраного) та його неоднозначність. Клік, але не лайк і не покупка це що? Подобається результат користувачеві чи ні?;

- різноманітність типів і тем контенту;
- холодний старт користувача;
- холодний старт контенту;
- короткостроковість трендів та старіння контенту;
- глобальна популярність;
- оновлення рекомендацій (для користувачів з 100500 процесів, 100500+1 мало змінює видачу);
- врахування послідовності взаємодії (смаки змінюються);
- врахування сезонності (на Новий рік, на Великдень чи 14 лютого у тренді будуть різні фільтри);
- включення різних типів інформації (щоб враховувати взаємодії і сам контент: картинки і текст, і навіть метадані користувача: геолокацію, платоспроможність та інше).

Інфраструктурні складнощі здебільшого виростають із необхідності постійного оновлення моделей. На практиці поширеними проблемами стають:

- масштабованість, швидкість видачі рекомендацій, кешування, навантаження на сервери;
- оновлення рекомендацій (швидке реагування на останні дії, новий контент);
- регулярне оновлення моделей;
- правильне проведення тестів.

Але головна, мабуть, проблема, у тому, що, попри всю схожість завдань у цій галузі, в кожного сервісу залишається своя специфіка. Важливо точно визначати критичні сервісу особливості рекомендаційної системи. Наприклад: у музичних сервісів композиції можуть слухати багаторазово, а фільм навіть найцікавіший рідко дивляться двічі. Рекламні рекомендації в соціальних мережах потрібно оптимізувати з урахуванням геотаргетингу (для придбання їжі не замовляють доставку з одного міста до іншого, але придбані книжки можна переправляти з міста до міста, навіть з інших країн).

4 ЕКСПЕРИМЕНТАЛЬНА ЧАСТИНА

4.1 Вибір програмного середовища

Для дослідження ефективності запропонованих рішень під час підбору нових видань було обрано програмне середовище Node.js [13].

Node.js — це серверне середовище з відкритим кодом, яке працює на різних платформах за допомогою JavaScript і дозволяє виконувати код JavaScript на стороні сервера. Однією з найважливіших переваг розробки Node.js є те, що її можна використовувати для створення широкого спектру рішень. Основні напрямки, де використання Node.js дає безумовні переваги:

- розробка додатків потокової передачі даних: за допомогою вбудованого потокового модуля веб-додатку Node.js можна обробляти великі фрагменти даних у менших пакетах і послідовно;

- для розробки мікросервісів: природа Node.js дозволяє розробнику легко створювати масштабовані рішення; якщо очікується, що додаток зазнаватиме високих пікових навантажень і масштабованості, відповіддю є послуги розробки додатків Node JS. Раніше Node.js пропонувався лише для додатків із невеликою кількістю завдань, які інтенсивно використовують ЦП. Однак після появи версії 10.5.0 стали доступними потоки для виконання таких складних процесів;

- для розвитку SPA: Node.js може створити односторінкову програму (Single Page Application - SPA), яка виглядає як програма для робочого столу, так само як односторінкові веб-сторінки. В основі роботи Node.js лежить асинхронність. Окрім неблокуючого вводу-виводу, це дозволяє програмі на стороні сервера на Node.js обробляти набагато більше клієнтських запитів за одиницю часу, ніж аналогічна програма, розроблена на основі більшості інших технологій розробки на стороні сервера.

Крім того, Node.js особливо корисний для обмеження циклу виходу на ринок, оскільки може значно скоротити час розробки додатків, не змінюючи наявні функції. З Node можна миттєво перейти від концепції до кінцевого продукту, не збільшуючи істотно вартості продукту. Гіпотези, сформовані під час початкового дослідження ринку, перевірятимуться швидше, коректування продукту відбуватиметься швидше, а кошти, зрештою, швидше обертаються. Це дуже важлива характеристика для інвесторів проекту.

Масштабованість: Node.js створено з урахуванням масштабованості. Це одна з найважливіших функцій Node для стартапів, які прагнуть масштабуватися з часом. Стартапи, що базуються на додатках, віддають перевагу розробці легких і швидких систем, що працюють у реальному часі, які можна згодом розширити та додати більше модулів чи структур до існуючих.

Node.js працює як для Front-end, так і для Back-end тому, що є універсальною мовою програмування. Ця базова функція полегшує виконання проектів, оскільки знайти команду кваліфікованих повних розробників може бути складно. До складу команди розробників увійдуть фахівці з більш широким поглядом на проблему, а значить, навантаження на керівництво буде зменшено. Процес розробки буде більш ефективним, тому що людині (full-stack розробнику) завжди простіше і швидше домовитися з собою, ніж з іншою людиною. Це зменшує витрати на спілкування між фахівцями.

Величезна спільнота: середовище є відкритим кодом, тому можна уникати ліцензійних зборів; багато активних ентузіастів Node.js і велика кількість розробників програмного забезпечення з Microsoft, SAP, IBM та багатьох інших регулярно вносять свій внесок у спільноту Node.js, яка постійно розвивається, щоб допомагати іншим розробникам.

Багата екосистема: екосистема пакетів Node, яка включає NPM (керування пакетами вузлів), є найбільшим у світі реєстром програмного забезпечення, який найшвидше розвивається. Він містить багато бібліотек і багаторазових шаблонів, які можна використовувати у своєму кодї, щоб

додати більше цінності з меншими ресурсами та часом. В екосистемі NPM є модуль, доступний майже для всього, що може придумати розробник програмного забезпечення.

Node.js є легким і ефективним для програм, яким потрібне безперервне з'єднання від браузера до сервера, що дозволяє їм негайно оновлюватися новими публікаціями або діями користувача. Прикладами можуть бути:

- Інтернет-магазини;
- послуги розміщення відео та аудіо;
- месенджери;
- соціальна мережа;
- будь-які продукти b2c.

Для всіх цих напрямків Node дозволяє реалізувати швидкий старт, а потім швидко, відносно недороге горизонтальне масштабування. Саме такі властивості відзначають систему підбору книжкових новинок, яку можна підключити, наприклад, до Інтернет-магазину, або залишити як самостійний додаток.

4.2 Реалізація системи підбору книжкових новинок

Механізм прийняття рішень для агрегації новин про книжкові видання реалізований для невеликого інтернет-магазину (кількість покупців приблизно 500 тисяч) та працює з даними, які надають користувачі. На основі цих даних можна робити пропозиції користувачеві. В даному випадку сценарій проходитиме наступні кроки:

а) навчання:

- 1) форматування даних в придатний для використання стан;
- 2) обчислення TF-IDF і створення векторів з відформатованих документів;
- 3) розрахунок показників схожості подібних документів.

б) використання навчених даних для створення рекомендації на основі вибору користувача.

Завдяки можливостям обраного програмного середовища Node.js, сценарій можна реалізувати в асинхронному режимі, тобто, форматування даних, обчислення ступеня схожості документів та навчання системи виконується офлайн, а в реальному часі проводиться взаємодія з користувачем (рис. 4.1).

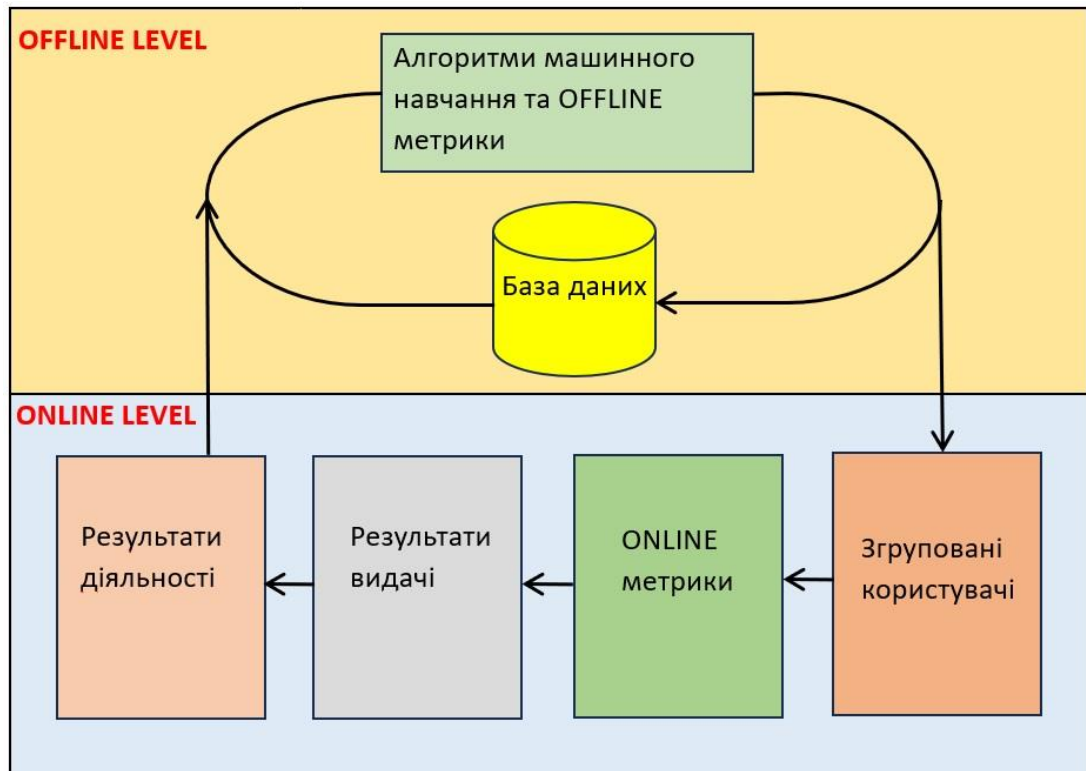


Рисунок 4.1 – Асинхронний режим дії системи агрегації новин про книжкові видання

Для розв’язання проблеми «холодного старту» в роботі пропонується вживати комбінацію «user-based» та «item-based» підходів. В режимі офлайн обробляється інформація про тих користувачів, які певний час були клієнтами сервісу та обирали певні книжки. Зібрані дані форматуються у вигляді таблиці, де стовпці відповідають певним атрибутам: ім’я, вік, користувацький рейтинг тощо (рис. 4.2). Враховуються також покупки книжок одного жанру (якщо вони є, то краще одразу вказати піджанр, якщо немає, то виконується перехід до етапу 2). На цьому ж етапі користувачі оцінюють книжки та коментують їх.

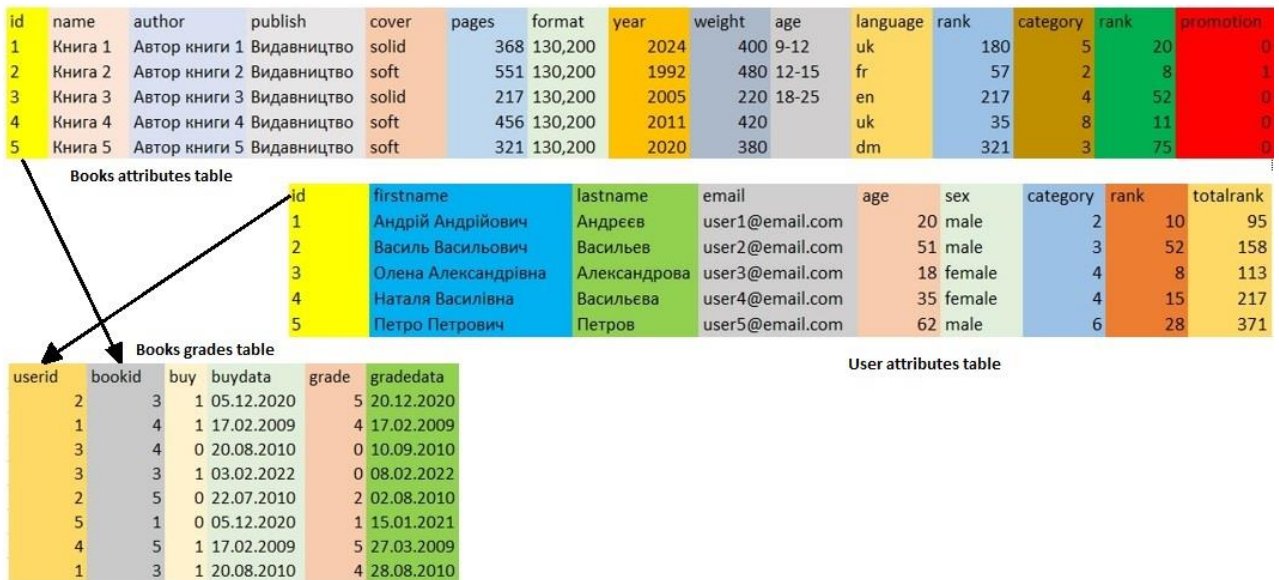


Рисунок 4.2 – Схема процесу агрегації новин про книжкові видання

Таким чином, тепер можна поділити користувачів на групи.

1 група - ті, хто купують книги в певному жанрі (піджанрі) постійно (наприклад, мильні опери для молодих дівчаток на пару вечорів або в потяг, або фантастика з подорожами в космосі, або про зомбі-апокаліпсис).

2 група – ті, хто купував книжки у цьому жанрі, але не так часто, тобто, можна припустити, що покупка книг була лише після того, як книга отримала певний рейтинг.

3 група – ті, хто купував книги цього жанру хоча б один раз (наприклад, у подарунок іншій людині).

4 група – ті, хто не купував до сих пір жодної книжки цього жанру, тобто, дана книга вже здобула настільки високий рейтинг, що зацікавила далеких від цього жанру користувачів.

5 група – нові користувачі магазину. Книга також рекомендується як і в групі 4.

Схожим чином обробляється інформація про книжкові новинки, що надходить до системи або з видавництва, або з книжко-розповсюджувальних представництв. Таблиця відомостей про книжки містить значення таких атрибутів, як назва видавництва, тип обкладинки, кількість сторінок, формат,

рік видання, рекомендовану вікову категорію читачів, рейтинг вподобань та інше (рис. 4.2).

Тепер в режимі онлайн має відбуватися взаємодія з користувачем з урахуванням наявних даних в системі.

Сценарій взаємодії.

Крок 1. Приходить нова книжка. У книги є характеристики, жанр (піджанр), обсяг, формат, автор, видавництво, палітурка і т.д. За цими показниками можна одночасно порадити книгу групі користувачів 1. Користувачі групи 1 купують і оцінюють книгу. Рейтинг перераховується лише на рівні OFFLINE, оскільки це вимагає обчислювальних потужностей, а магазин невеликий і багато потужностей не має.

Крок 2. Після перерахунку в книжки з'являється рейтинг. Тепер можна порадити книгу користувачам із групи 2. Користувачі книгу купують та оцінюють.

Крок 3. Перерахунок рейтингу на рівні OFFLINE (рис. 4.1), книжку можна пропонувати користувачам із групи 3.

Крок 4. Перерахунок рейтингу на рівні OFFLINE (рис. 4.1), книжку можна пропонувати користувачам із групи 4 та 5.

На кожному рівні можна встановлювати максимальний рейтинг для переходу книги на наступний рівень користувачів. Тобто, може вийти так, що одну книгу можна буде одразу пропонувати на 3 рівні, а інша може і для 2-го не дійти. Різниця між 3 і 4 рівнями груп користувачів дуже мала, оскільки користувачі з групи 3 можуть книгу купити, але не оцінити, а можуть і не купити, і не оцінити. Тобто, перерахунок рейтингів книжок OFFLINE доводиться виконувати періодично на всіх рівнях, щоб піднімати рейтинг за рахунок нових оцінок та покупок.

У процесі рекомендацій можна застосовувати додаткові ознаки, наприклад, вік або вікова група читача, книга-подарунок, художня палітурка або книга для сліпих та інші випадки.

Реалізований цей сценарій в системі завдяки інтерфейсу, де користувачі налаштовують вибір показників для підбору рекомендацій, а також обирають потрібну дію: купити книгу, оцінити чи коментувати її (рис. 4.3).

Рисунок 4.3 – Інтерфейс системи агрегації новин про книжки

Для визначення відносної важливості терміну використовуються поняття термінової частоти (TF) і інверсної частоти документа (IDF). З цим використовується концепція косинусної подібності (формула (3.6)). TF – це частота появи слова в документі. IDF – частота терміну в цілому корпусі документів. Це означає рідкість слова та допомагає підвищити кількість рідкісних термінів. TD-IDF використовується тому, що ця модель поєднує важливість слова в документі (локальне значення) з тим, наскільки важливим є це слово в усьому корпусі (глобальне значення). В даному контексті вектори значень атрибутів будуть об'єктами, що містять термін як ключ і TF-IDF як значення.

Першим кроком у «навчанні» системи є форматування даних у структуру, якою можна користуватися та якою легко керувати. Необхідно, щоб усі описи були в одному рядку. Це дозволить легше обробляти їх пізніше.

```
// Sample user-item interactions data
const userItemInteractions = {
  user1: { item1: 5, item2: 3, item3: 4 },
  user2: { item1: 4, item2: 2, item3: 5 },
  user3: { item1: 2, item2: 5, item3: 4 },
  // Add more users and items as needed
};
```

Наступним кроком є обчислення TF-IDF документів і створення вектора, що містить термін як ключове значення (вектор) TF-IDF. На цьому кроці також виконується обчислення схожості між користувачами за допомогою косинусної подібності та скалярного добутку.

```
// Function to calculate user similarities using cosine similarity
function calculateUserSimilarities(interactions) {
  const userSimilarities = {};

  // Loop through users
  for (const user1 in interactions) {
    userSimilarities[user1] = {};
    for (const user2 in interactions) {
      if (user1 !== user2) {
        const itemsForUser1 = Object.keys(interactions[user1]);
        const itemsForUser2 = Object.keys(interactions[user2]);

        // Calculate common items between users
```

```

const commonItems = itemsForUser1.filter(item =>
itemsForUser2.includes(item));

if (commonItems.length > 0) {
  // Calculate cosine similarity
  const sumProduct = commonItems.reduce((acc, item) => acc +
interactions[user1][item] * interactions[user2][item], 0);

  const sumSquaredUser1 = itemsForUser1.reduce((acc, item) => acc +
Math.pow(interactions[user1][item], 2), 0);

  const sumSquaredUser2 = itemsForUser2.reduce((acc, item) => acc +
Math.pow(interactions[user2][item], 2), 0);

  const similarity = sumProduct / (Math.sqrt(sumSquaredUser1) *
Math.sqrt(sumSquaredUser2));

  // Store the similarity value
  userSimilarities[user1][user2] = similarity;
}
}
}
}

return userSimilarities;
}

```

Аналогічно відбувається сортування книжок за схожістю ознак.

```

// Function to calculate item similarities using cosine similarity
function calculateItemSimilarities(interactions) {
  const itemSimilarities = {};

```

```

// Loop through items
for (const item1 in interactions) {
  itemSimilarities[item1] = {};
  for (const item2 in interactions) {
    if (item1 !== item2) {
      const usersForItem1 = Object.keys(interactions[item1]);
      const usersForItem2 = Object.keys(interactions[item2]);

      // Calculate common users between items
      const commonUsers = usersForItem1.filter(user =>
usersForItem2.includes(user));
      if (commonUsers.length > 0) {
        // Calculate cosine similarity
        const sumProduct = commonUsers.reduce((acc, user) => acc +
interactions[item1][user] * interactions[item2][user], 0);
        const sumSquaredItem1 = usersForItem1.reduce((acc, user) => acc +
Math.pow(interactions[item1][user], 2), 0);
        const sumSquaredItem2 = usersForItem2.reduce((acc, user) => acc +
Math.pow(interactions[item2][user], 2), 0);
        const similarity = sumProduct / (Math.sqrt(sumSquaredItem1) *
Math.sqrt(sumSquaredItem2));

        // Store the similarity value
        itemSimilarities[item1][item2] = similarity;
      }
    }
  }
}
return itemSimilarities;
}

```

Нарешті, ми сортуємо їх так, щоб елемент із найбільшою косинусною подібністю був першим у масиві. Саме його система і порадить обрати користувачеві.

```
// Function to get item recommendations for a user
function getItemRecommendations(user, itemSimilarities, interactions) {
  const userItems = interactions[user];
  const recommendations = {};

  // Loop through user items
  for (const userItem in userItems) {
    // Loop through similar items
    for (const similarItem in itemSimilarities[userItem]) {
      // Check if user hasn't interacted with the similar item
      if (!(similarItem in userItems)) {
        // Calculate recommendation score
        if (!recommendations[similarItem]) {
          recommendations[similarItem] = 0;
        }
        recommendations[similarItem] +=
itemSimilarities[userItem][similarItem] * userItems[userItem];
      }
    }
  }

  return recommendations;
}
```

Повний текст програмного коду наведений у Додатках А та Б.

4.3 Оцінювання якості запропонованої системи агрегації новин про книжкові видання

Щоб визначити, наскільки корисною буде запропонована система, що дозволить користувачам обирати книжки з новинок, вживається метод експертного оцінювання на основі порівняння з існуючими Інтернет-сервісами:

- Клуб сімейного дозвілля КСД [14];
- Book24.ua [15];
- Yakaboo [16];
- Vukva.ua [17].

Незважаючи на те, що пошукові системи означених інтернет-магазинів достатньо потужні і пропонують вживати розгалужені фільтри для підбору книжок, розділи огляду новинок в них не відзначаються такими ж можливостями. Зокрема, в магазині Клуб сімейного дозвілля можна сортувати новинки за датою появи, популярністю та ціною (рис. 4.4).

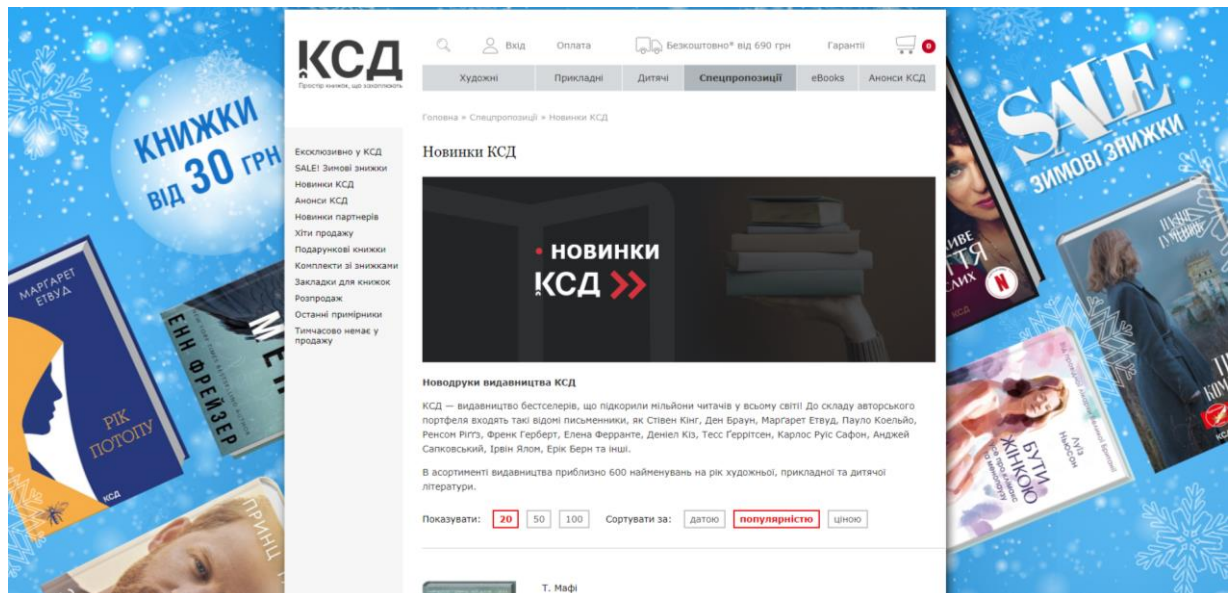


Рисунок 4.4 – Розділ новинок в Інтернет-магазині Клуб сімейного дозвілля

В Інтернет-магазинах Book24.ua та Yakaboo є можливість сортування новинки за популярністю та ціною (рис. 4.5, 4.6), а в магазині Вуква.ua (рис. 4.7) взагалі немає можливості сортувати новинки, вони усі разом подаються без врахування жодних атрибутів чи вимог читачів.

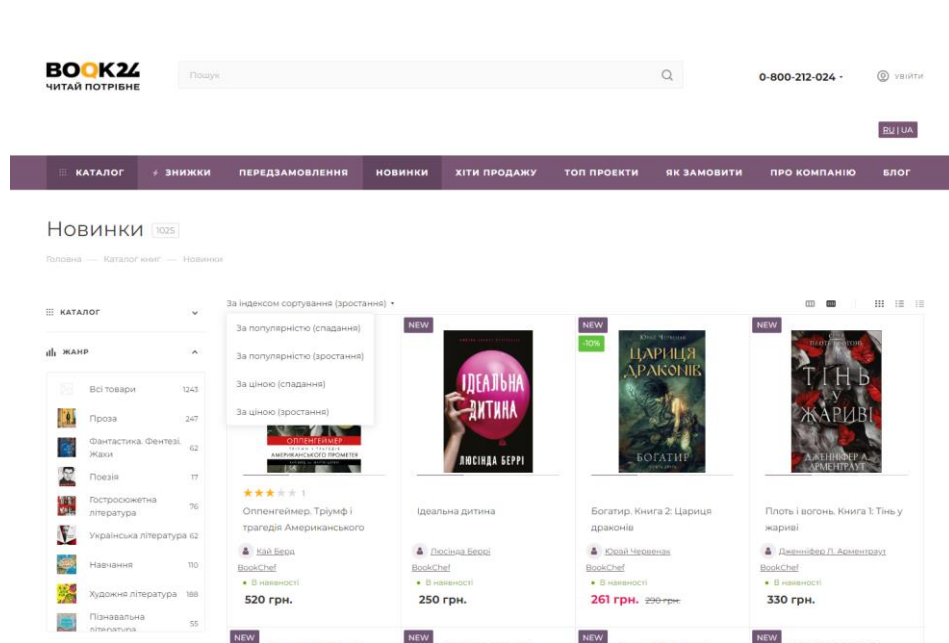


Рисунок 4.5 – Розділ новинок в Інтернет-магазині Book24.ua

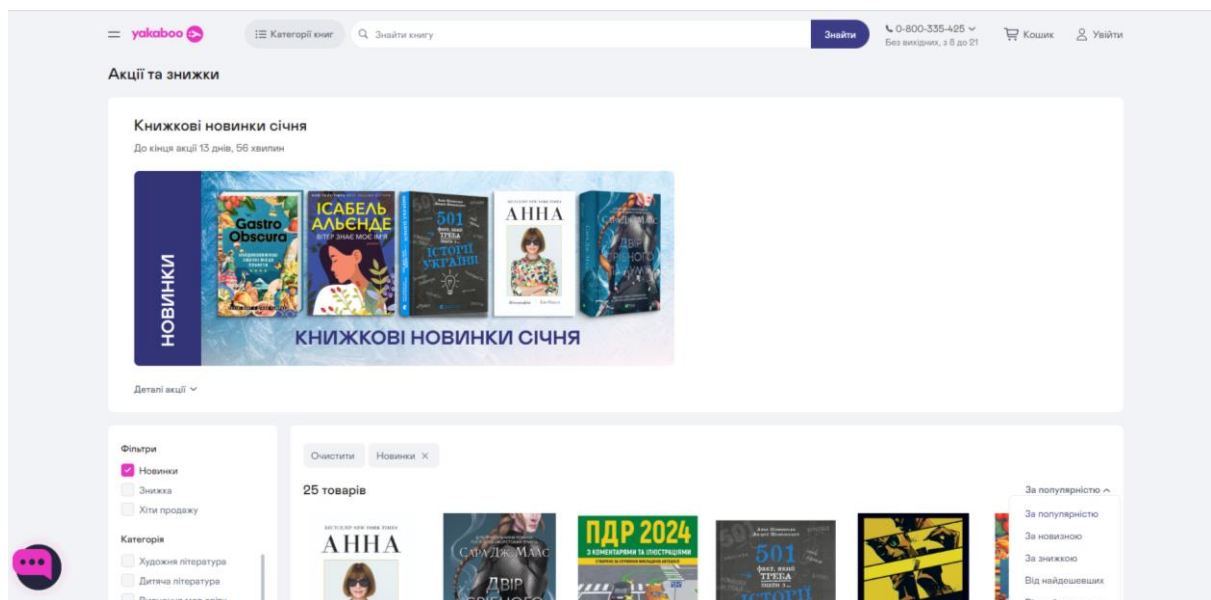


Рисунок 4.6 – Розділ новинок в Інтернет-магазині Yakaboo

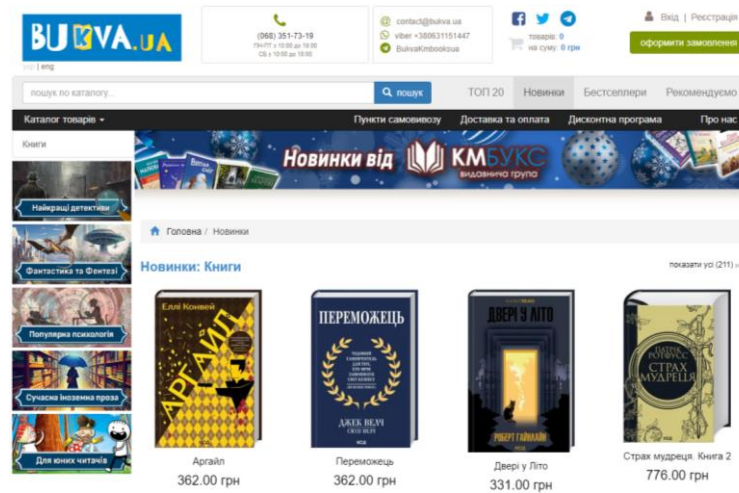


Рисунок 4.7 – Розділ новинок в Інтернет-магазині Bukva.ua

Щоб співставити ці варіанти з розробленим експериментальним зразком агрегації книжкових новин, довелося запросити двох експертів. Результати оцінювання варіантів методом ранжування наведено в табл. 4.1.

Таблиця 4.1 – Результати експертного оцінювання варіантів рекомендацій книжкових новинок методом ранжування

Проект	Ранги		Середній ранг
	Перший експерт	Другий експерт	
Клуб сімейного дозвілля КСД	2	1	1,5
Book24.ua	3	4	3,5
Yakaboo	4	3	3,5
Bukva.ua	5	5	5
Розроблений експериментальний зразок	1	2	1,5

Запропонована методика розділяє першість з варіантом рекомендацій новинок в працюючому магазині Клуб сімейного дозвілля КСД. Для оцінки узгодженості думок двох експертів вживається коефіцієнт рангової кореляції Спірмена:

$$\rho = 1 - \frac{6 \sum_{i=1}^m (x_{ij} - x_{ik})^2}{m(m^2 - 1)} = 1 - \frac{6 \sum_{i=1}^m d_i^2}{m(m^2 - 1)}, \quad (4.1)$$

де x_{ij} – оцінка і-го варіанту j-м експертом;

m – кількість варіантів;

d_i – різниця між оцінками експертів для і-го варіанту.

Результати розрахунків наведені в табл. 4.2.

Таблиця 4.2 – Результати розрахунків різниць оцінок та їх квадратів

Проект	Ранги		d_i	d_i^2
	Перший експерт	Другий експерт		
Клуб сімейного дозвілля КСД	2	1	1	1
Book24.ua	3	4	1	1
Yakaboo	4	3	1	1
Bukva.ua	5	5	0	0
Розроблений експериментальний зразок	1	2	1	1
			$\sum_{i=1}^m d_i^2 =$	4

$$\rho = 1 - \frac{6 \cdot 4}{5(4^2 - 1)} = 1 - \frac{24}{75} = 0.68.$$

Значення коефіцієнта Спірмена лежать в інтервалі $[-1; 1]$, і чим вони ближчі до 1, тим вища узгодженість думок експертів. У даному випадку можна сказати, що узгодженість досить висока, результати співставлення надійні.

5 ЕКОНОМІЧНА ЧАСТИНА

5.1 Характеристика науково-дослідного рішення

Метою даного розділу є економічне обґрунтування витрат на проведення науково-дослідної роботи (НДР), в межах якої передбачається дослідження процесу прийняття рішень для агрегації новин про друковані видання за допомогою рекомендаційної системи. Під час такого обґрунтування буде здійснено: розрахунок трудовитрат та заробітної плати працівникам, розрахунок одноразових витрат і прибутку, оцінку результатів НДР.

Реалізація НДР передбачає такі етапи:

- аналіз підходів до розробки рекомендаційних систем;
- обґрунтування методу експерименту, що буде використаний в дослідженні;
- розробка рекомендаційної системи під час вибору книжкової новинки;
- оцінка доцільності використання розробленої системи;
- розробка рекомендацій для підтримки прийняття рішення під час агрегації новин про книжкові видання.

5.2 Етапи виконання НДР, їх трудомісткість та заробітна плата

Під час виконання науково-дослідної роботи було проведено огляд підходів до створення рекомендаційних систем, досліджено процес прийняття рішення про покупку. Проведено експеримент з розробки автоматичної системи видачі рекомендації про книжкові новинки.

Умовно НДР можна розділити на три етапи: підготовчий, основний і заключний.

На стадії виконання підготовчого етапу були виконані підбір і аналіз інформації для проведення відповідних до постановки завдання робіт. Проведено пошук інформації в мережі Internet та у фаховій літературі.

На етапі виконання основної частини НДР були виконані такі роботи:

- дослідження процесу прийняття рішення про покупку користувачем;
- аналіз існуючих підходів до створення рекомендаційних систем;
- розробка експериментального зразку автоматичної системи подання рекомендації з придбання книжкової новинки;
- оцінка ефективності прийнятих рішень;
- обробка та аналіз результатів експерименту.

У заключній частині здійснюється оцінка ефективності виконання НДР, складання звіту з НДР, захист звіту.

Найбільш складною та відповідальною частиною при плануванні НДР є розрахунок трудомісткості робіт, тому що трудові витрати часто становлять основну частину вартості науково-дослідних робіт і безпосередньо впливають на строки розробки.

Дану роботу виконував фахівець з обробки даних. Середня заробітна плата фахівця становить 60 000,00 грн.

Проведемо розрахунок трудовитрат і заробітної плати виконавця робіт.

Середньоденна заробітна плата виконавця робіт ($Z_{ср.дн.}$) розраховується:

$$Z_{ср.дн.} = \frac{Z_{ср.міс.}}{n}, \quad (5.1)$$

де $Z_{ср.міс.}$ – середньомісячна зарплата виконавця роботи;

n – число робочих днів у місяці, ($n = 22$).

Середньоденна заробітна плата фахівця з обробки даних складає:

$$Z_{ср.дн.} = \frac{60000,00}{22} = 2727,27 \text{ грн.}$$

Етапи виконання НДР, перелік і зміст робіт, трудомісткість їх виконання, заробітна плата виконавця робіт представлені в табл. 5.1.

Таблиця 5.1 – Розрахунок трудовитрат і заробітної плати виконавців робіт

Перелік робіт	Кількість виконавців	Посада виконавця	Трудомісткість робіт, люд.-днів	Середньоденна заробітна плата, грн	Сума заробітної плати, грн
1	2	3	4	5	6
1. Підготовчий етап					
1.1. Розробка та затвердження ТЗ	1	Фахівець з обробки даних	2	2727,27	5454,54
1.2 Підготовка довідкових матеріалів та даних для виконання НДР	1	Фахівець з обробки даних	2	2727,27	5454,54
2. Основний етап					
2.1 Постановка задачі	1	Фахівець з обробки даних	1	2727,27	2727,27
2.2 Дослідження процесу прийняття рішення про покупку користувачем	1	Фахівець з обробки даних	3	2727,27	8181,81
2.3 Аналіз існуючих підходів до створення рекомендаційних систем	1	Фахівець з обробки даних	5	2727,27	13636,35
2.4 Розробка автоматичної системи подання рекомендації з придбання книжкової новинки	1	Фахівець з обробки даних	5	2727,27	13636,35
2.5 Проведення оцінки ефективності прийнятих рішень		Фахівець з обробки даних	0,5	2727,27	1363,64
2.6 Обробка результатів експерименту	2	Фахівець з обробки даних	0,5	2727,27	1363,64
3. Заключний етап					
3.1 Формування висновків та пропозицій з використання розроблених рекомендацій	1	Фахівець з обробки даних	1	2727,27	2727,27
3.2 Технічне оформлення звіту виконання НДР	1	Фахівець з обробки даних	2	2727,27	5454,54
Усього			22		59999,94

5.3 Розрахунок одноразових витрат на розробку НДР

Калькуляція собівартості розраховується відповідно до існуючих нормативних актів України. До складу калькуляції входять такі статті витрат:

- матеріальні витрати;
- витрати на оплату праці;
- єдиний соціальний внесок;
- амортизація основних засобів (вартість машинного часу);
- витрати на спожиту електроенергію;
- інші витрати.

До інших витрат відносяться адміністративні витрати (водопостачання, водовідведення, опалення, освітлення) та вартість послуг зв'язку.

Під час дослідження малоцінні предмети, що швидко зношуються та не належать до матеріальних витрат, не вживалися, література, необхідна для роботи знайдена в Інтернеті у відкритому доступі безкоштовно.

Витрати на оплату праці розраховуються, виходячи з необхідного для виконання робіт складу й кількості працівників, а також із середньомісячної заробітної плати. Відповідно до проведених розрахунків витрати на оплату праці виконавців роботи дорівнюють 59999,94 грн.

Єдиний внесок на загальнодержавне соціальне страхування (ЄСВ) – консолідований страховий внесок, збір якого здійснюється в систему загальнообов'язкового державного соціального страхування в обов'язковому порядку і на регулярній основі з метою забезпечення захисту у випадках, передбачених законодавством, прав застрахованих осіб і членів їх сімей на отримання страхових виплат (послуг) за діючими видами загальнообов'язкового державного соціального страхування.

Ставка єдиного соціального внеску складає 22 % від витрат на оплату праці, тобто розмір ЄСВ дорівнює 13199,99 грн.

Під час виконання НДР застосовувалось наступне обладнання: комп'ютер вартістю 20000,00 грн.

Вищенаведене устаткування є власністю організації виконавця, тому доцільно розрахувати суму амортизаційних відрахувань на період виконання НДР. Амортизація основних засобів розраховується за формулою:

$$AB = \sum_{k=1}^L \frac{BO_k}{TE_k} \times T, \quad (5.2)$$

$$AB = \frac{20000,00 \times 22}{1460} = 301,37 \text{ грн.}$$

де AB – сума амортизаційних відрахувань, нарахованих під час проведення науково-дослідної роботи;

BO_k – вартість основних засобів k -го виду;

TE_k – термін експлуатації основних засобів k -го виду, днів;

T – термін науково-дослідницької роботи, днів;

L – кількість видів обладнання.

Витрати на використану обладнанням електроенергію (B_e):

$$B_e = M \cdot t \cdot T_{kBm}, \quad (5.3)$$

де M – потужність устаткування, тобто кількість енергії, споживаної за одиницю часу (кВт/година);

t – кількість годин використання устаткування за період проведення науково-дослідницької роботи;

T_{kBm} – тариф, тобто вартість використання 1 кВт електроенергії.

Споживна потужність комп'ютера складає 0,5 кВт за годину. Тариф складає 2,64 грн./кВт. Підставивши значення у формулу (5.3), визначимо величину витрат (B_e) на спожиту електроенергію:

$$B_e = 0,5 \cdot 176 \cdot 2,64 = 232,32 \text{ грн.}$$

До інших статей витрат відносяться адміністративні витрати: (водопостачання, водовідведення, освітлення, опалення), які прийнято у розмірі 20% від витрат на оплату праці, та вартість оплати послуг зв'язку.

Вартість оплати послуг зв'язку Інтернет становитиме 250,00 грн. за 22 дні виконання НДР.

Результати розрахунку кошторису витрат, тобто одноразових витрат, на виконання НДР, наведені в табл. 5.2.

Таблиця 5.2 – Кошторис витрат на розробку НДР

№ з/п	Стаття витрат	Сума, грн.
1	Заробітна плата	59999,94
2	Єдиний соціальний внесок (22,0 % від п.1)	13199,99
3	Амортизація основних засобів	301,37
4	Витрати на спожиту електроенергію	232,32
5	Інші витрати, у тому числі:	–
5.1	Адміністративні витрати (20,0 % від п.1)	11999,99
5.2	Вартість послуг зв'язку	250,00
6	Усього витрати	85983,61

Таким чином, кошторис витрат на виконання даної НДР відбиває сумарні витрати за статтями і складає 85983,61 грн.

5.4 Оцінка результатів науково-дослідної роботи

Результат – це наслідок послідовності дій, виконаних під час НДР, виражений якісно або кількісно. В загальному випадку оцінка результатів НДР – це визначення ефективності отриманих рішень порівняно з сучасним науково-технічним рівнем.

Відповідно до теми даного дослідження у якості результату впровадження НДР визначено вживання автоматичної рекомендаційної

систему з вибору книжкових новинок, що можна використати в роботі інтернет-магазину з продажу книжок.

Результат від впровадження НДР визначається за формулою:

$$\Delta P_j = |X_{бj} - X_{нj}|, \quad (5.4)$$

де ΔP_j – покращення j -ої характеристики досліджуваного процесу за рахунок впровадження результатів НДР ($j = 1, m$);

m – кількість досліджуваних характеристик;

$X_{бj}$ – базове значення j -ої характеристики;

$X_{нj}$ – нове значення j -ої характеристики після впровадження НДР.

У якості досліджуваної характеристики обрано ранг системи агрегації книжкових новин у порівнянні з найгіршим з обраних варіантів в результаті оцінки експертів. Підставивши відповідні значення до формули (5.4), визначимо результати від впровадження НДР у чисельному вигляді:

$$\Delta P_1 = |5 - 1,5| = 3,5,$$

$$\Delta P_2 = |3,5 - 1,5| = 2.$$

Отриманий результат свідчить про те, що запропоновану систему агрегації книжкових новин можна ефективно використовувати під час розробки інтернет-магазинів з продажу книжок або самостійних додатків для любителів читання.

Далі проведено оцінку економічної ефективності отриманого результату виконаної науково-дослідної роботи.

5.5 Визначення економічної ефективності результатів НДР

Для визначення економічної ефективності результатів НДР необхідно порівняти витрати на розробку НДР з отриманими результатами.

Основним показником економічної ефективності науково-дослідної роботи є коефіцієнт «ефект-витрати», який розраховується за формулою:

$$K_{ев} = \frac{\Delta P_j}{B_p}, \quad (5.5)$$

$$K_{ев} = \frac{\Delta P_j}{B_p} = \frac{2}{85983,61} \cdot 100\% = 0.0023\% .$$

де B_p – витрати (кошторисна вартість) на виконання НДР, грн;

$K_{ев}$ – коефіцієнт «ефект-витрати», який відбиває, наскільки кожна гривня витрат НДР змінює j -ту характеристику досліджуваного процесу.

Враховуючи те, що покращення, яке забезпечує запропонована система агрегації книжкових новинок у порівнянні з іншими альтернативами, майже однакове, розрахуємо коефіцієнт «ефект-витрати» для даного стандарту по відношенню до наступної за ним альтернативи – до варіантів рекомендаційних фільтрів в магазинах Book24.ua та Yakaboo: $\Delta P_2 = |3,5 - 1,5| = 2$.

У результаті проведених досліджень, можна зробити висновок про те, що кожна гривня витрат на розробку НДР забезпечує підвищення оцінки експертів, а тому й підвищену якість розроблених інтернет-магазинів або додатків для читачів на 0,0023 %. Дана науково-дослідна робота має позитивний показник економічної ефективності. Роботу у цілому можна вважати ефективною або такою, що має науковий та технічний рівень.

ВИСНОВКИ

Незважаючи на стрімкий розвиток мультимедійних технологій залишається велика кількість людей, що полюбляють читати книжки, в тому числі і паперові. Але кількість письменників та нових книжок щороку зростає, тому для читачів незамінними будуть сервіси, що допоможуть зорієнтуватися в цьому потоці інформації.

У ході виконання роботи було досліджено теоретичні моделі поведінки споживача та прийняття рішення про покупку, зокрема, про покупку книжок.

Проведено аналіз підходів до реалізації рекомендаційних систем, які дозволяють приймати обґрунтовані рішення під час вибору книжкових новинок. Досліджено зміст проблеми «холодного старту», запропоновано підхід до подолання цієї перешкоди.

Було розроблено план проведення експерименту, створено експериментальний зразок системи агрегації книжкових новинок, проведено співставлення зразку з існуючими системами в українських книжкових інтернет-магазинах. Проведено оцінку зразка експертним методом. Визначено, що запропонований підхід до агрегації новинок про книжки може успішно конкурувати із вже існуючими рекомендаційними системами.

Доцільність дослідження було економічно обґрунтовано. Проведено розрахунок трудомісткості робіт, калькуляцію собівартості проведення дослідження. Загальні витрати на виконання роботи становлять 85 983,61 грн.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Гуцол Г. Прикладні наукові дослідження Книжкової палати України у 2022 році: зміст, результати, впровадження // Вісник Книжкової палати. 2023. №3. С. 3-11.
2. Cremonesi P., Donatucci A., Garzotto F., Turrin R. Decision-Making in Recommender Systems: The Role of User's Goals and Bounded Resources // Proc. of the RecSys 2012, Workshop on Human Decision Making in Recommender Systems, September 9-13, 2012, Dublin, Ireland. P. 1-7.
3. How to Build a Book Recommendation System. Raghav Agrawal. URL: <https://www.analyticsvidhya.com/blog/2021/06/build-book-recommendation-system-unsupervised-learning-project/> (дата звернення: 10.10.2024).
4. Books Recommendation System. URL: <https://github.com/PritamAich/Books-Recommendation-System> (дата звернення: 10.10.2024).
5. How to Build a Real-Time Book Recommendation System on Amazon Books Dataset? URL: <https://memgraph.com/blog/how-to-build-a-real-time-book-recommendation-system-on-amazon-books-dataset> (дата звернення: 10.10.2024).
6. Awesome Public Datasets. URL: <https://github.com/awesomedata/awesome-public-datasets> (дата звернення: 10.10.2024).
7. Fast Incremental and Personalized PageRank - chrome-extension. URL: <https://snap.stanford.edu/class/cs224w-readings/bahmani10pagerank.pdf> (дата звернення: 10.10.2024).
8. Blackwell R.D., Miniard P.W., Engel J.F. Consumer behavior. South-Western, Thomson Learning, Mason, Ohio, 9th edition, 2001.
9. Chen L., de Gemmis M., Felfernig A., Lops P., Ricci F., Semeraro G. Human Decision Making and Recommender Systems // ACM Transactions on Interactive Intelligent Systems, Volume 3, Issue 3, Article No.: 17. P. 1-7. <https://doi.org/10.1145/2533670.2533675>.

10. Knotzer N. Product Recommendations in E-commerce Retailing Applications. – Frankfurt am Main, Berlin, Bern, Bruxelles, New York, Oxford, Wien: Peter Lang, 2018. 222 p.
11. Falk K. Practical recommender systems. Manning Publications, 2019. 278 p.
12. Aggarwal C.C. Data Mining. Springer, 2015. 746 p.
13. Node.js. URL: <https://nodejs.org/en> (дата звернення: 10.10.2024).
14. КСД Простір книжок, що захоплюють. URL: <https://bookclub.ua/catalog/special/novinki-ksd/> (дата звернення: 10.10.2024).
15. Book24 Читай потрібне. URL: <https://book24.ua/ua/catalog/newproduct/> (дата звернення: 10.10.2024).
16. Yakaboo. URL: https://www.yakaboo.ua/ua/promotion/knizhkovi-novinki-sichnja-2024.html?published_at=latest (дата звернення: 10.10.2024).
17. Bukva.ua. URL: <https://bukva.ua/ua/noveltyies> (дата звернення: 10.10.2024).