

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерних наук _____
(повна назва)

Кафедра _____ Інформаційних управляючих систем _____
(повна назва)

АТЕСТАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти _____ другий (магістерський) _____
(рівень вищої освіти)
(позначення документа)

Дослідження методів _____
вирішення проблеми холодного старту _____
в проектах побудови рекомендаційних систем _____
(тема)

Виконав: студент 2 курсу, групи УПГІТм-18-1
Орехова І. В. _____
(прізвище, ініціали)

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

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

Освітня програма _____
Управління проектами в галузі _____
(повна назва освітньої програми)

Керівник д.т.н, проф. каф. ІУС Чалий С.Ф. _____
(посада, прізвище, ініціали)

Допускається до захисту

Зав. каф. ІУС

_____ Петров К.Е. _____
(підпис) (прізвище, ініціали)

2019 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук

Кафедра Інформаційних управляючих систем

Рівень вищої освіти другий (магістерський)

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

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

Освітня програма Управління проектами в галузі інформаційних технологій
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

“___” _____ 2019 р.

ЗАВДАННЯ
НА АТЕСТАЦІЙНУ РОБОТУ

студентові Ореховій Ізабеллі Віталіївни
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження методів вирішення проблеми холодного старту в проектах побудови рекомендаційних систем

затверджена наказом по університету від 31.10.2019р. № 1588Ст

2. Термін подання студентом роботи до екзаменаційної комісії 13.12. 2019 р.

3. Вихідні дані до роботи Існуючі методи вирішення проблем холодного старту, методи побудови рекомендаційних систем, алгоритми фільтрації даних, кейс рекомендаційної системи, модель системи, яка дозволяє отримувати розширені відповіді на запити, використовуючи дані із зовнішніх джерел

4. Перелік питань, що потрібно опрацювати в роботі Аналіз проблеми холодного старту в проектах побудови рекомендаційних систем, дослідження методів вирішення проблеми холодного старту, дослідження особливостей реалізації усунення проблеми холодного старту під час побудови рекомендаційної системи, опис впровадження рекомендаційної системи до ІТ-продукту

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на атестаційну роботу магістра	04.11.2019	
2	Попереднє дослідження галузі завдання	05.11.2019 08.11.2019	-
3	Аналіз проблеми холодного старту в проектах побудови рекомендаційних систем	09.11.2019 14.11.2019	-
4	Дослідження методів вирішення проблеми холодного старту	15.11.2019 19.11.2019	—
5	Дослідження особливостей реалізації усунення проблеми холодного старту під час побудови рекомендаційної системи	20.11.2019 03.12.2019	—
6	Опис впровадження рекомендаційної системи до ІТ-продукту	04.12.2019 09.12.2019	—
7	Оформлення пояснювальної записки та графічного матеріалу	10.12.2019 12.12.2019	—
8	Подання студентом роботи для перевірки на плагіат	13.12.2019	

Дата видачі завдання ____ 2019 р.

Студент _____
(підпис)

Керівник роботи _____ д.т.н, проф. каф. ІУС Чалий С.Ф.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Реферат атестаційної роботи магістра: 120 с., 30 рис., 12 табл., 33 джерел.

РЕКОМЕНДАЦІЙНА СИСТЕМА, МЕТОД КОЛЛАБОРАТИВНОЇ ФІЛЬТРАЦІЇ, МЕТОД КРИТИЧНОГО ШЛЯХУ, ДІАГРАМА ГАНТА, ORGANIZATION BREAKDOWN STRUCTURE, WORK BREAKDOWN STRUCTURE.

Об'єкт дослідження – процеси формування рекомендацій в умовах холодного старту.

Предмет дослідження – методи та моделі, що використовуються для вирішення проблеми холодного старту у проектах побудови рекомендаційної системи.

Мета роботи – дослідження методів вирішення проблеми холодного старту в проектах побудови рекомендаційних систем.

Методи дослідження – дослідження існуючих методів вирішення проблеми холодного старту, їх формальних описів, дослідження особливостей реалізації усунення проблеми холодного старту під час побудови рекомендаційної системи, дослідження архітектури рекомендаційних систем.

Визначено підхід до вирішення проблеми холодного старту в проектах побудови рекомендаційних систем. Запропоновано метод та модель вирішення проблеми холодного старту в умовах відсутніх або розріджених даних.

Виконано впровадження рекомендаційної системи до ІТ-продукту. Виконано аналіз критичного шляху проекту [1,2].

ABSTRACT

Abstract on master's thesis: 120 p., 30 pic., 12 tables, 33 sources

RECOMMENDATION SYSTEM, COLLABORATIVE FILTRATION METHOD, CRITICAL WAY METHOD, GUNT DIAGRAM, ORGANIZATION BREAKDOWN STRUCTURE, WORK BREAKDOWN STRUCTURE.

The object of study - the design process of the recommendation system.

Subject of research - methods and models used to solve the problem of cold start in projects of construction of the recommendation system.

The purpose of the work is to investigate methods of solving the problem of cold start in projects of construction of recommendation systems.

Research methods - research of existing methods of solving cold start problem, their formal descriptions, research of methods of evaluation of recommendation systems, research of peculiarities of realization of elimination of problem of cold start during construction of recommendation system, study of architecture of recommendation systems.

The approach to solving the problem of cold start in the projects of construction of recommendation systems is determined. The method of solution of cold start problem in the conditions of missing or sparse data is offered.

Implementation of a recommendation system for an IT product. The critical path of the project was analyzed [1,2].

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ	8
ВСТУП	9
1 АНАЛІЗ ПРОБЛЕМИ ХОЛОДНОГО СТАРТУ В ПРОЕКТАХ ПОБУДОВИ РЕКОМЕНДАЦІЙНИХ СИСТЕМ.....	11
1.1 Дослідження проблеми рекомендаційних систем	11
1.2 Методи побудови рекомендаційних систем.....	16
1.3 Дослідження проблеми холодного старту.....	22
1.4 Постановка задач магістерської атестаційної роботи	26
1.5 Висновки до розділу	27
2 ДОСЛІДЖЕННЯ МЕТОДІВ ВИРІШЕННЯ ПРОБЛЕМИ ХОЛОДНОГО СТАРТУ	29
2.1 Дослідження особливостей побудови рекомендаційної системи в умовах холодного старту.....	29
2.2 Дослідження методу побудови рекомендацій в умовах холодного старту з використанням темпоральних обмежень за вхідними даними	34
2.3 Удосконалення методу побудови рекомендацій в умовах холодного старту з використанням темпоральних обмежень за вхідними даними	38
2.4 Висновки до розділу	41
3 ОПИС ВПРОВАДЖЕННЯ РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ ДО ІТ-ПРОДУКТУ	43
3.1 Опис рішень з ініціації ІТ-проекту	43
3.2 Опис результатів планування ІТ-проекту	47
3.3 Аналіз критичного шляху проекту.....	58

3.4 Висновки до розділу	61
4 ДОСЛІДЖЕННЯ ОСОБЛИВОСТЕЙ РЕАЛІЗАЦІЇ УСУНЕННЯ ПРОБЛЕМИ ХОЛОДНОГО СТАРТУ ПІД ЧАС ПОБУДОВИ РЕКОМЕНДАЦІЙНИХ СИСТЕМ.....	62
4.1 Опис особливостей реалізації рекомендаційної системи як самостійного ІТ–продукту	62
4.2 Опис послідовностей операцій під час роботи рекомендаційної системи	70
4.3 Опис практичної апробації методу та тестування результатів	79
4.4 Висновки до розділу	92
ДОДАТОК А.....	102

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

БД – база даних;

ІС – інформаційна система

МКФ – метод колаборативної фільтрації;

МКШ – метод критичного шляху;

РС – рекомендаційна система;

СКБД – система керування базами даних;

API –Application Programming Interface;

CBF – content-based filtering;

CF – collaborative filtering;

CSP – Cold-start Problem;

CoCoS – Continuous Cold-start Problem;

ICoCoS – Item Continuous Cold-Start;

GPS –Global Positioning System;

OBS – Organization Breakdown Structure;

RBS – Resource Breakdown Structure;

SVD – Singular value decomposition;

UCoCoS – User Continuous Cold-Start;

WBS – Work Breakdown Structure.

ВСТУП

У сучасних умовах разом з розвитком Інтернет-сервісів велику увагу приділяють персоналізації. Інтернет-магазини, пошукові системи, соціальні мережі, форуми, ресурси новин намагаються пристосувати зовнішній вигляд і зміст своїх сторінок до вимог конкретного користувача.

За результатами досліджень компанії Evergage у 2015 році персоналізацію у реальному часі використовували 44% Web-сайтів, 17 % мобільних сайтів, 13% Web-додатків і 9 % мобільних додатків [3]. На сьогоднішній день ці відсотки значно більші. Таким чином можна стверджувати, що збільшення користувацького досвіду та залучення користувачів до інтерактивного спілкування з сервісами є важливими результатами застосування персоналізації.

Наявність персоналізованого контенту дозволяє суттєво підвищити зокрема такий показник як конверсія. Для якісної персоналізації сайтів, що орієнтовані роботу з великою кількістю користувачів, зазвичай використовують комплексний підхід, який поєднує маркетингові дослідження та аналіз поведінки користувачів сайту. Інформацію про маркетингові дослідження можна отримати за допомогою систем Web-аналітики, наприклад: Adobe Digital Marketing Suite, Google Analytics або Siteapps.com.

Вихідні дані для аналізу поведінки користувачів можуть бути отримані явним та неявним способом. Явним способом отримують результати опитувань, голосувань, тощо. Проте більша частина інформації про активність користувачів отримана неявним способом. У таких випадках предметом дослідження є переходи по сторінках сайтів, покупки товарів і

послуг, час перебування на окремих сторінках та ін. При цьому йдеться про великі масиви неоднорідних даних.

У сфері електронної комерції основним інструментом персоналізації є рекомендаційні системи, що забезпечують обробку даних активності користувача та формування рекомендацій на товари, які можуть зацікавити користувача.

Під час роботи рекомендаційної системи можуть виникати такі проблеми як, наприклад, холодний старт або інші.

Основне завдання, яке автор роботи ставив перед собою, є дослідження методів побудови рекомендаційних систем в умовах холодного старту, яке дозволить повною мірою використовувати вихідні дані про відвідувачів Інтернет-магазину з метою формування рекомендацій, що максимально відображають очікування.

Пояснювальна записка виконана відповідно до методичних вказівок.

1 АНАЛІЗ ПРОБЛЕМИ ХОЛОДНОГО СТАРТУ В ПРОЕКТАХ ПОБУДОВИ РЕКОМЕНДАЦІЙНИХ СИСТЕМ

1.1 Дослідження проблеми рекомендаційних систем

У повсякденному житті люди покладаються на рекомендації інших людей, поданих у вигляді промовлених слів, рекомендаційних листів, повідомлень у новинах, результатів опитувань, путівників. Рекомендаційні системи допомагають і розширюють цей природний соціальний процес, і вже давно стали популярним рішенням проблеми інформаційного завантаження.

Таким чином можна сказати, що рекомендаційні системи – це програми та сервіси, основне завдання яких передбачити, які об'єкти (товари або послуги) будуть цікаві користувачеві, за наявності інформації про його профілі або інші дані.



Рис. 1.1 – Приклад роботи рекомендаційної системи сайту elmir.ua

Зазвичай РС формують рекомендації незалежно для кожного конкретного користувача на основі його минулих покупок і пошуків, а також на основі поведінки інших користувачів. Відповідно до цього завдання рекомендаційної системи полягає у інформуванні користувача про товар, який може бути цікавим у даний момент часу.

У результаті такої взаємодії користувач отримує інформацію, а сервіс заробляє на наданні послуг. При цьому послуги – це не обов'язково прямі продажі пропонованого товару, а, наприклад, комісійні або збільшення лояльності користувачів, яка потім впливає на рейтинги різного типу.

З огляду на це можна сказати, що РС змінили способи взаємодії Web-сайтів зі своїми користувачами, тобто замість надання статистичної інформації збільшили ступінь інтерактивності і таким чином розширили можливості користувача.

Рекомендаційні системи можна зустріти на багатьох Web-сайтах різного призначення. Наприклад, сайт LinkedIn бізнес-орієнтованої соціальної мережі пропонує користувачеві рекомендації щодо людей, робочих місць, груп, компаній, якими він міг би зацікавитися. Спеціалізована система колаборативної фільтрації LinkedIn заснована на технології Apache Hadoop.

Amazon – відомий сайт електронної комерції – використовує рекомендації на основі контенту. В основі формування рекомендацій Amazon лежить підхід, коли на основі обраного користувачем товару рекомендуються інші товари, придбані іншими користувачами (за допомогою матриці покупки наступного товару на основі його схожості з попередньою покупкою). Компанія Amazon запатентувала цей підхід під назвою item-to-item collaborative filtering (колаборативна фільтрація від елемента до елемента) [4].

Hulu – сайт потокового відео – використовує у режимі оффлайн механізм Hadoop для масштабованої обробки великих обсягів даних при виконанні колаборативної фільтрації на основі елементів” (item-based collaborative filtering).

Компанія Netflix, яка є постачальником відеоконтенту на умовах оренди і у вигляді потокового сервісу, у 2009 році вдосконалила свою

рекомендаційну систему під назвою Cinematch за рахунок 107 рекомендаційних алгоритмів, що формують єдиний прогноз.

Крім того, рекомендаційні механізми використовуються на таких сайтах, як Facebook, Twitter, Google, MySpace, Last.fm, Del.icio.us, Pandora, Goodreads, тому слід зазначити, що використання того чи іншого рекомендаційного механізму поступово стає стандартним елементом будь-якого сучасного Web-сайту.

Існує декілька основних характеристик, за допомогою яких можна описати будь-яку рекомендаційну систему.

1. Предмет рекомендації. Предметом рекомендацій можуть бути товари (Amazon, Ozon), статті (Arxiv.org), новини (unian.net, tsn.ua), зображення (500px), відео (YouTube, Netflix), люди (Linkedin, LonelyPlanet), музика (Last.fm, Pandora) та багато іншого.

2. Мета рекомендації. Наприклад: покупки, інформування, навчання, рекомендація закладів різного типу, тощо.

3. Контекст рекомендації. Відповідає поточним діям користувача: перегляд товарів, прослуховування музики, спілкується з людьми, пошук інформації, тощо.

4. Джерело рекомендації.

- окремі користувачі (форми та чати);
- аудиторія (наприклад, рейтинги закладів громадського харчування);
- схожі за інтересами користувачі (форми та чати);
- експертне співтовариство (складний товар, наприклад, вино).

5. Ступінь персоналізації.

- неперсональні рекомендації – рекомендації загального характеру (не враховують особисті переваги);
- рекомендації за даними з поточної сесії користувача (після перегляду кількох товарів пропонують схожі);

– персональні рекомендації – рекомендації відповідно до всієї доступної інформації про клієнта (разом з історією покупок).

6. Прозорість. Рекомендації, отримані незрозумілим шляхом, не викликають довіри. Рекомендаційна система повинна бути здатна відсіювати “накрутки” продавців, сфабриковані відгуки, “проплачені” товари та ін. Проте бувають випадки, коли рекомендації не відповідають дійсності з об’єктивних причин (новий блокбастер насамперед дивляться фанати, тому у перші місяці рейтинг може бути завищений).

7. Формат рекомендації. Залежить від дизайну: банер, результат запиту, спливаюче вікно, відсортований список, стрічка, тощо.

8. Алгоритми. Існує безліч алгоритмів, найбільш поширеними є Summary-based (неперсональні), Content-based (моделі засновані на описі товару), Collaborative Filtering (коллаборативна фільтрація), Matrix Factorization (методи засновані на розкладанні матриць).

Більшість сучасних рекомендаційних систем побудована за одним принципом – на основі матриці переваг (табл. 1.1). Одна вісь матриці для клієнтів сервісу (Users), інша – для об’єктів рекомендації (Items). На перетині деяких пар (user, item) клітини матриці заповнені оцінками (Ratings) показник зацікавленості користувача у даному товарі (шкала довільна, наприклад, від 1 до 5).

Таблиця 1.1 – Матриця переваг

	Товар 1	Товар 2	Товар 3	Товар 4	Товар 5
Клієнт 1		3		5	
Клієнт 2	1		1	1	
Клієнт 3	2			3	2
Клієнт 4		4			5
Клієнт 5	5		2	3	4

Головне завдання РС полягає в узагальненні інформації про вибір користувачів і передбаченні їх ставлення до інших невідомих товарів (тобто заповнення порожніх клітин).

Рекомендаційні системи можуть бути основою бізнесу (наприклад, у TripAdvisor), а можуть бути додатковим сервісом (будь-який інтернет-магазин), призначення якого покращення Customer Experience та зручності навігації.

Таким чином можна сказати, що рекомендаційні системи змінили засоби взаємодії Web-сайтів з користувачами персоналізуючи онлайн-маркетинг.

За оцінками McKinsey, 35% виручки Amazon або 75% Netflix припадає саме на рекомендовані товари [5].

На основі цієї оцінки можна зробити висновки про ефективність використання РС проте існує декілька проблем, пов'язаних з їх використанням.

Метод колаборативної фільтрації дозволяє спростити збір даних з мережі Інтернет, але велика кількість цих даних може знизити ефективність РС за рахунок зсуву результатів. Це відбувається через те, що одні користувачі на відміну від інших не демонструють типової поведінки або взагалі можуть використати рекомендаційну систему для підвищення рейтингу одного товару порівняно з іншим (збільшення кількості позитивних відгуків). Рекомендаційна система повинна вирішувати проблеми такого характеру.

Ще одна проблема великих РС, пов'язана з обробкою великих обсягів даних у режимі реального часу. У таких випадках традиційні алгоритми не дозволяють отримати якісні результати обробки, а РС потребує використання більш спеціалізованих підходів.

Великою проблемою використання рекомендаційних систем стала також конфіденційність. Як виявилось, РС можуть розпізнавати приховані

закономірності, про існування яких люди навіть не здогадуються. Наприклад, одна велика компанія за допомогою РС змогла обчислити індекс прогнозування вагітності на основі переваг покупців [6].

Проблема холодного старту – це ще одна типова ситуація, коли для коректної роботи РС немає достатньої кількості даних (наприклад, новий товар). У таких випадках рейтинг можуть бути штучно скореговані.

У окремих випадках виникає проблема актуальності рекомендацій (статті, пости, відповіді на форумах). Для того щоб нові записи частіше були використані у рейтингах, використовують коригувальні коефіцієнти (damping factors).

Implicit Ratings (неявні рейтинги) теж є істотною проблемою при використанні РС, особливо коли йдеться про визначення інтересів користувачів за непрямими ознаками (наприклад, банки, автомайстерні або інші сервіси, де неможливо налаштувати систему оцінювання).

1.2 Методи побудови рекомендаційних систем

Рекомендаційні системи можна класифікувати за способами відбору даних необхідних користувачеві. У більшості РС застосовують один з двох базових підходів: колаборативна фільтрація (CF) та тематична або контентна фільтрація (CBF). Також існує гібридна фільтрація, яка поєднує в собі колаборативну та контентну фільтрацію.

Коллаборативна фільтрація формує рекомендації, засновані на моделі попередньої поведінки користувача або групи користувачів з подібною поведінкою. У випадках, коли береться до уваги поведінка групи, РС використовує знання про групу (group knowledge). Для формування груп зазвичай використовують різні методи (data mining) для отримання/аналізу

інформації, наприклад, алгоритми кластеризації. Які, у свою чергу, можуть бути використані для формування профілів користувачів, а потім – груп на основі цих профілів [7].

Основна перевага CF полягає у високому ступеню точності рекомендацій, а недолік – система не може створювати рекомендації на основі інтересів однієї людини.

Колаборативна фільтрація має на увазі той факт, що є дані про користувача та сайт (об'єкт), який він оцінив. Таким чином фільтрація поділяється на два типи: user-based та item-based.

Коли йдеться про user-based, розподіл на групи можна проводити різними методами, наприклад, методом найближчого сусіда можна поєднати у групи людей зі схожими інтересами, а за допомогою діаграми Венна визначити і схожість і відмінності

Головні проблеми CF типу user-based – це відсутність рекомендацій для нових/нетипових користувачів (оскільки немає схожості) та холодний старт (нові об'єкти не рекомендуються).

У випадку з item-based рекомендацій будують за принципом: “знайти об'єкти, схожі на заданий”.

Проблеми CF для item-based – це той самий холодний старт та тривіальність рекомендацій. Проте item-based має переваги перед user-based: оцінка близькості товарів більш точна ніж оцінка близькості користувачів; уподобання користувачів можуть змінюватися з часом, а опис товарів більш стійкий; якщо користувачів багато, задача обчислення найближчого сусіда погано вирішується; в user-based описи користувачів зазвичай дуже розріджені.

Слід зауважити, що термін “колаборативна” (тобто суспільна) “фільтрація” застосовують незалежно від факту, що той, хто рекомендує, не може чітко співпрацювати з тим, хто отримує рекомендації. А самі

рекомендації можуть запропонувати особливо цікаві пункти на додаток до зазначених (які слід відфільтрувати) [8].

Усі методи CF застосовують базу даних уподобань для користувачів, щоб передбачувати додаткові теми або товари, які можуть сподобатися новому користувачеві. У типовому сценарії CF є список з m користувачів $\{u_1, u_2, \dots, u_m\}$ та список з n елементів $\{i_1, i_2, \dots, i_n\}$, і кожен користувач u_i має список елементів I_{u_i} , які він оцінив, або які були визначені через його поведінку.

При цьому рейтинги можуть бути або явними ознаками, або неявними, такими як покупки або кліки. Наприклад, ми можемо перетворити список людей та фільмів, які їм подобаються/не подобаються, на матрицю оцінок елементів користувача.

Таким чином слід зазначити ще один суттєвий недолік CF – це обчислення, які потребують багато ресурсів (для формування рейтингів необхідно тримати у пам'яті оцінки всіх користувачів).

Контент-орієнтовані системи формують рекомендації за принципом, що користувачеві сподобаються елементи які мають схожі властивості (наприклад, фільми одного жанру, книги одного автора) з елементами, якими він користувався раніше, або користувачі з аналогічними характеристиками (наприклад, однакове географічне положення, подібні демографічні дані), мають однаковий смак.

Системи CBF спираються виключно на контент, до якого звертається користувач, а не на поведінку інших користувачів у системі. Наприклад, якщо користувач читає статті про Linux, то РС використає цю інформацію для виявлення подібного контенту та сформує рекомендацію для цього користувача виключно з подібним контентом. Цей контент може бути визначений у ручному режимі або автоматично на основі інших методів подібного. При цьому одна сторона – це користувач, а інша – ранжований

набір подібних елементів, то схожість ігнорується, і ранжована різниця дає можливість сформувати рекомендації.

Останнім часом саме CBF набули великої популярності, тому що користувачі не мають бажання витратити час на “навчання” системи своїм уподобаннями, а власники систем не мають бажання втрачати користувачів відразу після реєстрації.

Гібридна фільтрація, яка у класичному варіанті поєднує колаборативну та контентну фільтрацію, підвищує ефективність і складність РС. Таке поєднання дозволяє підвищити точність рекомендацій та є частковим вирішенням проблеми холодного старту (розрідженість даних).

Гібридний підхід дозволяє присвоїти вагові коефіцієнти результатам відповідно до принципів контентної фільтрації, а потім зміщувати ці коефіцієнти у напрямку до колаборативної фільтрації.

Слід зауважити, що єдиного алгоритму побудови гібридних систем не існує. Різні розробники використовують машинне навчання, методи оптимізації та інші алгоритми. Яскравим прикладом гібридного підходу стала рекомендаційна система Netflix (BellKor), яка є комбінацією з 27 алгоритмів. Ця система має найвищу точність, у Netflix похибка становить приблизно 0,86 бала для 5-бальної шкали оцінок. Єдиний недолік – найвища складність розробки.

Рекомендаційні системи засновані на знаннях (knowledge-based), як і гібридні, не спираються на оцінки інших користувачів, а беруть до уваги тільки профіль користувача і товар. Об'єднує ці два підходи те, що існує наявність знань про якусь предметну область, користувача, товари, чи будь-що, що може допомогти виконати ранжирування.

Часто CBF визначають як окремий випадок knowledge-based, де знаннями є відомості про товар, але CBF настільки поширена, що у більшості літературних джерел визначена як окремий тип.

CBF підхід передбачає наявність додаткової сутності: вимог користувача (user requirements). Наприклад: “дзеркальна камера, з 32 точками авто фокусування, ціна до \$ 2000”. Ці вимоги можуть задаватися шляхом вибору потрібних чекбоксів. Завдання підходу полягає у тому, щоб знайти рекомендовані товари, які відповідають вимогам.

Інші окремі випадки системи заснованої на знаннях наступні.

Demographic-based враховує особисті дані кожного користувача: соціальний статус, місце проживання, зарплатня, тощо.

Utility-based розраховує відносну корисність кожного товару для користувача. Наприклад, якщо користувач купив фотоапарат, швидше за все він за об'єктами для нього прийде.

Critique-based мають додатковий інтерфейс для “критики”. Наприклад, дві осі, за якими користувач буде ставити “хочу дорожче, але з великою кількістю мегапікселів”, а система підлаштується під запит.

Досвід фірми Netflix показує, що у рекомендаційних системах можливе використання різних алгоритмів. Отримані результати будуть відрізнятися та залежати від поставленої задачі та організації даних.

Схожість між двома користувачами (їх атрибутами, наприклад, прочитаними статтями) може бути обчислена за допомогою кореляції Персона. Цей алгоритм вимірює лінійну залежність між двома змінними (користувачами) як функцію їх атрибутів, але не обчислює цю залежність для всієї множини користувачів. Цю множину необхідно попередньо ранжувати за принципом близьких елементів. Кореляція Персона є популярним алгоритмом у сфері CF.

Алгоритми кластеризації – це різновид “спонтанного навчання” (unsupervised learning), який дозволяє виявити закономірність у випадкових даних. Алгоритм заснований на виявленні схожості між елементами (користувачами одного сайту) за допомогою обчислення їх відстані до інших елементів у просторі ознак (feature space) (це може бути кількість прочитаних

статей у наборі сайтів). Кількість незалежних ознак визначає розмір простору ознак. Якщо елементи “близькі”, їх можна об’єднати у один кластер.

Алгоритми факторизації ще називають спектральним розкладанням або високочастотною фільтрацією (видаляють шум і залишають корисний сигнал). Найбільш поширеним з алгоритмів такого типу є метод SVD, який використовують у CF.

Алгоритми факторизації дозволяють узагальнювати модель (опис користувача) і вирішують проблему великих розмірів даних за рахунок опису інтересів користувача не за допомогою вектора товару, а за допомогою вектора вподобань.

Інші алгоритми, які успішно використовують для розробки РС, це Байєсовські мережі довіри (Bayesian Belief Nets), ланцюги Маркова (Markov chains), класифікація за методом Рококо (Rocchio classification), тощо.

Разом зі складними методами формування рекомендацій також використовують простіші алгоритми, такі як статистичні метрики. Їх використовують для виявлення популярних/дешевих/дорогих/... товарів, які повністю або частково відповідають заданим характеристикам. Такі товари пропонують користувачам без урахування їх персональних переваг.

Більш складні алгоритми здатні виявити переваги користувачів за допомогою їх поведінкового профілю, а той, у свою чергу, формується у результаті аналізу його активності під час вибору товарів і послуг. Отримали свій розвиток і підходи, засновані на використанні нечіткої логіки, яка дозволяє враховувати різні типи невизначеностей та розподілити за групами призначені для користувача профілі.

1.3 Дослідження проблеми холодного старту

Проблемою холодного старту називається задача видачі підходящих рекомендацій для нових користувачів.

Цю проблему можна розділити на холодний старт для користувачів (нові товари, нові користувачі) і холодний старт для сайтів (рекомендація нових сайтів).

Холодний старт для користувачів обчислюють на основі демографічних даних, які користувачі вказують під час реєстрації на сайтах. Зазвичай це мінімальний набір типу: стать, дата народження та місце розташування. Проте, за допомогою соціальних мереж можна отримати набагато більше даних, наприклад, рівень освіти, соціальний статус, особисті вподобання та багато іншого [9].

ХИТЫ ГРУППЫ	ТОВАР ГРУППЫ С ЛУЧШИМИ ОЦЕНКАМИ
Материнская плата s-AM4 AMD B450 Asus TUF B450-Pro Gaming ATX  ★★★★★ 2666 грн Материнские платы Отзывы (9)	Материнская плата s-FM2+ AMD A68H ASRock FM2A68M-DG3+ mATX  ★★★★★ 1019 грн Материнские платы Отзывы (16)
Материнская плата s-AM4 AMD B450 MSI B450 Tomahawk Max ATX  ★★★★★ 3189 грн Материнские платы Отзывы (2)	Материнская плата s-AM4 AMD B450 ASRock B450M Pro4 mATX  ★★★★★ 1923 грн Материнские платы Отзывы (8)
Материнская плата s-AM4 AMD B450 MSI B450-A Pro MAX ATX  ★★★★★ 2465 грн Материнские платы Отзывы (4)	Материнская плата s-1151 B360 Asus Prime B360-Plus ATX  ★★★★★ 2532 грн Материнские платы Отзывы (8)
<ul style="list-style-type: none"> • Материнская плата s-AM4 AMD A320 Gigabyte GA-A320M-S2H V2 mATX • Материнская плата s-1151 Z390 Asus TUF Z390-Plus Gaming ATX • Материнская плата s-AM4 AMD B450 Gigabyte B450 AORUS Elite ATX • Материнская плата s-1151 H310 Asus Prime H310M-K R2.0 mATX • Материнская плата s-AM4 AMD A320 ASRock A320M-HDV R4.0 mATX 	<ul style="list-style-type: none"> • Материнская плата s-AM4 AMD B450 Asus TUF B450M-Pro Gaming mATX • Материнская плата s-AM3 AMD 760G GigaByte GA-78LMT-USB3 R2 mATX • Материнская плата + процессор J1900 ASRock Q1900M mATX • Материнская плата s-1151 H110 GigaByte GA-H110-D3A ATX • Материнская плата + процессор J1900 ASRock Q1900B-ITX mini-ITX

Рис. 1.2 – Приклад холодного старту рекомендаційної системи сайту elmir.ua

Розрізняють два основні підходи до застосування у РС демографічної інформації.

1. Експертний підхід. Передбачає, що експерт визначає які дані показувати користувачам різних категорій на холодному старті. За такої умови користувачеві будуть рекомендовані тільки товари/сайти суб'єктивно обрані експертом. При цьому об'єм роботи експерта збільшується за рахунок кількості категорій користувачів.

2. Автоматичний підхід. Демографічні категорії визначаються автоматично шляхом виявлення категорій користувачів зі схожими інтересами. За умови автоматичного підходу рекомендації будують на основі рейтингів користувачів одного віку, статі, місця розташування та ін. На відміну від експертного підходу автоматичний не потребує експертів і дозволяє створювати необмежену кількість категорій.

Проблема холодного старту сайту вирішується за рахунок використання різних методів аналізу тексту та іншого контенту сторінок (зображення, відео, flash, посилання, тощо).

Таким чином можна сказати, що холодний старт – це типова ситуація, коли у системі ще не накопичена достатня кількість даних для коректної роботи РС. Дані будуть достовірними, якщо рейтинг був вирахований за оцінками декількох користувачів, інші відвідувачі сайту це зрозуміють. Зазвичай у таких випадках компанії штучно коректують рейтинги застосовуючи один з двох способів [10].

Перший спосіб отримав назву “згладжене середнє” (Damped Mean). Тобто за умови недостатньої кількості оцінок відображається рейтинг безпечного “середнього” показника, а коли набирається достатня кількість нових оцінок, “усереднене” коригування перестає діяти.

Другий спосіб – це розраховувати для кожного рейтингу інтервали достовірності (confidence Intervals). При цьому, чим більше оцінок, тим менше змінюється середнє значення і, як наслідок, більше вірогідність його

правильності. В якості рейтингу можна показувати нижню межу інтервалу (Low CI Bound). Такий спосіб буде відрізняється заниженням оцінок з нових товарів.

Звичайний спосіб обчислення інтервалів досить складно застосувати до персоніфікованих рекомендацій через обмеженість оцінки (наприклад, шкала від 0 до 1) за симетричною шкалою. Проте існує більш точний спосіб – Wilson Confidence Interval.

На рис.1.2 по горизонталі відкладене середнє значення рейтингу, а по вертикалі – розподіл навколо середнього значення. Розмір вибірки визначається кольором (що більша вибірка, то менше інтервал достовірності).

Для не персоніфікованих рекомендацій CSP теж актуальна. Вирішити проблему можна використовуючи середній рейтинг, прості алгоритми обчислення, або взагалі не використовувати товар, поки не збереться потрібна кількість даних.

Описуючи CSP слід згадати також про проблему постійного холодного старту – CoCoS.

Велика кількість методів, які застосовують для обчислення рейтингів, і, зокрема, метод колаборативної фільтрації у класичному вигляді, не придатний за умови появи нових користувачів (з порожньою історією переваг) або сайтів (не обраних жодним користувачем). Також “відомі” користувачі, які рідко з'являються у системі та змінюють свої інтереси, викликають циклічність CSP. Такі ситуації отримали назву “Проблема постійного холодного старту”.

Існує велика кількість підходів до вирішення CSP: використання базових наборів рекомендацій, гібридних систем (наприклад, поєднання комбінації колаборативної фільтрації та тематичних рекомендацій), вилучення допоміжної інформації. Проте всі ці підходи не діють у випадку з

CoCoS через припущення, що “відомий” користувач залишається таким необмежений проміжок часу.

Як і у випадку з CSP, проблема CoCoS поділяється на UCoCoS та ICoCoS. Перший тип, щойно описаний, виникає за умови появи нових користувачів або таких, які рідко з'являються. Другий – за наявності нових сайтів або таких, які змінювалися з часом (наприклад, сайт компанії Київстар упродовж останніх сіми років повністю оновлювався не менше 4–х разів).

Для вирішення проблеми CoCoS зазвичай використовують комбінацію колаборативної фільтрації і тематичних методів висуваючи припущення, що “схожим” користувачам сподобаються “схожі об'єкти”.

Такий підхід вирішує проблему ICoCoS, проте у випадку з UCoCoS не діє, через неможливість визначити “схожість” користувачів.

У сучасних РС для подолання проблеми CoCoS використовують наступні методи.

1. Метод “контекстного багаторукого бандита” (Tang et al [11]).
2. Алгоритм для вирішення проблеми UCoCoS (Sun et al [12]).
3. Метод на основі тем (Tavakol and Brefeld [13]). Наміри користувача моделюють за допомогою імовірнісного розподілу тем стосовно сайтів або інших об'єктів (наприклад, статей). Під час взаємодії користувача з РС його наміри передбачається через марковський процес. Метод був апробований на даних з систем e-commerce та продемонстрував покращення рекомендацій.

Слід зауважити, що під час проектування та розробки РС необхідно враховувати усі особливості середовища, для якого розробляють систему. Проблема CoCoS і є однією з таких особливостей, тому її не можна зводити до CSP.

1.4 Постановка задач магістерської атестаційної роботи

Для вирішення проблеми холодного старту у проектах побудови рекомендаційних систем необхідно вирішити задачу формування релевантних прогнозів в умовах відсутності вихідних даних. Підходи до рішення зазначеної проблеми були розглянуті у першому розділі. Кожен з них має свої переваги та недоліки. Проте усі розглянуті методи є модифікаціями алгоритмів фільтрації даних, які застосовують рекомендаційні системи для формування рекомендацій.

Крім того, у підрозділі 2.2 буде виконано дослідження методу побудови рекомендацій в умовах холодного старту з використанням темпоральних обмежень за вхідними даними. На основі саме цього методу базується автоматизована побудова рекомендацій в умовах холодного старту з використанням даних із зовнішніх джерел.

На основі цього факту прийнято рішення використати інший підхід до вирішення зазначеної проблеми. В рамках цього підходу пропонується:

- застосувати не математичний, а архітектурний спосіб формування рекомендацій в умовах холодного старту;
- удосконалити метод побудови рекомендацій в умовах холодного старту з використанням темпоральних обмежень за вхідними даними для його подальшого використання у поставленій задачі.

Спосіб полягає у використанні системи, яка дозволяє отримувати розширені відповіді на запити, використовуючи дані із зовнішніх джерел[20], замість звичайної БД рекомендаційної системи. Удосконалення методу передбачає додаткову фільтрацію даних із зовнішніх джерел.

Таким чином необхідно отримати спосіб формування релевантних прогнозів, який працював би в умовах відсутніх або розріджених даних. Це

можливо на основі моделі системи, яка дозволяє отримувати розширені відповіді на запити, використовуючи дані із зовнішніх джерел.

Вихідними даними для рішення поставленої задачі є:

- блочна схема архітектури рекомендаційної системи, яка розглядається у підрозділі 2.1;
- модель системи, яка дозволяє отримувати розширені відповіді на запити, використовуючи дані із зовнішніх джерел;
- метод побудови рекомендацій в умовах холодного старту з використанням темпоральних обмежень за вхідними даними.

Рішення поставленої задачі передбачає виконання наступних кроків:

- адаптація моделі системи, яка дозволяє отримувати розширені відповіді на запити, використовуючи дані із зовнішніх джерел;
- удосконалення методу побудови рекомендацій в умовах холодного старту з використанням темпоральних обмежень за вхідними даними на основі додаткової фільтрації даних із зовнішніх джерел;
- програмна реалізація;
- тестування результатів.

1.5 Висновки до розділу

У підрозділі 1.1 були розглянуті загальні питання щодо рекомендаційних систем і визначене коло проблем, пов'язаних з їх розробкою та експлуатацією.

У підрозділі 1.2 досліджені методи побудови рекомендаційних систем із зазначенням їх переваг та недоліків. Традиційними підходами є колаборативна фільтрація та тематична або контентна фільтрація. Також

існує гібридна фільтрація, яка поєднує в собі колаборативну та контентну фільтрацію.

У підрозділі 1.3 виконано дослідження проблеми холодного старту та розглянуті підходи до її вирішення.

У підрозділі 1.4 визначена проблема магістерської атестаційної роботи та підходи до розв'язання цієї проблеми, а також виконано постановку задачі магістерської роботи.

2 ДОСЛІДЖЕННЯ МЕТОДІВ ВИРІШЕННЯ ПРОБЛЕМИ ХОЛОДНОГО СТАРТУ

2.1 Дослідження особливостей побудови рекомендаційної системи в умовах холодного старту

Проблема холодного старту та її формальне рішення залежать від особливостей РС.

Одним з головних завдань будь-якої рекомендаційної системи Інтернет-магазину є виявлення зв'язків – які товари з якими разом зазвичай купують люди. На основі цих даних можна також формувати груп людей за схожістю покупок, тому що у більшості випадків це дозволяє робити висновки, що для покупців 1 і 2, які робили схожі покупки, можна рекламувати одні і ті самі товари.

Існує три основні складові рекомендаційних систем, які мають велике значення під час формування рекомендацій: перша – це користувачі (users), друга – це товари (items), третя – події (events). Тоді подією E слід вважати те, що користувачеві U сподобався (або не сподобався) товар I . Можна також ввести поняття оцінки R , яка характеризує, наскільки сподобався товар. Тобто подією $e [u, i]$ слід вважати те, що U поставив товару I оцінку R . Таким чином можна сказати, що введена закономірність є “базовою механікою” будь-якої рекомендаційної системи.

Зважаючи на це слід зауважити, що основне завдання РС полягає у тому, щоб передбачити спираючись тільки на наявні дані про користувача, яку саме оцінку R користувач U поставить товару I . Знаючи оцінки можна давати рекомендації, наприклад, прогнози типу, якщо для товару I користувач U поставив високу оцінку, то доцільно цьому користувачеві про цей товар розповісти. Нехай існує пересічний набір товарів, то можна їх

ранжувати за прогнозованими R , потім сформувати п'ятірку кращих, виходячи з максимуму R , і запропонувати цю п'ятірку товарів користувачеві.

Зазвичай РС зберігають дані про користувачів, товари, події та оцінки у вигляді матриці. Наприклад, у рядках зазначені користувачі, у стовпцях – товари, а на перетинах – оцінки.

Таким чином користувачеві рекомендують товари, схожі на ті які він вже купував. Схожість визначають за ознаками об'єктів (наприклад, тип товару, виробник, ціна, тощо). Слід зауважити, що рекомендації залежать від предметної області, тобто мають обмежене застосування (РС для товарів і для фільмів будуть різними).

Коллаборативна фільтрація для рекомендацій використовує історію оцінок користувачів. Такий підхід сьогодні є найбільш поширеним через свою зрозумілість, а особливо через те, що добре зводиться до завдань машинного навчання.

Користувачів розбивають на велику кількість груп і вважають, що користувачі однієї групи мають схожі вподобання та цікавляться однаковими товарами. Якщо один з групи придбав або оцінив новий товар (новий для групи), тоді можна використати дані про це для всієї групи.

Детальніше розглянемо архітектуру рекомендаційної системи Інтернет–магазину.



Рис. 2.1 – Блочна схема архітектури рекомендаційної системи

Об'єкти, які наявні у системі – це користувачі, товари та події. Зрозуміло, що необхідно зібрати якнайбільше даних про кожну з цих складових.

На першому етапі збирають дані про користувачів (з реєстраційних анкет, оцінок, тощо). Дані про товар вже наявні у магазині, а для їх зберігання достатньо звичайної інформаційної бази підприємства. Так само і дані про користувачів можуть зберігатися у вигляді звичайних таблиць на сервері магазину.

На відміну від цих даних дані про транзакції та дії користувачів (вибір товару до кошика, лайки та ін.) можуть займати приблизно 15 ГБ на день (близько 100 подій на секунду). Для зберігання таких обсягів даних не підійде звичайна реляційна база даних. Рекомендаційна система, наведена на рис. 2.1, це приклад роботи з Big Data.

Дані про дії користувачів отримують завдяки лічильникам, які стоять практично на всіх сайтах і фіксують практично будь-які дії. Ці дані через такі інструменти як RabbitMQ і Flume надходять до черги і складаються у файлову систему HDFS. Механізм Hive у цій системі дозволяє за допомогою простих запитів отримувати необхідні вибірки з розподіленої файлової системи. Алгоритм працює з даними в Hadoop-кластері, куди також додаються дані про користувачів і товари через Sqoop.

Сформовані рекомендації відправляють до БД. Таке рішення з розподілу даних та їх обробки дозволяє отримати швидку видачу результатів.

Швидкість видачі є ключовим моментом тому, що користувач може піти з сайту до того як, його дані обробили і йому будуть видані рекомендації.

Hadoop-кластер не обов'язково повинен бути розташований в середині компанії. Проте більшість компаній, які використовують цей механізм, вважає, що це конкурентна перевага і краще його створити у себе.

Однією з проблем для Інтернет-магазину є проблема ідентифікації користувача, який зайшов на сайт. З цією метою використовують, наприклад, карти лояльності. Зазвичай ідентифікація є більш складним завданням, ніж побудова РС.

Оскільки Hadoop-кластер написаний мовою Java, систему спочатку теж вирішили писати цією мовою. Проте проблеми, з якими зіткнулися математики через особливості Java та обмежений MapReduce API цієї мови, призвели до вибору на користь Apache Spark.

Використання Spark – це оптимальне рішення через те, що алгоритми машинного навчання є ітеративними. Таким чином є велика кількість логів, які знаходяться у кластері Hadoop, а саме у HDFS-системі, і з якими потрібно робити якісь ітеративні обчислення (порахувати щось за цими даними, а потім покласти результат назад). Операції, пов'язані з читанням дані з диска

до оперативної пам'яті, а потім вивантаження дані на назад диск – це досить дорогі операції.

Hadoop-кластері крім великої кількості жорстких дисків є оперативна пам'ять. Фреймворк Spark завантажує дані до оперативної пам'яті і працює саме в ній. На жорсткий диск записують тільки кінцевий результат великої кількості різних операцій, що суттєво скорочує число операцій вводу/виводу. Таким чином Spark забезпечує якісне та ефективне підключення до Hive-таблиць, та дозволяє прямо зі Spark діставати дані з кластера. У Spark є чимало бібліотек і, зокрема, готовий модуль “рекомендаційна система”, якій належним чином сприймає матрицю типу користувач/товар/оцінки, і видає достатньо релевантні рекомендації.

Саму рекомендаційна система написана мовою Hive. Зазвичай все, що можна сформулювати у вигляді SQL-запиту швидко обчислюється алгоритмом. Недолік полягає у тому, що якщо запит не виходить написати засобами SQL, то доведеться писати UDF (User-Defined Function) на Java.

Таким чином у якості висновку можна сказати, що для побудови рекомендаційної системи доцільно використовувати наступні інструменти.

40% Apache Spark (Python) + 50% Hive on TEZ + 10% Hive UDF (Java)

Цей підхід має наступні переваги: зручно парсити дані до Spark на Python, потім ці дані можна скласти до Hive-таблиці і працювати з ними за допомогою SQL-запитів. Оскільки дані зберігаються у кластері, з ними можна виконувати будь-які ETL-операції, готувати їх так, як це необхідно для подальшого формування рекомендацій. Ці дії займають майже половину всього часу, їх виконує інструмент Hive. Для складних перетворень користуються Hive UDF.

На шляху від створення прототипу до впровадження виходить 70% повторного використання коду, зазвичай, на UDF переписують тільки критичні за продуктивністю і не дуже складні функції, які є універсальними. На відміну від ML, де прототип доводиться потім переписувати на

продуктивній стадії на інших мовах програмування, у випадку з Apache Spark спочатку працюємо з продуктивною структурою і формуємо алгоритм, а коли він починає працювати, можна буде довести до продуктивного рішення порівняно швидко.

2.2 Дослідження методу побудови рекомендацій в умовах холодного старту з використанням темпоральних обмежень за вхідними даними

При вирішенні проблеми холодного старту використовуються гібридні методи, засновані на доповненні колаборативної фільтрації аналізом контекстних даних, а також з використанням демографічних даних [20]. Для обліку циклічних змін інтересів користувачів використовуються їх темпоральні характеристики. Дослідження в даному напрямку використовують два підходи: на основі моделі користувача і на основі фільтрації даних. [34].

Проблеми побудови рекомендацій в умовах холодного старту з використанням темпоральних залежностей розглядалися в роботах [34–37].

Особливий інтерес представляють методи з використанням темпоральних операторів типу “Next” та “Future” [34, 36].

Метод з використанням темпоральних обмежень типу “Next” доповнює набір вхідних даних перед виконанням колаборативної фільтрації, яка використовується для побудови рекомендацій. Доповнення даних виконується на основі використання темпоральних залежностей. Темпоральні залежності мають вигляд правил, що пов’язують пари подій у часі. Такими подіями можуть бути вибір визначеної підмножини товарів та послуг для заданої підмножини користувачів[36,38].

Мета доповнення вхідних даних полягає в тому, щоб поставити для нового, “холодного” користувача обмеження у вигляді допустимих темпоральних залежностей вибору товарів або послуг.

В якості вихідних даних в рекомендаційних системах зазвичай використовується матриця E вибору (наприклад, покупок) споживача. Дана матриця є розрідженою, тобто лише незначна частина елементів матриці є ненульовий і відображає зв'язок користувачів з речами.

Кожен елемент e_{kj} цієї матриці характеризує вибір споживачем u_k з безлічі споживачів U будь-якої речі (товару, послуги) i_j . Кожен ненульовий елемент e_{kj} містить або значення рейтингу товару i_j , або кількість куплених споживачем u_k одиниць цього товару. Надалі для спрощення будемо вважати, що елемент e_{kj} містить кількість куплених одиниць товару.

Традиційно в такій матриці підсумовується інформація про кожний вибір (купівлю) e_{kj}^n товару i_j користувачем u_k .

Цей вибір здійснюється послідовно протягом заданого інтервалу часу $[\tau_1, \tau_N]$.

Темпоральне правило послідовного вибору об'єктів в рекомендаційній системі задає пару послідовних подій вибору товарів або послуг і користувачем в моменти часу і на заданому рівні деталізації часу. Дане правило задається з використанням темпорального оператора (Next) [34].

$$\exists u_k : (e_{kj}, \tau_n) X (e_{km}, \tau_l) \mid \mid \tau_n - \tau_l \mid \approx \Delta \tau \quad (2.1)$$

Відповідно до правила (1) проміжні покупки на рівні деталізації часу $\Delta \tau$ відсутні.

Темпоральне обмеження при послідовному виборі об'єктів в рекомендаційній системі задає пару послідовних подій, для яких задано строгий порядок на заданому рівні деталізації часу $\Delta\tau$:

$$\exists j, m: \forall k (e_{kj}, \tau_n) X(e_{km}, \tau_l) \mid \tau_n - \tau_l \approx \Delta\tau \quad (2.2)$$

В силу умови $\forall k$ обмеження (2.2) буде надалі представлятися без індексу користувача k , парою (e_j, e_m) .

Обмеження (2.2) можуть бути використані для доповнення інформації про нового користувача. Такі залежності моделюють зміна інтересів користувачів і дозволяють модифікувати вихідні дані в ситуації циклічного холодного старту.

Формування та використання темпоральних обмежень при побудові рекомендацій виконується в рамках розробленого методу. Метод використовує темпоральні залежності для заданого рівня грануляції часу.

В якості вхідних даних використовуються: журнал продажів, що містить набір трійок $L = \{(u_k, i_j, \tau_n)\}$; часовий інтервал $[\tau_1, \tau_N]$ для вхідних даних; рівень деталізації часу $\Delta\tau$; порогова кількість повторень правила α . Останній параметр використовується при виборі темпоральних обмежень.

Метод включає в себе наступні етапи.

Етап 1. Відбір підмножини записів журналу продажів за заданий інтервал часу $[\tau_1, \tau_N]$. Цей інтервал залежить від предметної області. Наприклад, для завдання бронювання готелів такий інтервал може становити рік, продажу одягу – сезон, і тому подібне. Умова відбору на даному етапі має вигляд: $\tau_n \in [\tau_1, \tau_N]$.

Етап 2. Узагальнення подій вибору для заданого рівня деталізації часу $\Delta\tau$. На даному етапі значення τ_n узагальнюються відповідно до рівня деталізації – до годин, днів, тощо.

Етап 3. Формування пар (e_j, e_m) послідовного вибору (покупок) користувачів на заданому рівні грануляції часу. Результатом даного етапу є множина R , яка включає пари (e_j, e_m) і кількість повторів цих пар n_{jm} на наборі вхідних даних:

$$R = \left\{ (e_j, e_m, n_{jm}) : e_j \times e_m, n_{jm} \geq 2 \right\} \quad (2.3)$$

Етап 4. Формування темпоральних обмежень C для вибору користувачів:

$$C = \left\{ (e_j, e_m) : \forall j \left| \{e_j\} \right| = \left| \{e_j, e_m\} \right| = n_{jm} \right\} \quad (2.4)$$

Обмеження (2.4) означає, що після вибору товару i_j користувачі обов'язково вибирають товар i_m . Кількість випадків вибору становить n_{jm} .

Етап 5. Відбір темпоральних обмежень для користувачів.

На даному етапі для кожного користувача з кожної множини C відбирається підмножина обмежень, для яких $n_{jm} > \alpha$:

$$C_\alpha = \left\{ (e_j, e_m) : n_{jm} > \alpha \right\} \quad (2.5)$$

Результатом даного етапу є безліч обмежень по всім користувачам.

Етап 6. Доповнення вхідних даних L записами обмежень згідно (2.5) для нових, “холодних” користувачів.

Отриманий журнал L^{Cold} має наступний вигляд:

$$L^{Cold} = L \cup \left\{ (u_k^{Cold}, i_m, \tau_l) : n_{jm} > \alpha \right\} \quad (2.6)$$

Етап 7. Побудова рекомендацій з використанням традиційних методів, зокрема колаборативної фільтрації [39]. Результатом даного етапу є перелік рекомендованих товарів для нового користувача, який враховує цикли зміни інтересів відомих користувачів.

Однак, даний метод видає в якості рекомендацій в умовах холодного старту найбільш відвідувані сторінки за вказаний проміжок часу. При цьому не враховується додаткова інформація про користувача, що призводить до необхідності аналізу всього списку продажів за вказаний проміжок часу.

В той же час можливість отримання додаткової інформації, наприклад, місце проживання (або місце входу в мережу), стать, вік і тощо, дозволяє істотно звузити область пошуку і відповідно скоротити загальний час виконання методу. В результаті, видача рекомендацій виконується за більш короткий час, так як обробляється не всі дані, а тільки певний набір даних.

2.3 Удосконалення методу побудови рекомендацій в умовах холодного старту з використанням темпоральних обмежень за вхідними даними

Метод, викладений у п. 2.2, має наступні недоліки: при формуванні рекомендацій переглядається весь масив даних, що потребує значну кількість часу. Тому проблема скорочення часу обробки даних за рахунок звуження області пошуку залишається відкритою.

Таким чином, метод побудови рекомендацій в умовах холодного старту з використанням темпоральних обмежень за вхідними даними доцільно удосконалити за рахунок введення фільтрації даних на підставі додаткової інформації про користувача.

Доповнимо етап 5 наступними кроками.

Етап 5.

Крок 1. Аналіз набору даних, які повідомляє про себе користувач в ході реєстрації.

Якщо користувач при реєстрації повідомляє тільки ім'я l і пароль p то необхідно перейти до кроку 3.

Якщо користувач при реєстрації повідомляє про себе необов'язкові додаткові дані (стать s , місце розташування q , вік d), то необхідно перейти до кроку 2.

Крок 2. Виділення підмножини товарів, якими найбільш сильно цікавиться підмножина користувачів з аналогічними додатковими характеристиками, наприклад, всі чоловіки більше 40 років або всі жителі міста Харкова. Результати вибору представити у вигляді тимчасового масиву даних та перейти до кроку 3

Вводимо наступні позначення:

$Event(s, q, d)$ – подія, коли користувач U_m вводить додаткові данні s, q, d , де: s – стать користувача;

q – місце розташування користувача;

d – вік користувача, $d_1 < d < d_2$;

$U = \{U_m\}$ – множина користувачів.

$U_m^{Lim} = \{s, q, d\}$ – підмножина користувачів яка сформувалася на основі відбору даних по фільтру $Lim^1 = \{s, q, d\}$;

$E = \{E_n\}$ – множина товарів;

$E_n^{Lim^2} = \{e_j \dots e_n\}$ – підмножини товарів, якими найбільш сильно цікавиться підмножина користувачів $U_m^{Lim^1}$ (на основі фільтру $Lim^2 = \{c, n, z\}$);

$Lim^1 = \{s, q, d\}$ – обмеження за додатковими характеристиками: $\{s, q, d\}$;

$U = \{U_m\}$ – множина користувачів;

$U_m^{lim} = \{s, q, d\}$ – підмножина користувачів яка сформувалася на основі відбору даних по фільтру $Lim^1 = \{s, q, d\}$;

$E = \{E_n\}$ – множина товарів;

$E_n^{Lim^2} = \{e_j \dots e_n\}$ – підмножини товарів, якими найбільш сильно цікавиться підмножина користувачів $U_m^{Lim^1}$ (на основі фільтру $Lim^2 = \{c, n, z\}$).

$Lim^2 = \{c, n, z\}$ – обмеження за додатковими характеристиками: $\{c, n, z\}$, де c – множина кодів товару; n – множина назв товару; z – множина цін.
Правило формування підмножини користувачів:

$$if Event(s, q, d) then Lim^1 = \{s, q, d\} then U_m^{Lim^1} = \{U_1 \dots U_m\} \quad (2.7)$$

Правило формування підмножини товарів :

$$if U_m^{Lim^1} = \{U_1 \dots U_m\} then Lim^2 = \{c, n, z\} then E_n^{Lim^1} = \{e_j \dots e_n\} \quad (2.8)$$

При цьому виконується відповідність між підмножиною товарів і підмножиною користувачів:

$$U \rightarrow E \quad (2.9)$$

Крок 3. Відбір темпоральних обмежень для користувачів.

На даному етапі для кожного користувача з кожної множини C відбирається підмножина обмежень, для яких $n_{jm} > \alpha$:

$$C_\alpha = \left\{ (e_j, e_m) : n_{jm} > \alpha \right\} \quad (2.11)$$

Результатом даного етапу є безліч обмежень по всім користувачам.

Темпоральні правила формуються не для всіх товарів і користувачів, а тільки для однієї підмножини товарів і однієї підмножини користувачів, які були виділені на кроках 1 і 2.

Таким чином модифікація методу побудови рекомендацій в умовах холодного старту з використанням темпоральних обмежень за вхідними даними відбувається шляхом додавання нових етапів фільтрації даних на підставі додаткової інформації про користувача. В наслідок цього, видача рекомендацій здійснюється за більш короткий час тому, що обробляється не вся база даних, а певний набір даних.

2.4 Висновки до розділу

У підрозділі 2.1 виконано дослідження особливостей побудови рекомендаційних систем, детально розглянута архітектура рекомендаційної системи Інтернет–магазину.

У підрозділі 2.2 проведено дослідження методів вирішення проблеми холодного старту, виконан аналіз методу побудови рекомендацій в умовах холодного старту з використанням темпоральних обмежень за вхідними даними. Метод з використанням темпоральних обмежень доповнює набір вхідних даних перед виконанням колаборативної фільтрації, яка використовується для побудови рекомендацій.

У підрозділі 2.3 виконано удосконалення методу побудови рекомендацій в умовах холодного старту з використанням темпоральних обмежень за рахунок введення фільтрації даних на підставі додаткової інформації про користувача.

Запропонований метод дозволяє підвищити ефективність побудови рекомендацій в умовах холодного старту на основі скорочення часу обробки відфільтрованого набору даних.

3 ОПИС ВПРОВАДЖЕННЯ РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ ДО ІТ-ПРОДУКТУ

3.1 Опис рішень з ініціації ІТ-проекту

Оснoву успішного впровадження будь-якої розробки становить задоволення потреб і очікувань зацікавлених сторін: користувачів послуг, замовників розробки, партнерів, інвесторів, працівників та ін.

Діяльність, пов'язана з розробкою будь-якого проекту починається з розробки концепції проекту та вибору стратегічних рішень виконання проекту та деталей.

Першим етапом роботи над проектом є його ініціація.

Можна сказати, що ініціація – це процес формального визнання необхідності виконання проекту.

У окремих організаціях проект не підлягає формальній ініціації до тих пір, поки не виконана якась форма попереднього аналізу, який часто є самостійним проектом. Принципи, за якими обирають проекти, можна занести до однієї з наступних категорій:

- вимірювання передбачуваного прибутку (економічні моделі, порівняльні оцінки, тощо);
- оптимізація (математичні моделі, які використовують лінійне, нелінійне, динамічне та інше програмування).

Розроблене програмне рішення підпадає під другу категорію.

Таким чином у результаті діяльності з ініціалізації проекту був сформований статут проекту, елементи якого наведені у таблиці 3.1.

Таблиця 3.1 – Елементи статуту проекту.

Параметр	Вимоги та рекомендації
1	2
Назва проекту	Рекомендаційна система, заснована на методі колаборативної фільтрації типу user-based
Замовник проекту	Керівництво Інтернет-магазину
Мета проекту	Створення програмного продукту для подолання проблем холодного старту у рекомендаційних системах, заснованих на методі колаборативної фільтрації типу user-based
Концепція проекту	<ul style="list-style-type: none"> – програмний додаток, призначений для прогнозування інтересів користувачів на підставі інформації про їх профілі; – збирає та фільтрує дані про користувачів; – розроблене програмне рішення базується на сайті Інтернет-магазину і не має власного інтерфейсу користувача.
Обґрунтування ініціації проекту	На підставі досліджень, проведених ініціатором проекту, були виявлені недоліки, пов'язані з фільтрацією даних під час використання алгоритму..... для методу колаборативної фільтрації типу user-based
Ролі та функції команди проекту	Менеджер проекту, програмісти, тестувальники, консультант з предметної області, розробник БД, аналітик, технічний письменник
Споживачі продукту проекту	Користувачі Web-сайту Інтернет-магазину

Кінець таблиці 3.1

1	2
Вимоги до програмного продукту	Програмний продукт має серверну частину, яка потребує наявності сервера та СКБД. Клієнтська частина відсутня, система використовує сайт Інтернет–магазину для відображення результатів фільтрації
Ризики проекту	– велика кількість подібних розробок; – неможливо точно спрогнозувати результати застосування розробки;
Висновки про можливість реалізації проекту	Проект може бути реалізований у межах атестаційної роботи як навчальне завдання

У продовж фази ініціації проекту було виконано концептуальне планування майбутнього проекту та планування робіт тимчасової робочої групи проекту.

Головною відмінністю управління проектом на фазі ініціації зазвичай є невизначений термін завершення фази. Також існує велика ймовірність відмови від переходу до наступних фаз життєвого циклу проекту (виявлена економічна недоцільність, програш тендера, тощо).

Для чіткого розуміння необхідності здійснення проекту було вивчено потенційних споживачів продукту, створеного в результаті реалізації проекту, і визначені характеристики кінцевого продукту. Саме тому фаза ініціації є найважливішою фазою проведення підготовчих робіт.

На цій фазі було здійснено обґрунтування проекту і прийнято рішення про його подальшу розробку. Результати опрацювання ідеї дозволять уникнути проблем на всіх наступних фазах життєвого циклу розробки РС.

Часто через певні умови під час проведення робіт з ініціації починають виконувати роботи наступної фази проекту – планування, а, інколи, і фази виконання проекту – така ситуація є досить поширеною помилкою.

MBOK Guide of knowledge (Звід знань з управління проектами) визначає ініціацію як процес формального санкціонування нового проекту або переходу проекту до наступної фази [32]. Проте формальне санкціонування проекту у більшості ІТ компаній не відбувається без попереднього аналізу і планування. Роботу на цій фазі можна інколи розглядати як окремий проект, який може бути виконаний окремою командою з окремими термінами завершення і бюджетом.

Усі роботи з моделювання загальної ієрархічної структури проекту, побудови діаграми Ганта та форми ресурсів, формування WBS і OBS та інших елементів виконання задачі календарно–мереженого планування були виконані засобами MS Project 10.

Таким чином на етапі ініціації проекту було виконано моделювання робіт проектної групи, розроблено концепцію майбутнього програмного продукту, заповнені таблиці задач, використання задач, ресурсів та ін.

Під час проведення робіт з ініціації проекту підписуються різні документи, проводяться наради та ведеться листування з замовником. Результатом успішного завершення фази ініціації проекту є укладення підписання з замовником узгоджувальних документів на проведення робіт, приймається рішення про продовження проекту і видається наказ про ініціацію проекту. Таким чином проект переходить на наступну фазу життєвого циклу – планування.

3.2 Опис результатів планування IT–проекту

Будь–який проект починається для досягнення визначеного результату у визначений терміни. Саме за допомогою плану проекту можна визначити, які роботи дозволять досягти результатів проекту, які фахівці й устаткування потрібні для виконання цих робіт, і в який час ці фахівці та обладнання можуть працювати над проектом. Таким чином проектний план має містити три основні елементи: завдання (Task), ресурси (Resource) і призначення (Assignment).

Ці елементи у середовищі MS Project мають вигляд таблиць, які необхідно заповнити для подальшої роботи над проектом. Таблиці заповнюють на основі загального опису стадій та етапів проекту (табл. 3.2), встановлюють зв'язки між ресурсами та етапами робіт, задають тривалість етапів, призначають зв'язки між етапами (вказують попередників і послідовників) та виконують усі необхідні дії стосовно планування та вистежування проекту.

Повномасштабні проекти складаються з великої кількості операцій, тому описати їх нічого не пропустивши без структуризації (розбиття проекту на фази, підфази, пакети, тощо) практично неможливо. Найбільш поширений підхід до структуризації – це розбиття проекту на стадії та етапи виходячи з об'єктів проекту.

Таблиця 3.2 – Опис стадій та етапів створення рекомендаційної системи.

Стадії	Етапи робіт	Результат виконання
1	2	3
1. Формування вимог до системи	1.1 Дослідження предметної області 1.2 Формування вимог до проекту 1.3 Виконання необхідних науково–дослідних робіт 1.4 Розробка концепції системи	1. Повний та актуальний опис предметної області 2. Список вимог до РС 3. Результати досліджень 4. Опис концепції системи, підписана угода (контракт) або інші документи
2. Технічне завдання	2.1 Розробка технічного завдання для БД 2.2 Розробка технічного завдання для створення системи	1. Технічне завдання для БД 2. Технічне завдання для системи
3. Проектування БД	3.1 Вибір моделі БД 3.2 Адаптація моделі БД до вимог предметної області	1. Обрана модель БД 2. Адаптована модель
4. Технічний проект	4.1 Розробка БД 4.2 Тестування БД 4.3 Розробка програмних модулів системи	1. Функціонуюча БД 2. Результати тестування 3. Функціонуючі програмні модулі

Кінець таблиці 3.2

1	2	3
5. Тестування	5.1 Тестування модулів системи 5.2 Оцінка результатів тестування 5.3 Загальна оцінка проекту	1. Результати тестування 2, 3. Звіти
6. Оцінка поточного проекту	6.1 Аналіз готовності проекту до експлуатації 6.2 Розробка робочої документації для системи	1. Звіт 2. Робоча документація
7. Введення до експлуатації	7.1 Підготовка об'єкта до експлуатації 7.2 Інсталяція продукту 7.3 Перевірка працездатності після інсталяції 7.4 Наповнення БД 7.5 Синхронізація 7.6 Тестування 7.7 Впровадження	1. Звіт 2. Заповнена БД 3. Результати тестування 4. Документація про проходження тестування та експлуатації

Поділивши проект на об'єкти з максимальною деталізацією потрібно описати процеси, пов'язані з реалізацією кожного з них. Проте можливі й інші підходи до створення плану проекту, наприклад, можна почати з процесів, а потім описувати, до яких об'єктів ці процеси слід докласти в даному проекті.

З цією метою у проекті була створена структура декомпозиції робіт (WBS) та заповнений словник WBS, який містить повний опис робіт і технічну документацію на кожний об'єкт.

Таким чином WBS – це ієрархічна структура робіт, які виконує проектна група поки триває життєвий цикл проекту. Ця декомпозиція відображає загальний зміст проекту та етапи робіт, на які він поділений.

WBS може мати декілька рівнів деталізації. Їх кількість залежить від типу життєвого циклу проекту та інших факторів.

Ще одна корисна структура – структура відповідальності, в якій етапи проекту співвідносяться з фахівцями, що відповідають за їх виконання.

Структуру відповідальності використовують для відображення зв'язків між етапами робіт та фахівцями проектної групи. Крім того відповідальність вказує на ролі та рівні повноважень і привілеїв у проектній групі.

Для створення такої структури необхідно описати ресурси. У середовищі розробки MS Project зазначено три типи ресурсів: трудові, матеріальні та витратні. У пояснювальній записці до атестаційної роботи наведено тільки опис трудових ресурсів.

Трудові ресурси попередньо описані як перелік фахівців, що приймають участь у роботі над проектом (табл. 3.3), з позначенням ролі у проектній групі відповідно до етапів робіт.

Таблиця 3.3 – Перелік фахівців та їх ролі у проектній групі.

Етапи робіт	Фахівці (ролі)
1.1 Дослідження предметної області	Аналітик (Виконавець) Консультант з предметної області (Консультант)
1.2 Формування вимог до проекту	Менеджер проекту (Виконавець)
1.3 Виконання необхідних науково–дослідних робіт	Аналітик (Виконавець)
1.4 Розробка концепції системи	Аналітик (Виконавець) Консультант з предметної області (Консультант)

Продовження таблиці 3.3

Етапи робіт	Фахівці (ролі)
2.1 Розробка технічного завдання для БД	Технічний письменник (Виконавець)
2.2 Розробка технічного завдання для створення системи	Технічний письменник (Виконавець)
3.1 Вибір моделі БД	Розробник БД (Виконавець)
3.2 Адаптація моделі БД до вимог предметної області	Розробник БД (Виконавець)
4.1 Розробка БД	Розробник БД (Виконавець)
4.2 Тестування БД	Розробник БД (Узгодження) Тестувальник (Виконавець)
4.3 Розробка програмних модулів системи	Програміст (Виконавець) Тестувальник (Виконавець)
5.1 Тестування модулів системи	Програміст (Виконавець) Тестувальник (Виконавець)
5.2 Оцінка результатів тестування	Менеджер проекту (Узгодження) Програміст (Узгодження) Тестувальник (Узгодження)
5.3 Загальна оцінка проекту	Менеджер проекту (Виконавець)
6.1 Аналіз готовності проекту до експлуатації	Аналітик (Виконавець) Менеджер проекту (Узгодження)
6.2 Розробка робочої документації для системи	Технічний письменник (Виконавець)
7.1 Підготовка об'єкта до експлуатації	Менеджер проекту (Узгодження) Програміст (Виконавець) Розробник БД (Виконавець)

Кінець таблиці 3.3

Етапи робіт	Фахівці (ролі)
7.2 Інсталяція продукту	Програміст (Виконавець) Розробник БД (Виконавець)
7.3 Перевірка працездатності після інсталяції	Програміст (Виконавець) Розробник БД (Виконавець)
7.4 Наповнення БД	Розробник БД (Виконавець)
7.5 Синхронізація	Програміст (Виконавець) Розробник БД (Виконавець)
7.6 Тестування	Тестувальник (Виконавець)
7.7 Впровадження	Програміст (Виконавець) Розробник БД (Виконавець)

На основі цих даних безпосередньо у середовищі розробки MS Project побудована організаційна структура виконавців (OBS) та структура споживання ресурсів (RBS). Приклади форми ресурсів наведені на рис.3.1–3.3.

Елементи цих структур були задіяні для визначення попередників (етапи робіт, які передують іншим етапам) та наступників (етапи робіт, які йдуть слідом за іншими етапами). Попередники та наступники пов'язані визначеним типом залежності етапів і часовими проміжками випередження та запізнення.

Название: Менеджер проекту Краткое назв.: М Макс. ед: 100% [Предыдущий](#) [Следующий](#)

Затраты

Стандартная ставка: 0,00р./ч На использование: 0,00р.

Ставка сверхурочных: 0,00р./ч Начисление: Пропорциональное

Баз. календарь: Стандартный

Группа:

Код:

Проект	Ид.	Название задачи	удозатрат	звн. задер	Задержка	планированное нача.	запланированное окончание
Проект 1	3	1.2 Формування вимог до проекту	16ч	0д	0д	Чт 05.09.19	Пт 06.09.19
Проект 1	18	5.2 Оцінка результатів тестування	16ч	0д	0д	Чт 07.11.19	Пт 08.11.19
Проект 1	19	5.3 Загальна оцінка проекту	16ч	0д	0д	Чт 07.11.19	Пт 08.11.19
Проект 1	24	7.1 Підготовка об'єкта до експлуатації	16ч	0д	0д	Чт 21.11.19	Пт 22.11.19
Проект 1	5	1.4 Розробка концепції системи	16ч	0д	0д	Вт 17.09.19	Ср 18.09.19
Проект 1	21	6.1 Аналіз готовності проекту до експлуатації	16ч	0д	0д	Пн 11.11.19	Вт 12.11.19

Рис. 3.1 – Форма ресурсів (Менеджер проекту)

Название: Аналітик Краткое назв.: А Макс. ед: 100% [Предыдущий](#) [Следующий](#)

Затраты

Стандартная ставка: 0,00р./ч На использование: 0,00р.

Ставка сверхурочных: 0,00р./ч Начисление: Пропорциональное

Баз. календарь: Стандартный

Группа:

Код:

Проект	Ид.	Название задачи	удозатрат	звн. задер	Задержка	планированное нача.	запланированное окончание
Проект 1	4	1.3 Виконання необхідних науково-дослідницьких робіт	48ч	0д	0д	Пн 09.09.19	Пн 16.09.19
Проект 1	5	1.4 Розробка концепції системи	16ч	0д	0д	Вт 17.09.19	Ср 18.09.19
Проект 1	21	6.1 Аналіз готовності проекту до експлуатації	16ч	0д	0д	Пн 11.11.19	Вт 12.11.19
Проект 1	2	1.1 Дослідження предметної області	24ч	0д	0д	Пн 02.09.19	Ср 04.09.19

Рис. 3.2 – Форма ресурсів (Аналітик)

Название: Програміст Краткое назв.: П Макс. ед: 100% [Предыдущий](#) [Следующий](#)

Затраты

Стандартная ставка: 0,00р./ч На использование: 0,00р.

Ставка сверхурочных: 0,00р./ч Начисление: Пропорциональное

Баз. календарь: Стандартный

Группа:

Код:

Проект	Ид.	Название задачи	Трудозатраты	звн. задер	Задержка	планированное нача.	запланированное окончание
Проект 1	15	4.3 Розробка програмних модулів	80ч	0д	0д	Пн 21.10.19	Пт 01.11.19
Проект 1	17	5.1 Тестування модулів системи	24ч	0д	0д	Пн 04.11.19	Ср 06.11.19
Проект 1	25	7.2 Попередні підготовчі роботи	40ч	0д	0д	Пн 25.11.19	Пт 29.11.19
Проект 1	27	7.4 Впровадження системи	16ч	0д	0д	Ср 04.12.19	Чт 05.12.19
Проект 1	18	5.2 Оцінка результатів тесту	16ч	0д	0д	Чт 07.11.19	Пт 08.11.19
Проект 1	24	7.1 Підготовка об'єкта до експлуатації	16ч	0д	0д	Чт 21.11.19	Пт 22.11.19

Рис. 3.3 – Форма ресурсів (Програміст)

На рис. 3.4 наведене подання верхнього рівня деталізації, яке містить елементи WBS, OBS і RBS.

№	Назва задачі	Длитель	Начало	Окончание	Прод.	Назначение ресурсов
1	1. Формування вимог до системи	13 днів	Пн 02.09.19	Ср 18.09.19		
2	1.1 Дослідження предметної області	3 днів	Пн 02.09.19	Ср 04.09.19		Аналітик; Консультант в предметній області
3	1.2 Формування вимог до проекту	2 днів	Чт 05.09.19	Пт 06.09.19	2	Менеджер проекту
4	1.3 Виконання необхідних науково-дослідних робіт	6 днів	Пн 09.09.19	Пн 16.09.19	3	Аналітик
5	1.4 Розробка концепції системи	2 днів	Вт 17.09.19	Ср 18.09.19	4	Аналітик; Консультант в предметній
6	2. Технічне завдання	3 днів	Чт 19.09.19	Ср 25.09.19	1	
7	2.1 Розробка технічного завдання для БД	3 днів	Чт 19.09.19	Пн 23.09.19	5	Технічний письменник
8	2.2 Розробка технічного завдання для створення системи	3 днів	Вт 24.09.19	Чт 26.09.19	7	Технічний письменник
9	3. Проектування БД	5 днів	Пт 27.09.19	Чт 03.10.19	8	
10	3.1 Вибір моделі БД	2 днів	Пт 27.09.19	Пн 30.09.19	8	Розробник БД
11	3.2 Адаптація моделі БД до вимог предметної області	3 днів	Вт 01.10.19	Чт 03.10.19	10	Розробник БД
12	4. Технічний проект	21 днів	Пт 04.10.19	Пт 01.11.19	9	
13	4.1 Розробка БД	8 днів	Пт 04.10.19	Вт 13.10.19	11	Розробник БД
14	4.2 Тестування БД	3 днів	Ср 16.10.19	Пт 18.10.19	13	Розробник БД; Тестувальник
15	4.3 Розробка програмних модулів системи	10 днів	Пн 21.10.19	Пт 01.11.19	14	Програміст
16	5. Тестування	5 днів	Пн 04.11.19	Пт 08.11.19	12	
17	5.1 Тестування модулів системи	3 днів	Пн 04.11.19	Ср 06.11.19	15	Програміст; Тестувальник
18	5.2 Оцінка результатів тестування	2 днів	Чт 07.11.19	Пт 08.11.19	17	Менеджер проекту; Програміст
19	5.3 Загальна оцінка проекту	2 днів	Чт 07.11.19	Пт 08.11.19	17	Менеджер проекту
20	6. Оцінка поточного проекту	8 днів	Пн 11.11.19	Ср 20.11.19	16	
21	6.1 Аналіз готовності проекту до експлуатації	2 днів	Пн 11.11.19	Вт 12.11.19	19	Аналітик; Менеджер проекту
22	6.2 Розробка робочої документації для експлуатації	6 днів	Ср 13.11.19	Ср 20.11.19	21	Технічний письменник
23	7. Введення до експлуатації	11 днів	Чт 21.11.19	Чт 05.12.19	20	
24	7.1 Підготовка об'єкта до експлуатації	2 днів	Чт 21.11.19	Пт 22.11.19	22	Менеджер проекту; Програміст
25	7.2 Інсталяція продукту	2 днів	Пн 25.11.19	Вт 26.11.19	24	Програміст; Розробник БД
26	7.3 Перевірка прохідності після інсталяції	1 день	Ср 27.11.19	Ср 27.11.19	25	Програміст; Розробник БД
27	7.4 Наповнення БД	2 днів	Чт 28.11.19	Пт 29.11.19	26	Розробник БД
28	7.5 Синхронізація	1 день	Пн 02.12.19	Пн 02.12.19		Програміст; Розробник БД
29	7.6 Тестування	2 днів	Вт 03.12.19	Ср 04.12.19		Тестувальник
30	7.7 Впровадження	1 день	Чт 05.12.19	Чт 05.12.19		Програміст; Розробник БД
31	- підпис погоджувальних документів	1 день	Пт 06.12.19	Пт 06.12.19		Консультант з предметної
32	- усунення недоліків	3 днів	Пт 06.12.19	Вт 10.12.19		Програміст; Розробник БД

Рис. 3.4 – Декомпозиція етапів робіт.

Різні організації застосовують різні підходи до подання даних (особливо проектних даних). Аналіз звичайного списку задач і декількох відрізків не може організувати необхідне уявлення про стан проекту.

Дані комп'ютерного моделювання проектів можуть бути наведені у різному вигляді: табличному, графічному та ін. (рис. 3.5).

Реж. зада	Название задачи	Трудозатр	Длительность	Начало	Окончание	Добавить на столбец	Подробнее
1	1. Формування вимог до системи	0 ч	13 дней	Пн 02.09.19	Ср 18.09.19		Трудозатр.
2	1.1 Дослідження предметної області	48 ч	3 дней	Пн 02.09.19	Ср 04.09.19		Трудозатр.
	Аналітик	24 ч		Пн 02.09.19	Ср 04.09.19		Трудозатр.
	Консультант з предметної області	24 ч		Пн 02.09.19	Ср 04.09.19		Трудозатр.
3	1.2 Формування вимог до проекту	16 ч	2 дней	Чт 05.09.19	Пт 06.09.19		Трудозатр.
	Менеджер проекту	16 ч		Чт 05.09.19	Пт 06.09.19		Трудозатр.
4	1.3 Виконання необхідних науково-дослідних робіт	48 ч	6 дней	Пн 09.09.19	Пн 16.09.19		Трудозатр.
	Аналітик	48 ч		Пн 09.09.19	Пн 16.09.19		Трудозатр.
5	1.4 Розробка концепції системи	48 ч	2 дней	Вт 17.09.19	Ср 18.09.19		Трудозатр.
	Аналітик	16 ч		Вт 17.09.19	Ср 18.09.19		Трудозатр.
	Менеджер проекту	16 ч		Вт 17.09.19	Ср 18.09.19		Трудозатр.
	Консультант з предметної області	16 ч		Вт 17.09.19	Ср 18.09.19		Трудозатр.
6	2. Технічне завдання	0 ч	5 дней	Чт 19.09.19	Ср 25.09.19		Трудозатр.
7	2.1 Розробка технічного завдання для БД	24 ч	3 дней	Чт 19.09.19	Пн 23.09.19		Трудозатр.
	Технічний письменник	24 ч		Чт 19.09.19	Пн 23.09.19		Трудозатр.
8	2.2 Розробка технічного завдання для створення системи	24 ч	3 дней	Вт 24.09.19	Чт 26.09.19		Трудозатр.
	Технічний письменник	24 ч		Вт 24.09.19	Чт 26.09.19		Трудозатр.
9	3. Проектування БД	0 ч	5 дней	Пт 27.09.19	Чт 03.10.19		Трудозатр.
10	3.1 Вибір моделі БД	16 ч	2 дней	Пт 27.09.19	Пн 30.09.19		Трудозатр.
	Розробник БД	16 ч		Пт 27.09.19	Пн 30.09.19		Трудозатр.
11	3.2 Адаптація моделі БД до вимог предметної області	24 ч	3 дней	Вт 01.10.19	Чт 03.10.19		Трудозатр.
	Розробник БД	24 ч		Вт 01.10.19	Чт 03.10.19		Трудозатр.
12	4. Технічний проект	0 ч	21 дней	Пт 04.10.19	Пт 01.11.19		Трудозатр.
13	4.1 Розробка БД	64 ч	8 дней	Пт 04.10.19	Вт 15.10.19		Трудозатр.
	Розробник БД	64 ч		Пт 04.10.19	Вт 15.10.19		Трудозатр.

Рис. 3.5 – Таблична форма подання використання задач

Засобами MS Project була сформована діаграма Ганта, яка дозволяє налаштувати відображення плану проекту.

Діаграма Ганта – це різновид стовпчастих діаграм (гістограм), який використовують для ілюстрації плану, графіка робіт проекту. Цей тип діаграм є одним з поширених методів планування проектів і фактичним стандартом у практиці та теорії управління проектами, який відображає структуру етапів робіт.

Діаграма Ганта представляє собою відрізки, розташовані за горизонтальною шкалою часу, кожен з яких відповідає стадіям та етапам робіт. Початок, кінець і довжина відрізків на шкалі часу відповідає початку, кінцю і тривалості стадії або етапу. Етапи робіт і ресурси проекту у вигляді діаграми Ганта наведені на рис. 3.6. Вертикальні лінії пов'язують етапи робіт і вказують на те, що наступна робота починається після завершення попередньої (життєвий цикл каскадного типу).

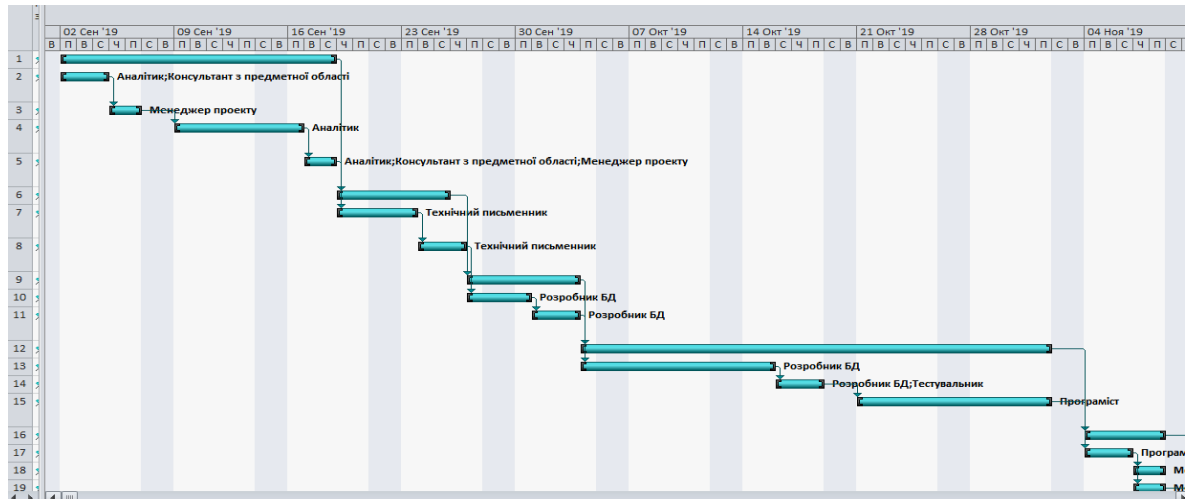


Рис. 3.6 – Діаграма Ганта

Відповідно до результатів досліджень, проведених різними ІТ компаніями, ймовірність успішної реалізації проекту, термін здачі якого визначений за типовими показниками, становить приблизно 20–38%, тому разом з іншими діями з управління проектом необхідно провести облік ризиків і невизначеностей.

У різних програмних засобах з цією метою використовують різні методи, наприклад, PERT–оцінку або імітаційне моделювання (метод Монте–Карло).

Аналіз невизначеностей, який зазвичай виникає під час виконання складеного плану, – це один зі складних етапів планування проекту. Від проведення цього аналізу, залежить вчасне виконання проекту.

Аналіз ризиків у MS Project складається з декількох етапів. Потрібно визначити можливі ризики, для кожного з них визначити стратегію пом'якшення впливу ризику на проект та ін.

Якщо план проекту складається з великої кількості стадій та етапів робіт, неможливо детально проаналізувати весь план. У таких випадках у першу чергу потрібно аналізувати ризики етапів робіт, які можуть стати критичними. Щоб правильно визначити критичні завдання, у атестаційні

роботі були побудовані оптимістичні та песимістичні діаграмами Ганта та виконаний аналіз методом PERT за формулою:

$$PERT = (Pert_{\text{опт}} + Pert_{\text{пес}} + 4 * Pert_{\text{факт}}) / 6 \quad (3.1)$$

де PERT – очікувана тривалість операцій; $Pert_{\text{опт}}$ – оптимістична тривалість операцій; $Pert_{\text{пес}}$ – песимістична тривалість операцій; $Pert_{\text{факт}}$ – фактична тривалість операцій.

Таким чином були змодельовані три варіанти розвитку подій для проекту: песимістичний, оптимістичний і фактичний. Виходячи з цих даних була розрахована PERT–оцінка для всіх стадій проекту. Наприклад для першої стадії відхилення від фактичного показнику розкладу (у днях) становить:

$$(10 + 15 + 4 * 13) / 6 = 12,8 \text{ дня}$$

Для інших стадій відхилення у розкладі становить не більше одного дня. Судячи з результатів, розподіл усіх варіантів часових проміжків виконання проекту практично збігається з фактичним, що означає, що економія часу не сильно істотна, а попередня оцінка була зроблена на достатньому рівні.

Оскільки ризики зазвичай визначають для трьох аспектів проекту: розкладу, ресурсів і бюджету, так само були обчислені ризики трудових ресурсів.

Таким чином можна зробити висновки, що події, які можуть перешкодити вчасному завершенню проекту або створити нестачу ресурсів у певний момент його виконання, загалом несуттєво вплинуть на проект.

Якщо при визначенні ризику стає ясно, як зменшити його, то потрібно відразу ж вносити відповідні зміни в план проекту.

Декомпозиція видів робіт (рис. 3.4) та діаграма Ганта (рис. 3.6) у роботі наведені для фактичного варіанту розвитку подій.

3.3 Аналіз критичного шляху проекту

Під час планування та виконання проекту частину операцій виконують паралельно, тому існує декілька можливих шляхів проходження кожного графа. Операції, які займають тривалий час, називають критичними. Зміна терміну початку або закінчення виконання цих операцій може викликати затримку завершення всього проекту. Послідовність операцій такого типу є критичним шляхом проекту. Таким чином можна сказати, що тривалість критичного шляху визначає загальну тривалість проекту.

Слід зазначити, що слово “критичний” у цьому контексті не має нічого спільного з тим, наскільки важливі ці завдання для проекту. Воно вказує лише на те, як їх розклад буде впливати термін здачі проекту. Проте дата закінчення багатьох проектів має важливе значення. Якщо потрібно скоротити тривалість проекту, щоб він закінчився увизначений день, потрібно почати скорочувати критичний шлях.

Під час протікання ЖЦ проекту критичний шлях може змінюватися через зміну планового терміну завершення завдань. Зміни у плані, такі як визначення зав’язків між завданнями або тривалостей задач, також можуть призвести до зміни критичного шляху. Після завершення завдання на критичному шляху, цей шлях перестає бути критичним, оскільки не впливає на дату закінчення проекту.

Резерв часу є основним показником, який характеризує критичний шлях. Існує два типи резерву часу: вільний і загальний. Вільний резерв часу – це час, на який може бути відкладене завдання, до того, як

вонозапустити інше. Загальний резерв часу – це час, на який може бути відкладене завдання до того як вонозапустить завершення проекту.

Завдання знаходиться на критичному шляху, якщо його загальний резерв часу менше визначеного значення (у MS Project за замовчуванням = 0 днів). Навпаки, некритичні завдання мають резерв часу, що означає, що вони можуть починатися або закінчуватися раніше або пізніше у межах свого резерву, не впливаючи на дату завершення проекту.

Таким чином для аналізу критичного шляху проекту потрібно застосувати метод критичного шляху. Цей метод, в основі якого лежить математичний алгоритм, дозволяє планувати операції.

Метод критичного шляху у атестаційній роботі був використаний для порівняння запланованого і фактичного ходу виконання проекту

Для використання методу необхідно створити модель проекту, яка містить список всіх операцій проекту, залежності між цими операціями, період часу виконання кожної операції (тривалість). Модель проекту атестаційної роботи описана та наведена у підрозділі 3.2 за допомогою табл. 3.3 та рис. 3.1–3.5.

Для відображення критичного шляху проекту у атестаційній роботі були застосовані засоби MS Project, який дозволяє сформувати подання зображення критичного шляху, визначити час, на який можна затримати виконання завдання не збільшуючи часовий проміжок виконання проекту, додати ресурси для пришвидшення виконання завдання зі зберіганням об'єму або розбити завдання на менші частини, які будуть виконувати різні працівники. На рис. 3.7 наведено докладну діаграму Ганта, яка візуалізує критичний шлях проекту.

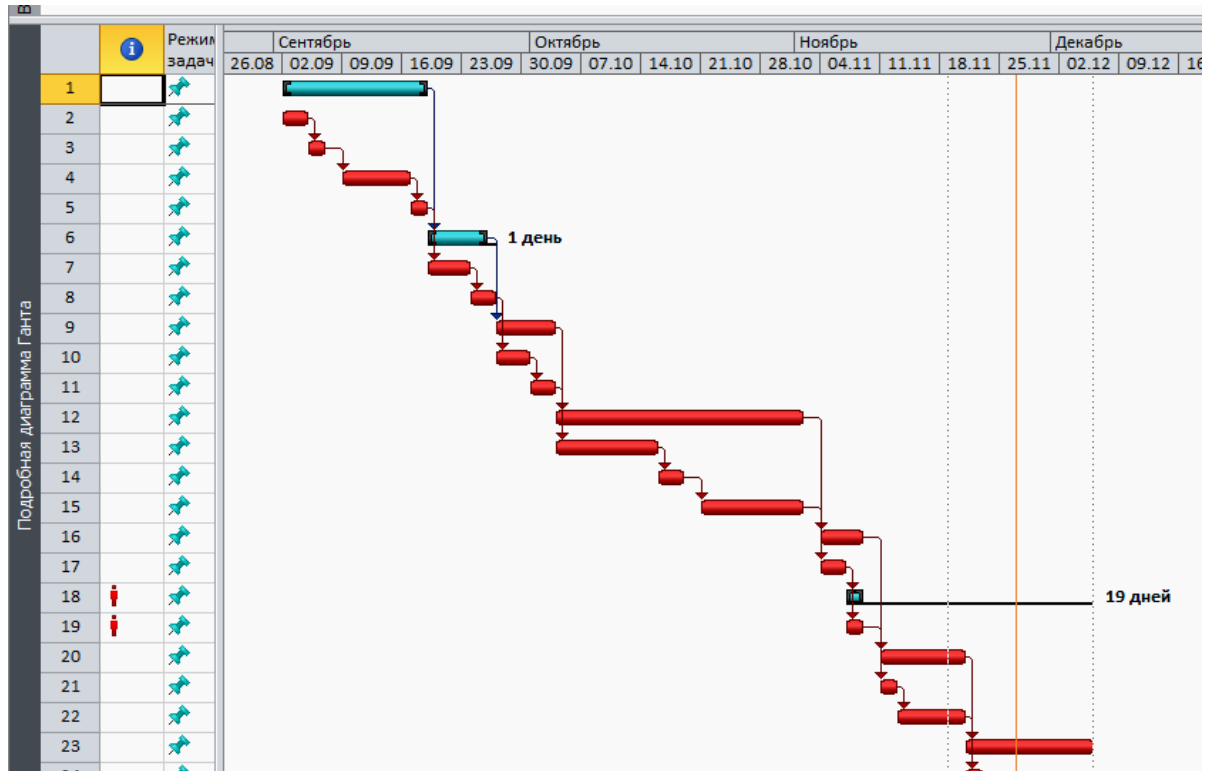


Рис. 3.7 –Докладна діаграма Ганта

Через застосування зав'язків послідовного типу між завданнями у наведеному плані майже всі стадії та етапи знаходяться на критичному шляху (виділені червоним кольором).

Після формування критичного шляху проекту був виконаний аналіз, результатом якого стала побудова діаграми Ганта (рис. 3.6).

Для усунення проблем, пов'язаних з критичним шляхом, були подовжені шляхи окремих етапів для стадій 3 та 4, а для стадій 6 та 7, навпаки – скорочені.

Після цього було перевірено ресурси (працівників) на предмет перевантаження після змін у плані (через наявність етапів з фіксованим обсягом робіт) та виконано перерозподіл навантаження.

3.4 Висновки до розділу

У підрозділі 3.1 проведено опис впровадження рекомендаційної системи до ІТ–продукту, рішень з ініціації ІТ–проекту. У результаті діяльності з ініціалізації проекту сформований статут проекту. На етапі ініціації проекту було виконано моделювання робіт проектної групи, розроблено концепцію майбутнього програмного продукту, заповнені таблиці задач, використання задач, ресурсів.

У підрозділі 3.2 виконано опис результатів планування ІТ–проекту. Приклади етапів планування наведені за допомогою діаграми Ганта.

У підрозділі 3.3 виконано аналіз критичного шляху проекту. Наведено докладну діаграму Ганта, яка візуалізує критичний шлях проекту.

4 ДОСЛІДЖЕННЯ ОСОБЛИВОСТЕЙ РЕАЛІЗАЦІЇ УСУНЕННЯ ПРОБЛЕМИ ХОЛОДНОГО СТАРТУ ПІД ЧАС ПОБУДОВИ РЕКОМЕНДАЦІЙНИХ СИСТЕМ

4.1 Опис особливостей реалізації рекомендаційної системи як самостійного ІТ–продукту

Під час розробки рекомендаційних систем зазвичай не використовують традиційні реляційні бази даних. Основна причина цього полягає у підвищених вимогах до швидкодії і великих обсягах даних, які необхідно обробляти.

Таким чином реляційні СКБД у свою чергу мало придатні для роботи з даними про активність користувача, які використовуються для формування поведінкового профілю у рекомендаційних системах. Основними причинами цього є той самий великий обсяг даних і підвищені вимоги до продуктивності (рекомендації повинні формуватися дуже швидко у процесі перегляду сторінок Інтернет–магазину відвідувачами).

Сьогодні у РС застосовують системи управління даними такі як NoSQL (HBase, Cassandra). Вони відрізняються високою швидкістю обробки запитів, відсутністю жорсткої схеми даних, а також відмовою від транзакцій, тобто мають практично лінійну масштабованість.

Слід відзначити наступні основні типи NoSQL БД: дані зберігаються у вигляді пари ключ/значення (key/value based); дані зберігаються у вигляді стовпців (column based); основним елементом зберігання даних є документ (document based); дані організовані у вигляді графа (graph based). За даними, наведеними у [26] на сьогоднішній день у десятці найбільш популярних СКБД знаходяться сім реляційних баз і три NoSQL СКБД (MongoDB, Cassandra, Redis). Наступний етап у розвитку систем зберігання даних – це

NoSQL (SAP Hana, MemSQL), які поєднали у собі переваги реляційних і NoSQL систем.

Наприклад, популярний сервіс, який надає доступ до перегляду різного відеоконтенту (телевізійні шоу, серіали, фільми та ін.), Hulu використовує систему управління даними HBase, а відомий Інтернет-магазин eBay – систему управління даними Cassandra (Data Stax Enterprise).

Системи управління даними HBase і Cassandra підтримують модель даних BigTable і мають такі основні характеристики як висока масштабованість, надійність, висока пропускна здатність (операції читання/запису), гнучка схема даних, реплікації, підтримка SQL подібної мови запитів до даних.

Завдяки використанню СКБД Cassandra рекомендаційна система Інтернет-магазину eBay має високий рівень продуктивності, виконуючи протягом однієї доби обробку близько 6 мільярдів операцій запису та понад 5 мільярдів операцій читання даних. РС eBay працює з графом, вузли якого – це користувачі (понад 200 мільйонів) і товари (понад 2 мільярд), а ребра – це зв'язки між ними (понад 40 мільярдів) [27].

Таким чином у атестаційній роботі теж була використана модель БД, яка дозволяє отримувати розширені відповіді на запити, використовуючи дані із зовнішніх джерел [20].

Результатом застосування даної моделі є підвищення ефективності використання обчислювальних ресурсів за рахунок забезпечення технології роботи з внутрішніми (дані про товари Інтернет-магазину) та зовнішніми (дані про відвідувачів) базами даних.

Звернення до БД, які обробляють, містять параметри для вибору даних із зовнішніх джерел даних, які у свою чергу містять ідентифіковані логічні об'єкти (наприклад, товари), для яких повинні бути отримані значення даних. При цьому визначають, що значення для першої підмножини ідентифікованих логічних об'єктів повинні бути доступними з першої бази

Система працює наступним чином. Обробник SQL приймає запити до БД, які містять різні параметри (параметри збагачень), наприклад ідентифікатор логічного об'єкту. Сервер збагачень здійснює доступ до даних логічного об'єкту з метою перевірки, що до кожного компоненту запиту є доступ з внутрішньої БД або за допомогою логічного об'єкту, який співпадає з ідентифікатором логічного об'єкту. Використовуючи механізм перетворень сервер збагачень отримує доступ до даних із зовнішніх джерел даних і таким чином відбирає показники зовнішніх компонентів логічного об'єкту. Ці показники можуть зберігатися у табличному вигляді всередині розширеної відповіді на звернення до БД, яка передається обробник SQL за допомогою сервера збагачень.

Обробник SQL може мати різні варіанти конфігурацій. В одному з варіантів він може заповнювати окремі частини внутрішньої БД розширеними відповідями на запити до бази.

Каталог містить інформацію, яка зберігається для кожного джерела даних, що відповідає компонентів логічного об'єкту. Механізм перетворень здійснює доступ до каталогу щоб “знати” як обмінятися інформацією з кожним зовнішнім джерелом. Наприклад у каталозі є записи для кожного зовнішнього джерела даних і кожна з них визначає правила обміну інформацією із зовнішнім джерелом (разом із викликами власних функцій, форматами даних, білінговими процедурами, обліковими даними тощо). Замість каталогу (або для його доповнення) може бути використаний будь-який механізм для визначення цієї інформації.

Сервер збагачень і механізм перетворень можуть зберігатися та експлуатуватися на локальній машині та/або у мережі для керування доступом до локальної/мереженої БД.

Так само сервер збагачень і механізм перетворень можуть входити до складу додатків хмарних обчислень (cloud computing), де локальну машину зазвичай використовують у якості інтерфейсу для спільного використання

обчислювальних ресурсів у центрі обробки даних віддаленого сервера. Протягом того часу, коли додатки хмарних обчислень зберігатимуть табличні дані у якісь внутрішні бази, ця база не зберігається на локальній машині.

Сервер збагачень може використовувати дані політики для встановлення відповідностей кожному з компонентів логічного об'єкта, який підходить одному із зовнішніх джерел даних. Наприклад, політики даних можуть містити уподобання, які асоціюються з відбором підходящого зовнішнього джерела даних. Якщо конкретизувати приклад, політики можуть означати уподобання відвідувачем (групою відвідувачів) сайту визначеного товару визначеного виробника перед іншими виробниками того самого товару. Таким чином політики можуть призначати пріоритети кожному з джерел даних. Сервер збагачень використає визначене джерело даних для розширення відповіді на запит до БД у випадку, коли компоненти логічного об'єкту не можуть бути отримані або обчислені.

Таким чином, сервер збагачень використовує механізм перетворень, щоб поєднати дані з внутрішньої БД з даними з одного або більше зовнішніх джерел даних. Оскільки внутрішня база даних може містити не загальнодоступні (тобто приватні) дані, сервер збагачень надає можливість прозорого поповнення приватних даних різноманітною загальнодоступною інформацією. Саме сервер збагачень видобуває параметри із запиту до БД, на основі яких ідентифікується логічний об'єкт.

Логічний об'єкт містить складові типів даних у вигляді компонентів, показники яких запитує користувач. Параметри так само містять показники, які використовують у якості критеріїв пошуку тощо, для відбору даних з зовнішніх джерел. Використовуючи ці параметри, механізм перетворень ініціює виклики відповідних власних функцій протоколу, щоб запитувати дані з зовнішніх джерел.

Під множиною зовнішніх джерел даних слід розуміти будь-які віддалені або Інтернет-джерела (наприклад, пошукові системи, інформаційні

канали та ін.), які можна використати для формування розширеної відповіді на запит до БД. Кожне з зовнішніх джерел реалізує визначений протокол, який дозволить іншим комп'ютерам здійснювати доступ до табличних даних. Визначений протокол містить прикладний програмний інтерфейс, у якому виклик окремих функцій призводить до видачі частини даних.

Зовнішні джерела даних можуть використовувати унікальні або запатентовані протоколи, наприклад, API для передачі даних, механізм перетворень здійснює доступ до конфігураційних даних, (наприклад, збережених у каталозі), щоб налаштувати виклик функції або віддалений запит, які сумісні з цим протоколом.

Виклик функції або віддалений запит містять критерії для ідентифікації інформації, щоб використовувати її для збагачення. Після того, як механізм перетворень ініціює виклик функції або запит, зовнішнє джерело даних повертає дані, які узгоджуються з цими критеріями. Дані систематизуються відповідно до власних форматів і механізм перетворень конвертує їх у форму сумісних табличних даних.

Механізм перетворень використовує контекстні параметри для ідентифікації даних, які відповідають показникам розташування та/або часу, з зовнішнього джерела даних. Наприклад, механізм перетворень використовує специфічні дату/час як параметр виклику власної функції для вибору вартість товару у поточну дату/час (за умови, що вартість залежить, наприклад, від коливання курсу валют). У іншому випадку механізм перетворень може ініціювати виклик іншій своїй функції для обчислення знижки на товар за певних умов (наприклад, за умови покупки декількох одиниць товару).

Вхідні параметри для власних функцій частково ідентифікують табличні дані, які повернуться. Наприклад, механізм перетворення викликає функцію з вхідними параметрами, які вказують назву товару та тип даних за

запитом, такий як вартість зі знижкою для відвідувача, що має книжкову картку.

Звернення до обробника SQL містить один або більше параметрів, які задають один або більше критеріїв для відбору даних із зовнішніх джерел. Один параметр містить ідентифікатор логічного об'єкта (наприклад, назву товару) для групи типів даних. Кожен тип даних може містити один або більше інших типів даних. Інші параметри містять один або більше показників для одного або більше типів даних з групи цих типів.

Інші окремі параметри можуть бути контекстними, такими що обмежують показники, що повертає зовнішнє джерело даних, відповідно до визначеного часу та/або місця розташування. Розширені відповіді на звернення до БД містять табличні дані, тому група типів даних займає один або більше стовпців, тоді як один або більше показників займають один або більше рядків. Так дані одного стовпця усередині кожного рядка відповідають одному або більше показників.

Запит до БД містить параметри для приватних даних відвідувача, наприклад, інформацією про географічне розташування, яке може бути у вигляді географічних координат або розташування Інтернет-ресурсу (IP адреси). Механізм перетворень може ініціювати виклик власної функції для даних, які входять до певного діапазону або місця розташування відповідно до GPS. Виклик власної функції може додатково запитувати дані, які стосуються логічного об'єкту, протягом визначеного проміжку часу, такі як час перебування відвідувача на сайті.

Якщо один з компонентів логічного об'єкта, не доступний (або невідомий) з одного джерела даних, механізм перетворень може задіяти програмні засоби забезпечення зв'язку іншого зовнішнього джерела даних, для виконання іншої власної функції, щоб отримати доступ до невідомого компоненту логічного об'єкта. Таким чином, механізм перетворень замінює

велику кількість окремих постачальників даних для обміну інформацією з кожним типом зовнішнього джерела даних.

З урахуванням удосконаленого методу пропонується блочно-структурна модель рекомендаційної системи, яка дозволяє отримати додаткові дані із зовнішніх джерел, а також, використовуючи модифікований механізм перетворень, скорочує час та підвищує точність видачі рекомендацій (рис. 4.2).



Рис. 4.2 – Модифікований механізм перетворень

Модифікований механізм перетворень містить похідний метод побудови рекомендацій в умовах холодного старту з використанням темпоральних обмежень за вхідними даними та запропоновані етапи, які модифікують метод.

Результатом виконання запропонованих етапів фільтрації є виділення підмножини товарів, якими найбільш сильно цікавиться підмножина користувачів з аналогічними додатковими характеристиками, які повідомив

про себе новий користувач (наприклад, стать – чоловіча, вік – 41 рік, місце розташування – Харків).

Результати вибору представляються у вигляді тимчасового масиву даних. На вхід похідного методу подаються не всі дані з БД, а певний набір даних. Таким чином, модифікований механізм перетворень скорочує час та підвищує точність видачі рекомендацій.

Модифікований механізм перетворень формує темпоральні обмеження за допомогою шаблону часового ряду, який являє собою інтервали часу. Ці інтервали залежать від предметної області, яка обирається з каталогу. Наприклад, для задачі продажу продуктів харчування тривалість такого інтервалу може становити тиждень, для продажу одягу – один сезон, а для ювелірних виробів – рік.

Крім того, шаблон часового ряду може бути введений в базу даних під час налаштування рекомендаційної системи на предметну область конкретного Інтернет–магазину.

4.2 Опис послідовностей операцій під час роботи рекомендаційної системи

Модель, використана у атестаційні роботі є універсальною та припускає різні модифікації, альтернативні варіанти побудови, еквіваленти. Тому доцільно приділити увагу більш детальному опису варіанта використання моделі, застосованого у даній роботі.

Розглянутий спосіб розширення відповідей на звернення до БД, використаний для побудови РС, складається наступних етапів: визначення значень для першої підмножини ідентифікованих логічних об'єктів; визначення значень для другої підмножини ідентифікованих логічних

об'єктів; видобуток значень з першої БД; генерування запиту видобутку; ідентифікація явної команди збагачення; перетворення явної команди збагачення на власні запити; отримання значень для другої підмножини; генерування розширеної відповіді на звернення.

Детальний опис етапів наведений у вигляді схем послідовностей операцій на рис. 4.3– 4.5.

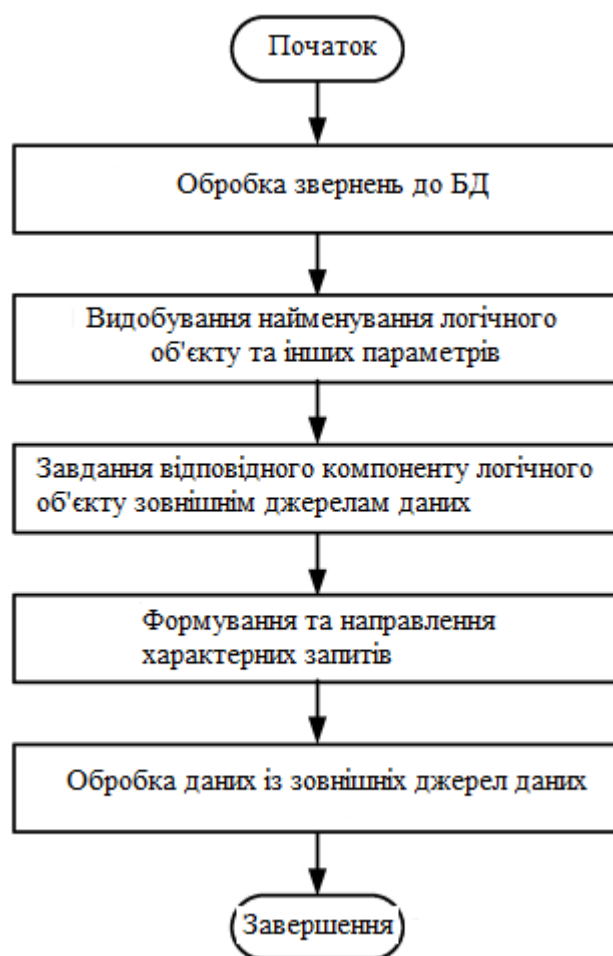


Рис. 4.3 – Схема послідовності операцій обробки звернень до БД

На рис. 4.3 зображена послідовність операцій, які потрібно виконати під час звернення до бази даних. Послідовність виконання відбувається наступним чином. Спочатку розпізнають параметри рішень всередині

звернення до БД: роблять витяг параметрів розширень зі звернення, які включають в себе ідентифікатор логічного об'єкта, і один або більше показників для одного або більше компонентів логічного об'єкта.

Компоненти логічного об'єкта містять типи даних, асоційовані з розширенням звернення до БД. Результати із зовнішніх джерел даних відповідають одному або більше показниками. Також необхідно завдати відповідність компонентів логічного об'єкта зовнішнім джерела даних. Дані політик визначають, яке зовнішнє джерело даних використовувати для отримання даних для конкретного компоненту логічного об'єкта.

Формування та направлення характерних запитів відбувається у механізмі перетворень, який виробляє і передає власні запити різноманітних даних на зовнішні джерела даних. Власні запити конфігурирують відповідно до протоколу для з'єднання з зовнішніми джерелами даних, для виконання сумісних команд і отримання результатів. Власні запити містять виклики власних функцій, які реалізують зовнішні джерела даних.

Після цього відбувається обробка даних, прийнятих від зовнішніх джерел даних, у якості відповіді на власні запити. Механізм перетворень конвертує дані, надані зовнішнім джерелом даних, у табличні дані, які використовуються, щоб згенерувати розширені відповіді. Сервер збагачень передає розширені відповіді на звернення до БД до обробника SQL, який повертає подання розширеної табличної відповіді та/або впроваджує табличні дані у внутрішню таблицю внутрішньої БД. Внутрішня таблиця може оновлюватися автоматично за планом, наприклад, за допомогою автоматичного генерування нового звернення на предмет зовнішніх даних і використання цих даних у розширених відповідях, щоб впроваджувати оновлені дані.

У необов'язковому порядку, сервер збагачень збирає статистичні дані, які стосуються пов'язаних з біллінгом дій, діагностики джерел даних, тощо. Ці статистичні дані можуть бути використані з різною метою, такою як

контроль витрат на білінг, аналіз джерел даних та оновлення політик. Наприклад, політики можуть містити обмеження використання для конкретного джерела даних. Якщо статистичні дані показують, що обмеження використання було досягнуто, може бути обране інше джерело даних відповідно до політик.

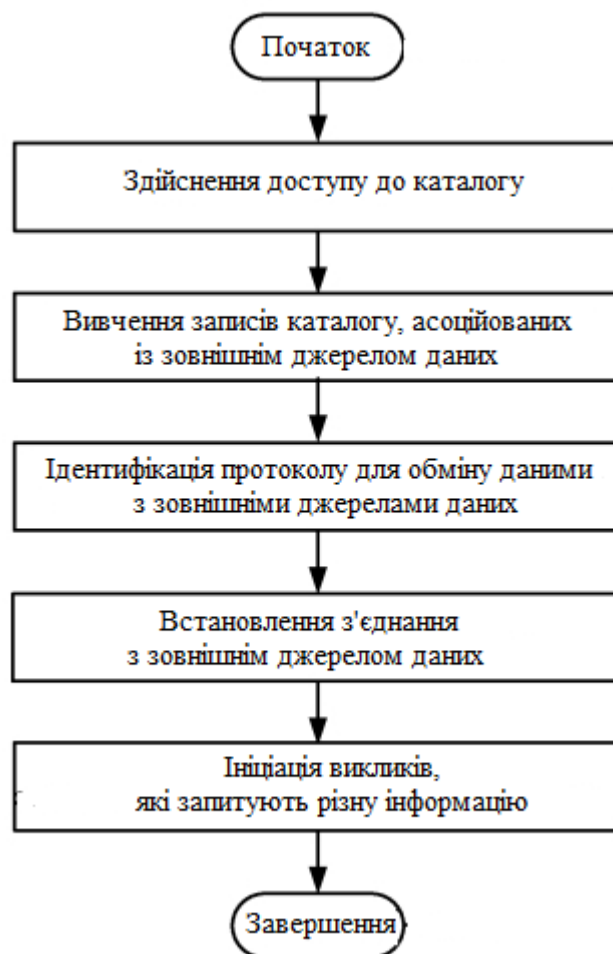


Рис. 4.4 – Схема послідовності операцій використання каталогу для запитів

На рис. 4.4 наведено схему послідовності операцій, яка ілюструє етапи для використання каталогу для передачі запитів різноманітної інформації під зовнішнє джерело даних. Ці запити конфігурирують так, щоб являти собою власну команду для зовнішнього джерела даних. Сервер збагачень

використовує механізм перетворень для створення цих запитів на основі звернення до бази даних, яке передається від обробника SQL.

Дослідження записів у каталозі, асоційованих з зовнішнім джерелом даних відбувається наступним чином. Ідентифікація протоколу для обміну інформацією з зовнішнім джерелом даних відбувається за рахунок перевірки як запис визначає протокол для обміну інформацією з зовнішнім джерелом, включаючи виклики власних функцій, формати даних, білінгові процедури, облікові дані безпеки, тощо. Отже, такий протокол використовує механізм перетворень для ініціації одного або більше викликів власних функцій до зовнішнього джерела даних.

Також необхідно підтримувати зв'язок з зовнішнім джерелом даних. Протокол записів каталогу містить розташування, таке як IP-адреса, та будь-які процедури аутентифікації для зовнішнього джерела даних. Наприклад, ім'я користувача, обліковий запис і пароль можуть бути затребувані під час доступу до зовнішнього джерела даних. Після встановлення сеансу (наприклад, сеансу HTTP або HTTPS), механізм перетворень відбирає одну або більше конкретних баз даних або таблиць і ініціює виклики, які запитують різноманітну інформацію. Ці виклики містять виклики власних функцій, асоційованих з API зовнішнього джерела даних. Після цього завершується використання каталогу, щоб передавати запити різноманітної інформації під зовнішнє джерело даних.

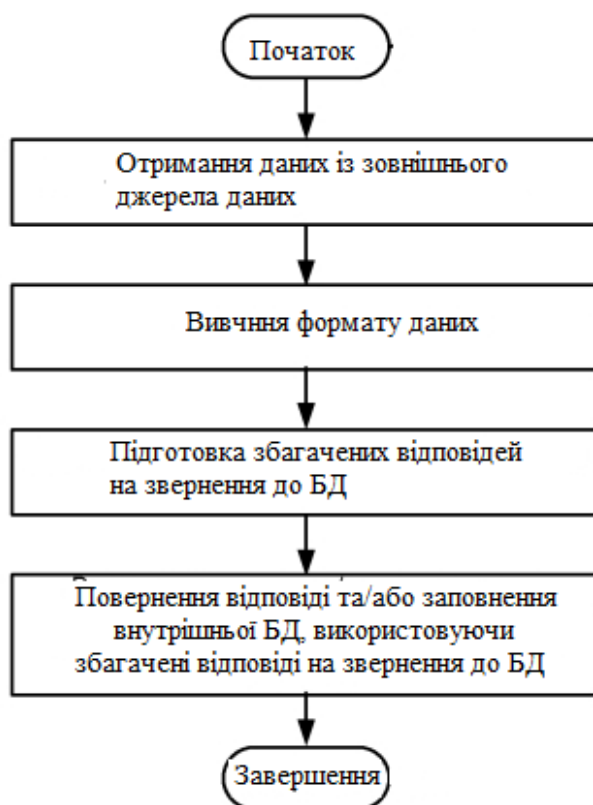


Рис. 4.5 – Схема послідовності операцій конвертації даних

На рис. 4.5 наведено схему послідовності операцій, яка ілюструє перетворення даних із зовнішніх джерел на дані у формі розширеної відповіді на звертання до бази даних. Механізм перетворень приймає дані від зовнішніх джерел, у відповідь на власні запити.

Відбувається вивчення формату, асоційованого з даними, прийнятими від зовнішніх джерел (хоча формат може бути відомим заздалегідь, наприклад, з інформації у каталозі). Визначене зовнішнє джерело даних повертає показники в особливому форматі, наприклад, такому як формат позначень об'єктів JavaScript (JSON). Механізм перетворень перетворює ці показники на табличні дані та готує розширені відповіді. Механізм перетворень може виконувати додаткову обробку цих показників.

Потім відбувається повернення подання розширеної таблиці та/або заповнення внутрішньої бази даних, використовуючи розширені відповіді на звернення до БД. За допомогою сервера збагачень, табличні дані, створені з

показників, які були повернуті конкретним зовнішнім джерелом даних, можуть бути впроваджені до таблиці або до внутрішньої бази даних. Зазначена таблиця є незаповненою або тимчасовою таблицею поза межами реляційної парадигми, асоційованої з внутрішньою базою даних.

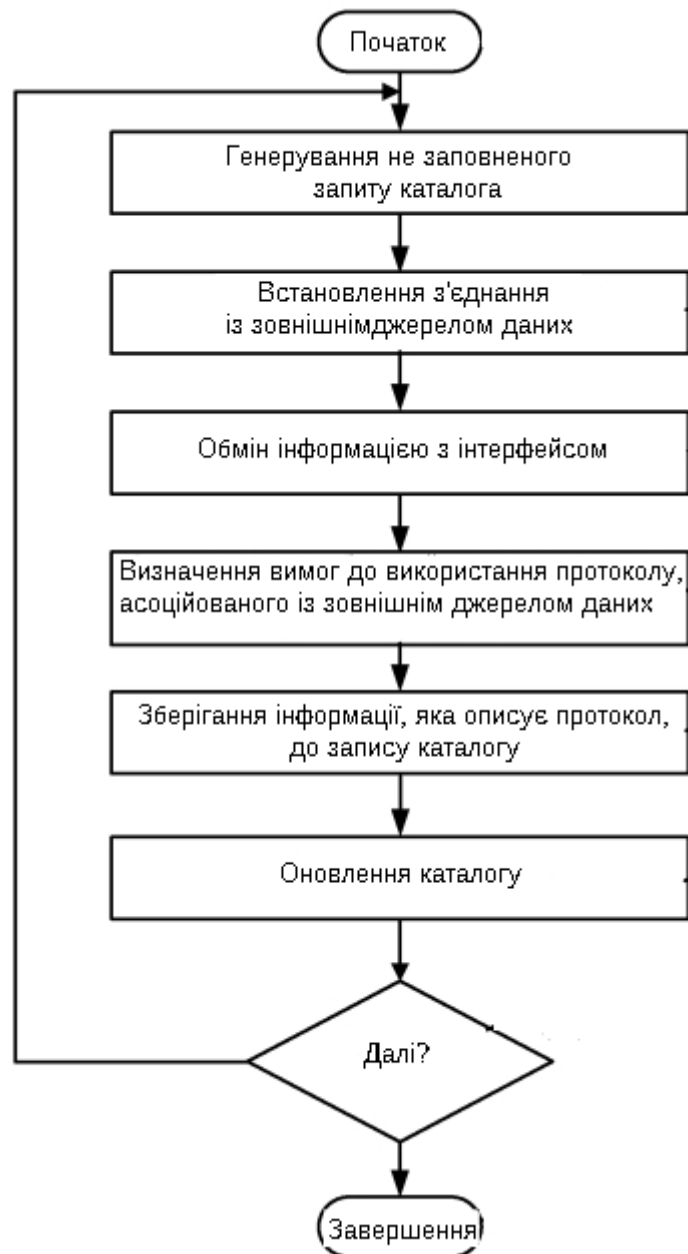


Рис. 4.6 Схема послідовності операцій створення каталогу

На рис. 4.6 наведено схему послідовності операцій, яка ілюструє створення каталогу, щоб перетворювати звернення до бази даних у власні запити до зовнішнього джерела даних. Каталог утворює частина механізму перетворень.

Механізм перетворень зберігає різноманітну інформацію до каталогу, наприклад, таку як протокол для обміну інформацією із зовнішнім джерелом даних. Протокол визначає виклики власних функцій, щоб пристрій запитував інформації та формати для систематизації інформації, яка запитується.

Також відбувається з'єднання із зовнішнім джерелом даних. Наприклад, механізм перетворень створює сеанс, який використовує IP-адреси, за допомогою інтерфейсу до зовнішнього джерела даних, такого як інтерфейс прикладного програмування (API). Механізм перетворень приступає до обміну інформацією з зовнішнім джерелом даних через цей інтерфейс. У окремих варіантах проваджень, механізм перетворень використовує інтерфейс для перерахування кожної наявної бази даних, і відбирає відповідну БД згідно з політикою.

Ще необхідно виконати визначення вимог для використання протоколу, асоційованого із зовнішнім джерелом даних. Механізм перетворень ідентифікує функції сконфігуровані для повернення різноманітної інформації, які виконує інтерфейс. Ці функції є власними для зовнішнього джерела даних, і надають можливість доступу до однієї або більше наявних БД. Механізм перетворень також визначає, які облікові дані необхідні для пошуку в одній або більше БД. Ще механізм перетворень можна використати для визначення використання унікального або запатентованого формату для обміну даними. Будь-яка інформація, що стосується вимог використання протоколу, зберігається у каталозі. Цей запис використовують, щоб оновлювати каталог.

Після попередніх дій виконують прийняття рішення щодо того, чи оновлювати додатково каталог іншим записом для наступного зовнішнього

джерела даних. Якщо є ще зовнішні джерела даних для вивчення, попередні дії повторюють, якщо немає – завершують створення каталогу. Каталог може бути перебудований та/або оновлений у будь-який час, наприклад, під час додавання нового джерела даних або коли він змінює свій протокол.

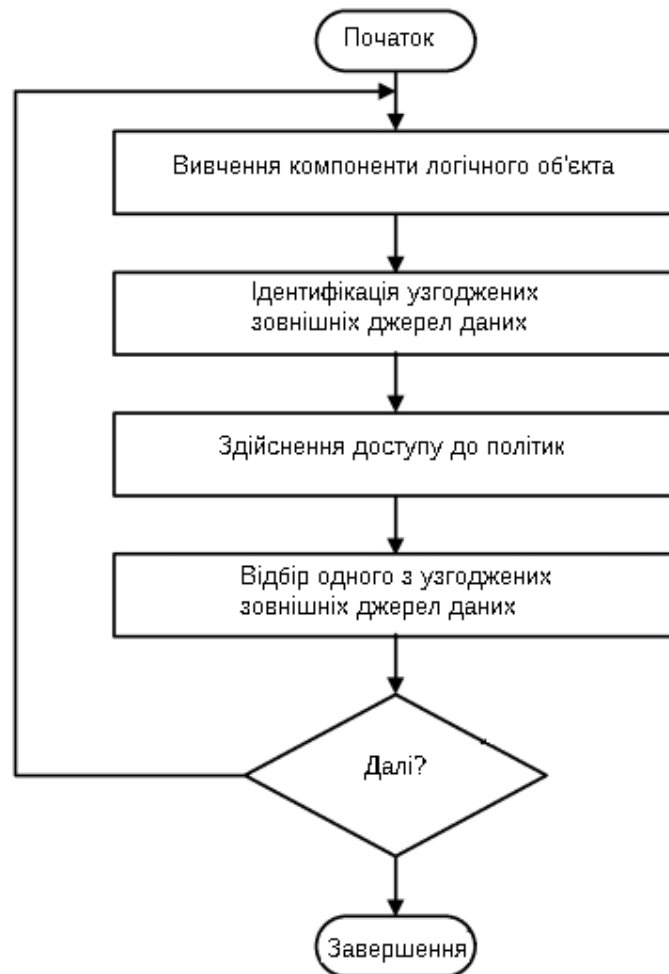


Рис. 4.7 – Схема послідовності операцій

На рис. 4.7 наведено схему послідовності операцій, яка ілюструє завдання відповідності компонентів логічного об'єкта зовнішніх джерел даних до одного ілюстративного варіанту проваджень. Кожний компонент логічного об'єкта містить тип даних, показник або показники якого

розширюють відповідь на звернення до БД. Звернення до бази даних задає логічний об'єкт, який є компоновкою типів даних.

Узгоджені зовнішні джерела даних конфігуруються таким чином, щоб надавати інформацію, яка відповідає компонентів логічного об'єкта. Наприклад, узгоджені зовнішні джерела даних надають дані, які мають той самий тип даних, що і компонент логічного об'єкта. Дані, які надаються, можуть потребувати додаткової обробки. Зазвичай, такі дані містять показники, які алгоритм використовує в якості вхідних параметрів для обчислення іншого показника, який зберігається у компоненті логічного об'єкта.

Після ідентифікації узгоджених зовнішніх джерел відбувається доступ до політик, щоб відібрати відповідне джерело. Політики містять переваги для конкретних зовнішніх джерел даних, засновані на різних факторах, таких як вимоги по білінгу, характеристики роботи, безкоштовні джерела даних тощо. У такому випадку, тільки безкоштовні джерела з узгоджених зовнішніх джерел даних вважаються придатними для вибору. Якщо більше ніж одне узгоджене зовнішнє джерело даних є безкоштовним, то будь-яке з них може бути відібраним.

4.3 Опис практичної апробації методу та тестування результатів

Розглянемо алгоритм колаборативної фільтрації за схожістю користувачів, яка визначається за допомогою косинусної міри. Фрагменти коду, наведені у підрозділі, написані мовою python.

Вхідні дані

Підмножина користувачів U та підмножина товарів I отримані шляхом фільтрації даних за трьома критеріями (місце проживання, стать, діапазон

вікових груп) з використанням перших двох етапів удосконаленого методу побудови рекомендацій в умовах холодного старту з використанням темпоральних обмежень.

У якості вхідних даних виступає матриця оцінок (табл. 4.1), виставлених користувачами товарів, для зручності товарам привласнені номери 1–9:

Таблиця 4.1 – Матриця оцінок

	1	2	3	4	5	6	7	8	9
U1	5.000 0	3.000 0			4.000 0				
U2	4.000 0					1.000 0		2.000 0	3.000 0
U3		5.000 0	5.000 0						
U4			4.000 0	3.000 0		2.000 0	1.000 0		

Задати її можна за допомогою csv-файлу, в якому першим стовпцем буде код користувача, другим – ідентифікатор товару, третім – виставлена користувачем оцінка. Таким чином, потрібно сформувати csv-файл з наступним змістом.


```
U1,1,5.0  
U1,2,3.0  
U1,5,4.0  
U2,1,4.0  
U2,6,1.0  
U2,8,2.0  
U2,9,3.0  
U3,2,5.0  
U3,3,5.0  
U4,3,4.0  
U4,4,3.0  
U4,6,2.0  
U4,7,1.0
```

Рис. 4.8 – Фрагмент csv-файлу, аркуш 1

Для початку необхідно розробити функцію, яка прочитає csv-файл. Для зберігання рекомендацій доцільно використовувати стандартну для python структуру даних dict: кожному користувачеві ставиться у відповідність довідник його оцінок виду “продукт”:”оцінка”. Вийде наступний код.

```

import csv
def ReadFile (filename = "<csv_file_location>"):
    f = open (filename)
    r = csv.reader (f)
    mentions = dict()
    for line in r:
        user      = line[0]
        product   = line[1]
        rate      = float(line[2])
        if not user in mentions:
            mentions[user] = dict()
        mentions[user][product] = rate
    f.close()
    return mentions

```

Рис. 4.9 – Фрагмент структури даних, аркуш 1

Міра схожості.

Інтуїтивно зрозуміло, що для рекомендації користувачеві U1 будь-якого продукту, обирати потрібно з товарів, які подобаються якимось користувачам 2, 3, 4, тощо, оцінки яких найбільш схожі з оцінками користувача U1. Щоб отримати чисельне вираження цієї “схожості” користувачів можна скористатися порівнюванням векторів. Наприклад, для M товарів оцінки, виставлені окремо взятим користувачем, є вектором у M-вимірному просторі цих товарів. Серед можливих засобів вимірювання можна відзначити наступні:

1. Косинусна міра.
2. Коефіцієнт кореляції Пірсона.
3. Евклідова відстань.
4. Коефіцієнт Танімото.
5. Манхеттенська відстань та ін.

Косинусна міра та коефіцієнт кореляції Танімото – це найбільш часто використані у сучасних рекомендаційних системах засоби вимірювання. Розглянемо більш детально косинусну міру, яку і буде реалізовано. Косинусна міра для двох векторів – це косинус кута між ними, у свою чергу косинус кута між двома векторами – це їх скалярний добуток, поділений на довжину кожного з двох векторів:

$$\cos(\vec{x} \bullet \vec{y}) = \frac{\vec{x} \bullet \vec{y}}{\|\vec{x}\|_2 \|\vec{y}\|_2} \quad (4.1)$$

Реалізуємо обчислення косинусної міри, з огляду на те, що набір оцінок користувача наведено у вигляді структури dict “продукт”:”оцінка”.

```
def distCosine (vecA, vecB):
    def dotProduct (vecA, vecB):
        d = 0.0
        for dim in vecA:
            if dim in vecB:
                d += vecA[dim]*vecB[dim]
        return d
    return dotProduct (vecA,vecB) / math.sqrt(dotProduct(vecA,vecA)) / math.sqrt(dotProduct(
vecB,vecB))
```

Рис. 4.10 – Фрагмент коду обчислення косинусної міри, аркуш 1

Під час реалізації був використаний факт, що скалярний добуток вектора самого на себе дає квадрат довжини вектора. З точки зору

продуктивності це не найкраще рішення, але у розробленій рекомендаційній системі швидкість роботи досягається за рахунок інших засобів.

Алгоритм колаборативної фільтрації.

На основі наявної матриці переваг користувачів можна визначати, наскільки два користувачі схожі один на одного. Після цього необхідно реалізувати алгоритм колаборативної фільтрації, виконуючи наступні кроки.

1. Обрати L користувачів, вподобання яких найбільше схожі на вподобання розглянутого користувача $U1$. З цією метою для кожного з користувачів потрібно обчислити обрану косинусну міру стосовно розглянутого користувача, та обрати L найбільших. Для користувача $U1$ з таблиці 4.1 отримаємо наступні значення.

Таблиця 4.2 – Косинусна міра користувачів

	U2	U3	U4	Σ
U1	0.516	0.000	0.066	0.583
	4	0	7	1

2. Для кожного з користувачів помножити його оцінки на обчислену величину міри, таким чином оцінки більш “схожих” користувачів будуть сильніше впливати на підсумкову позицію товару (табл. 4.3).

3. Для кожного з продуктів обчислити суму каліброваних оцінок L найбільш близьких користувачів, отриману суму розділити на суму заходів L обраних користувачів.

Таблиця 4.3 – Вплив оцінок на підсумкову позицію товару

	1	2	3	4	5	6	7	8	9
U2	2.582	1.549	0.000	0.000	2.065	0.000	0.000	0.000	0.000
	0	2	0	0	6	0	0	0	0
U3	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	0	0	0	0	0	0	0	0	0
U4	0.000	0.000	0.266	0.200	0.000	0.133	0.066	0.000	0.000
	0	0	8	1	0	4	7	0	0
sum	2.582	1.549	0.266	0.200	2.065	0.133	0.066	0.000	0.000
	0	2	8	1	6	4	7	0	0
result	4.428	2.656	0.457	0.343	3.542	0.228	0.114	0.000	0.000
	1	8	6	2	4	8	4	0	0

Сірим кольором позначені стовпчики продуктів, які вже були оцінені обраним користувачем і вдруге пропонувати їх йому не має сенсу.

У вигляді формули цей алгоритм може бути наведений як

$$r_{u,i} = k \sum_{u' \in U} \text{sim}(u, u') r_{u',i} \quad (4.2)$$

де функція sim – обрана міра схожості двох користувачів, U – безліч користувачів, r – виставлена оцінка, k – нормувальний коефіцієнт:

$$k = 1 / \sum_{u' \in U} |\text{sim}(u, u')| \quad (4.3)$$

Код, який реалізує всі кроки, наведений на рис. 4.10.

```

import math
def makeRecommendation(userID, userRates, nBestUsers, nBestProducts):
    matches = [(u, distCosine(userRates[userID], userRates[u])) for u in userRates if u <> userID]
    bestMatches = sorted(matches, key=lambda(x,y):(y,x), reverse=True)[:nBestUsers]
    print "Most correlated with '%s' users:" % userID
    for line in bestMatches:
        print "  UserID: %6s  Coeff: %6.4f" % (line[0], line[1])
    sim = dict()
    sim_all = sum([x[1] for x in bestMatches])
    bestMatches = dict([x for x in bestMatches if x[1] > 0.0])
    for relatedUser in bestMatches:
        for product in userRates[relatedUser]:
            if not product in userRates[userID]:
                if not product in sim:
                    sim[product] = 0.0
                sim[product] += userRates[relatedUser][product] * bestMatches[relatedUser]
    for product in sim:
        sim[product] /= sim_all
    bestProducts = sorted(sim.iteritems(), key=lambda(x,y):(y,x), reverse=True)[:nBestProducts]
    print "Most correlated products:"
    for prodInfo in bestProducts:
        print "  ProductID: %6s  CorrelationCoeff: %6.4f" % (prodInfo[0], prodInfo[1])
    return [(x[0], x[1]) for x in bestProducts]

```

Рис. 4.11 Фрагмент коду, аркуш 1

Результат перевірки працездатності коду, отриманий під час тестування, виглядає наступним чином (рис. 4.11).

```

Most correlated with 'ivan' users:
  UserID:  alex  Coeff: 0.5164
  UserID:  david Coeff: 0.0667
  UserID:   bob  Coeff: 0.0000
Most correlated products:
  ProductID: 5  CorrelationCoeff: 3.5426
  ProductID: 2  CorrelationCoeff: 2.6570
  ProductID: 3  CorrelationCoeff: 0.4574
  ProductID: 4  CorrelationCoeff: 0.3430
  ProductID: 7  CorrelationCoeff: 0.1143

```

Рис. 4.12 Фрагмент результатів перевірки, аркуш 1

Подальший розвиток цієї розробки можна вести у таких напрямках як оптимізація структур даних, оптимізація продуктивності, підбір міри схожості, модифікація алгоритму фільтрації.

Після реалізації проекту з розробки рекомендаційної системи та застосування її на сайті Інтернет–магазину відбулося порівняльне тестування. Тестувалися рекомендаційна система Інтернет–магазину та ново розроблена.

У якості методу тестування було використано А/В–тест (англ. A/B testing, Split testing), який є методом маркетингового дослідження [28].

Суть даного методу наступна: контрольну групу елементів порівнюють з набором тестових груп, у яких попередньо змінюють один або кілька показників. Це робиться з метою з'ясувати, які зі змін покращують цільовий показник. Різновидом А/В–тестування є багатоваріантне тестування. У такому випадку тестують не два цілісних варіанти, а відразу кілька елементів продукту або складових частин піддослідного об'єкту у різних поєднаннях, при яких кожен тестований елемент може бути двох видів (А або В). Перевагою методу є використання під час проектування об'єктивних даних [29].

Принцип проведення А/В–тесту наступний: всі відвідувачі сайту випадковим чином діляться на дві рівні групи, і кожна група бачить однакову версію сайту. Змінюються тільки блоки персональних рекомендацій – кожній групі показуються блоки, керовані однією з рекомендаційних систем (рис. 4.12).

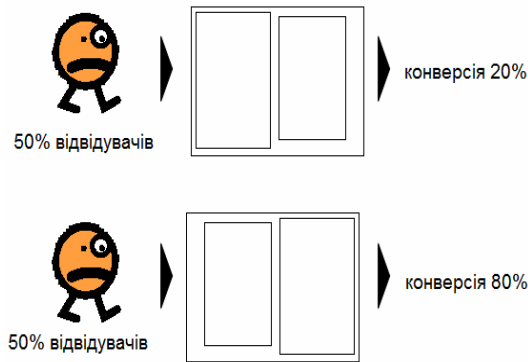


Рис. 4.13 – Ілюстрація А/Б-тесту

Під час тестування вимірюється конверсія кожного сегмента трафіку та порівнюється з іншими. Рішення про ефективність РС приймається за результатами.

Під конверсією слід розуміти дії відвідувача на сайті (покупка, замовлення, реєстрація та ін.) [30].

Відвідувачів на групи поділяють за допомогою JavaScript-коду, всі користувачі отримують ідентифікатор одного з двох сегментів тесту, який зберігається у Cookie та передається до Google Analytics під час кожної значимої дії на сайті.

Під Cookie (англ. Cookie, буквально – печиво) – невеликий фрагмент даних, відправлений Web-сервером, який зберігається на комп'ютері користувача. Web-браузер під час кожної спроби відкрити сторінку відповідного сайту пересилає цей фрагмент даних Web-сервера у складі HTTP-запиту [31].

Результати тесту отримані з Google Analytics – конверсія за сегментами (табл 4.4):

1. Сегмент А – рекомендаційна система Інтернет-магазину.
2. Сегмент В – розроблена рекомендаційна система.

Таблиця 4.4 – Результати конверсії.

	03.12.2019	04.12.2019	05.12.2019
Сегмент А	–6,15%	+8,98%	–2,78%
Сегмент В	+3,58%	+10,24%	+2,96%

Відповідно до результатів зміни конверсії щодо показників означають, що рекомендаційна система Інтернет–магазину програє розробленій рекомендаційній системі.

Одним зі способів оцінки ефективності та якості рекомендацій можна вважати звичайну візуальну оцінку. Наприклад, рекомендації сформовані обома розглянутими системами для товару “Материнська плата MSI MAG Z390 Tomahawk (s1151, Intel Z390, PCI–Ex16)” наведені на рис. 4.12 та рис. 4.13.

До материнської плати розроблена РС рекомендує корпус, оперативну пам'ять та термопасту (рис. 4.12), а РС Інтернет–магазину разом з процесором та оперативною пам'яттю рекомендує предмет одягу та іграшку.

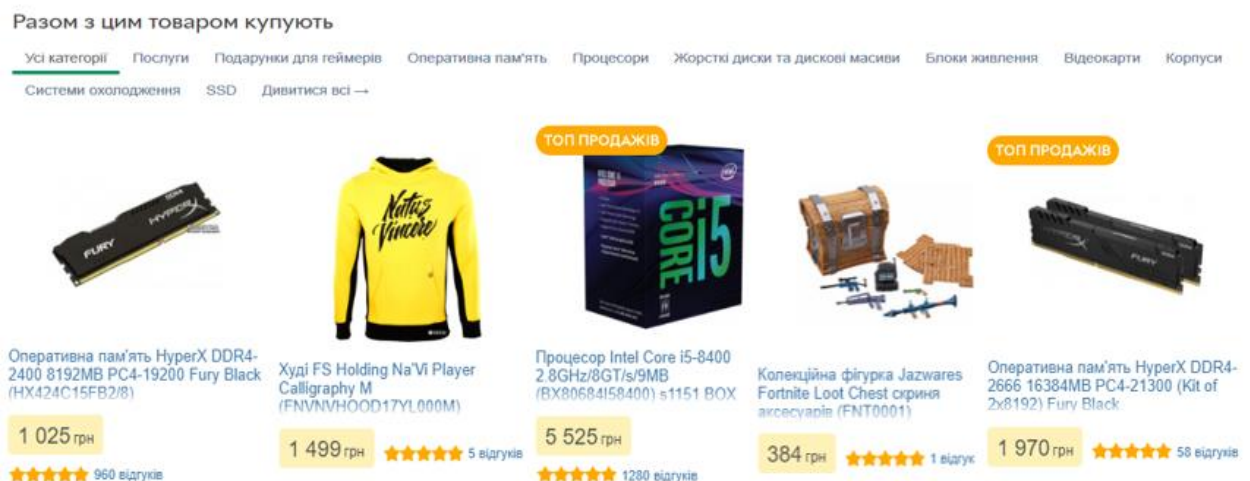


Рис. 4.14 – Рекомендації системи Інтернет–магазину

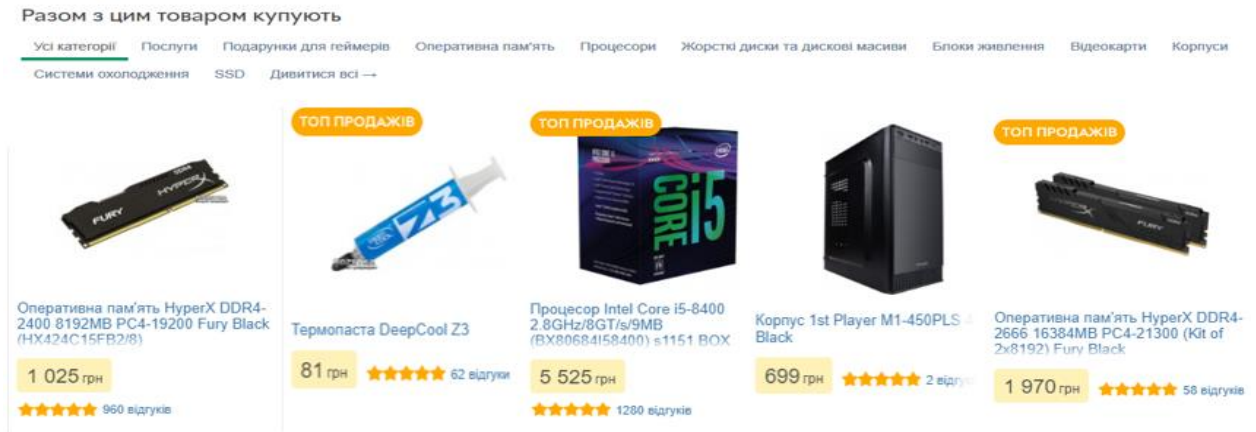


Рис. 4.15 – Розроблена рекомендаційна система

Щодо інших товарів, які часто відвідують (за таким товарами швидко накопичується статистика), результати приблизно такі самі. Слід зауважити, що візуальна експертна оцінка не впливає безпосередньо на цифри, це простий і швидкий спосіб, який використовують як показник якості роботи РС.

Для оцінки ефективної роботи розробленої РС була застосована, також непряма оцінка якості рекомендацій. З цією метою було досліджено аудиторію відвідувачів, яка взаємодіє з блоками товарних рекомендацій, за допомогою Google Analytics.

Відповідно до побудованих сегментів кліків (натискань на клавіші миші) на рекомендаційні блоки користувачів блоки розробленої системи отримали більше кліків, ніж блоки рекомендацій РС Інтернет-магазину.

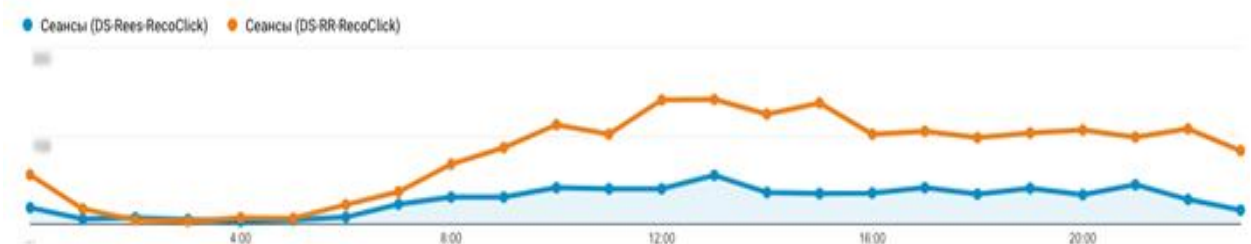


Рис. 4.16 – Результати з Google Analytics

З аналізу видно, що розроблена РС частіше рекомендує релевантні відвідувачеві товари, відвідувачі частіше клікають на ці товари, а рекомендації позитивно впливають на продажі.

Окремо було проведене дослідження аудиторії інтернет-магазину.

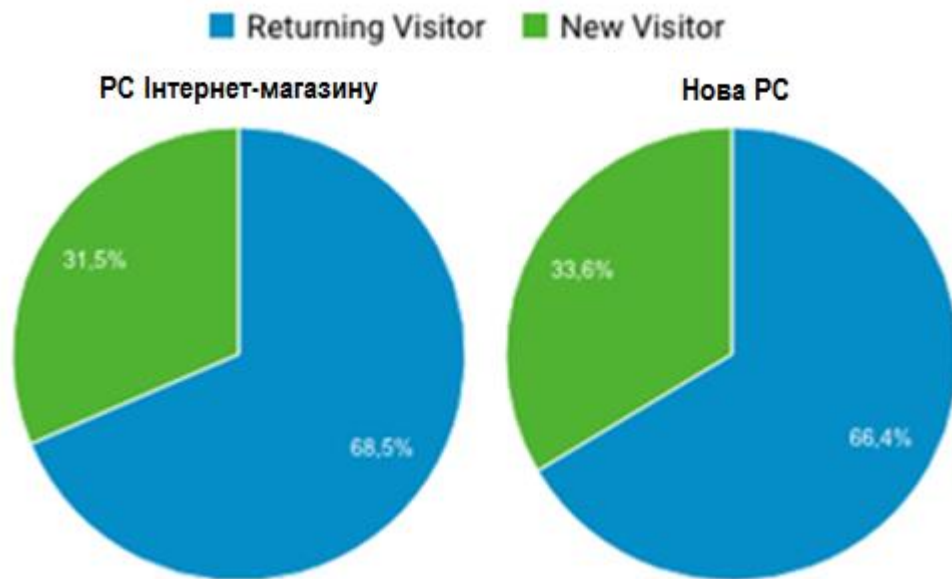


Рис. 4.17 – Розподіл аудиторії на нових і постійних відвідувачів.

З метою перевірки коректності роботи поділу трафіку інтернет-магазину на сегменти, сегментатор був протестований за допомогою коду, який використовує сайт: паралельно з основним поділом, запустили сегментування тієї ж аудиторії – похибка вийшла мінімальною:

Сегмент 2: 63500 користувачів

Сегмент 3: 63686 користувачів

Це означає, що сегментатор працює правильно і похибки немає.

4.4 Висновки до розділу

У підрозділі 4.1 проведено дослідження особливостей реалізації усунення проблеми холодного старту під час побудови рекомендаційної системи. Зроблено опис особливостей реалізації РС як самостійного ІТ-продукту.

З урахуванням удосконаленого методу пропонується блочно-структурна модель рекомендаційної системи, яка дозволяє отримати додаткові дані із зовнішніх джерел, а також, використовуючи модифікований механізм перетворень скорочує час та підвищує точність видачі рекомендацій. Розглянута модель дозволяє отримувати розширені відповіді на запити, використовуючи дані із зовнішніх джерел. Значну увагу було приділено схемам роботи окремих частин моделі як складових рекомендаційної системи та перевагам які надає сама модель під час роботи РС в умовах холодного старту.

У підрозділі 4.2 виконано опис послідовностей операцій під час роботи рекомендаційної системи на основі моделі системи, яка дозволяє отримувати розширені відповіді на запити, використовуючи дані із зовнішніх джерел.

У підрозділі 4.3 виконано опис практичної апробації удосконаленого методу та тестування результатів. У якості вхідних даних виступає матриця оцінок, виставлених користувачами товарів. Підмножина користувачів, отримана шляхом фільтрації даних з використанням перших трьох етапів удосконаленого методу.

Апробація розробленої рекомендаційної системи виконана за допомогою різних методів тестування: візуального, непрямого та засобами Google Analytics. Наведені результати тестування та проведений

порівняльний аналіз. Результати порівняння показують, що розроблена рекомендаційна система має кращі результати ніж РС Інтернет–магазину.

ВИСНОВКИ

У даній атестаційній роботі магістра виконано наступні роботи у межах дослідження обраної теми.

Розглянуті загальні питання щодо рекомендаційних систем і визначене коло проблем, пов'язаних з їх розробкою та експлуатацією. Проведено аналіз методів побудови рекомендаційних систем із зазначенням їх переваг та недоліків. Досліджені проблеми холодного старту та розглянуті підходи до її вирішення. Визначена проблема магістерської атестаційної роботи та підходи до розв'язання цієї проблеми. Зроблено постановку задачі магістерської роботи.

Виконано дослідження особливостей побудови рекомендаційних систем, детально розглянута архітектура рекомендаційної системи Інтернет-магазину.

Проведен аналіз методів вирішення проблеми холодного старту. Окремо проаналізований метод побудови рекомендацій в умовах холодного старту з використанням темпоральних обмежень за вхідними даними. Метод з використанням темпоральних обмежень доповнює набір вхідних даних перед виконанням колаборативної фільтрації, яка використовується для побудови рекомендацій.

Виконано удосконалення методу побудови рекомендацій в умовах холодного старту з використанням темпоральних обмежень за рахунок введення фільтрації даних на підставі додаткової інформації про користувача. Запропонований метод дозволяє підвищити ефективність побудови рекомендацій в умовах холодного старту на основі скорочення часу обробки відфільтрованого набору даних.

Проведено дослідження особливостей реалізації усунення проблеми холодного старту під час побудови рекомендаційної системи. Зроблено опис

особливостей реалізації РС як самостійного ІТ–продукту. З урахуванням удосконаленого методу пропонується блочно–структурна модель рекомендаційної системи, яка дозволяє отримати додаткові дані із зовнішніх джерел, а також, використовуючи модифікований механізм перетворень скорочує час та підвищує точність видачі рекомендацій.

Розглянута модель дозволяє отримувати розширені відповіді на запити, використовуючи дані із зовнішніх джерел. Значну увагу було приділено схемам роботи окремих частин моделі як складових рекомендаційної системи та перевагам які надає сама модель під час роботи РС в умовах холодного старту.

Виконано опис послідовностей операцій під час роботи рекомендаційної системи на основі моделі системи, яка дозволяє отримувати розширені відповіді на запити, використовуючи дані із зовнішніх джерел.

Виконано опис практичної апробації удосконаленого методу та тестування результатів. Апробація розробленої рекомендаційної системи виконана за допомогою різних методів тестування: візуального, непрямого та засобами Google Analytics. Наведені та результати тестування та проведений порівняльний аналіз.

Результати порівняння показують, що розроблена рекомендаційна система має кращі результати ніж РС Інтернет–магазину.

Проведено дослідження особливостей реалізації усунення проблеми холодного старту під час побудови рекомендаційної системи. Зроблено опис особливостей реалізації РС як самостійного ІТ–продукту. З урахуванням удосконаленого методу пропонується блочно–структурна модель рекомендаційної системи, яка дозволяє отримати додаткові дані із зовнішніх джерел, а також, використовуючи модифікований механізм перетворень, скорочує час та підвищує точність видачі рекомендацій.

Значну увагу було приділено схемам роботи окремих частин моделі як складових рекомендаційної системи та перевагам які надає сама модель під час роботи РС в умовах холодного старту.

Проведено опис впровадження рекомендаційної системи до ІТ-продукту, рішень з ініціації ІТ-проекту. У результаті діяльності з ініціалізації проекту сформований статут проекту.

Виконано опис результатів планування ІТ-проекту. Приклади етапів планування наведені за допомогою діаграми Ганта. Проведен аналіз критичного шляху проекту. Наведено докладну діаграму Ганта, яка візуалізує критичний шлях проекту.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Методичні вказівки щодо розробки та оформлення магістерської атестаційної роботи за спеціальністю 122 Комп'ютерні науки (освітня програма “Управління проектами в галузі інформаційних технологій” освітньо-кваліфікаційного рівня “магістр” / Упоряд.: Петров К.Е., Левикін В.М., Чалий С.Ф., Євланов М.В., Саєнко В.І., Міхнов Д.К., Міхнова А.В., Чала О.В. – Харків: ХНУРЕ, 2019. – 24 с.
2. ДСТУ 3008:2015. Інформація та документація. Звіти у сфері науки і техніки. Структура і правила оформлювання. . – Чинний від 22.06.2015. – Київ: ДП “УкрНДНЦ”, 2016. – 31 с.
3. Почему персонализация контента это еще не веб-персонализация // Стаття в сети Интернет, URL: <http://lpgenerator.ru/blog/2016/03/19/pochemupersonalizaciya-kontenta-eto-eshe-ne-vebpersonalizaciya/> (дата звернення: 10.11.2019)
4. Amazon product data / The University of California. URL: <http://jmcauley.ucsd.edu/data/amazon/> (date of access: 10.11.2019).
5. MacKenzie I., Meyer C., Noble S. How retailers can keep up with consumers // McKinsey & Company. URL: <https://www.mckinsey.com/industries/retail/our-insights/howretailers-can-keep-up-with-consumers> (date of access: 11.11.2019)
6. M. Tim Jones Introduction to approaches and algorithms URL: https://www.ibm.com/developerworks/opensource/library/os-recommender1/index.html?S_TACT=105AGX99&S_CMP=CP (date of access: 1.11.2019)
7. Ricci F., Rokach L., Shapira B. Recommender Systems Handbook. New York, 2015.

8. P. Resnick and H. R. Varian, “Recommender systems,” *Communications of the ACM*, vol. 40, no. 3, pp. 56–58, 1997. View at Google Scholar · View at Scopus
9. Kula M. Metadata Embeddings for User and Item Cold-start Recommendations // Eprint archive. URL: <https://arxiv.org/pdf/1507.08439.pdf> (date of access: 10.11.2019).
10. Lucas Bernardi, Jaap Kamps, Julia Kiseleva, Melanie J.I. Mueller The Continuous Cold Start Problem in e-Commerce Recommender Systems, 2015
11. L. Tang, Y. Jiang, L. Li, and T. Li. Ensemble contextual bandits for personalized recommendation. In *Proceedings of the 8th ACM Conference on Recommender Systems*, pages 73–80, 2014.
12. J. Z. Sun, K. R. Varshney, and K. Subbian. Dynamic matrix factorization: A state space approach. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1897–1900, 2012.
13. M. Tavakol and U. Brefeld. Factored mdps for detecting topics of user sessions. In *Proceedings of the 8th ACM Conference on Recommender Systems*, pages 33–40, 2014
14. G. Adomavicius and A. Tuzhilin, “Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734–749, 2005. View at Publisher · View at Google Scholar · View at Scopus
15. K. Yu, A. Schwaighofer, V. Tresp, X. Xu, and H.-P. Kriegel, “Probabilistic memory-based collaborative filtering,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 1, pp. 56–69, 2004. View at Publisher · View at Google Scholar · View at Scopus
16. D. Billsus and M. Pazzani, “Learning collaborative information filters”, in *Proceedings of the 15th International Conference on Machine Learning (ICML '98)*, 1998.

17. T. Landauer, M. Littman, and Bell Communications Research (Bellcore), “Computerized cross-language document retrieval using latent semantic indexing,” US patent no. 5301109, April 1994.

18. S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, “Indexing by latent semantic analysis,” *Journal of the American Society for Information Science*, vol. 41, no. 6, pp. 391–407, 1990. View at Google Scholar

19. K. Pearson, “On lines and planes of closest fit to systems of points in space,” *Philosophical Magazine*, vol. 2, pp. 559–572, 1901. View at Google Scholar

20. Расширение ответов на обращения к базе данных, используя данные из внешних источников данных: пат. RU 2611967US:Владельцы патента:МАЙКРОСОФТ ТЕКНОЛОДЖИ ЛАЙСЕНСИНГ, ЭлЭлСи (US)

21. X. Su and T. M. Khoshgoftaar, “Collaborative filtering for multi-class data using belief nets algorithms,” in *Proceedings of the International Conference on Tools with Artificial Intelligence (ICTAI '06)*, pp. 497–504, 2006. View at Publisher · View at Google Scholar · View at Scopus

22. R. Greinemr, X. Su, B. Shen, and W. Zhou, “Structural extension to logistic regression: discriminative parameter learning of belief net classifiers,” *Machine Learning*, vol. 59, no. 3, pp. 297–322, 2005. View at Publisher · View at Google Scholar · View at Scopus

23. Z. Huang, H. Chen, and D. Zeng, “Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering,” *ACM Transactions on Information Systems*, vol. 22, no. 1, pp. 116–142, 2004. View at Publisher · View at Google Scholar · View at Scopus

24. G. Linden, B. Smith, and J. York, “Amazon.com recommendations: item-to-item collaborative filtering,” *IEEE Internet Computing*, vol. 7, no. 1, pp. 76–80, 2003. View at Google Scholar

25. B. M. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Incremental SVD-based algorithms for highly scaleable recommender systems," in Proceedings of the 5th International Conference on Computer and Information Technology (ICCIT '02), 2002.
26. Ranking. URL: <http://db-engines.com/en/ranking>(дата звернення: 11.11.2019)
27. Jonathan Gottfried Graph Based Recommendation Systems at eBay // Стаття у мережі Інтернет, URL: <http://www.slideshare.net/planetcassandra/e-baynys>, 2013.
28. А/В-тестування. URL: <https://uk.wikipedia.org/wiki/A/B-тестування>(дата звернення: 11.11.2019)
29. Lukas Mathis. A/B Testing // Designed for Use. — Pragmatic Bookshelf, 2011. — 344 p
30. Конверсія (в інтернет-маркетингу). URL: [https://uk.wikipedia.org/wiki/Конверсія_\(в_інтернет-маркетингу\)](https://uk.wikipedia.org/wiki/Конверсія_(в_інтернет-маркетингу)) (дата звернення: 11.11.2019)
31. Куки. URL: <https://uk.wikipedia.org/wiki/Куки>(дата звернення: 11.11.2019)
32. PMBOK® Guide and Standards. URL: <https://www.pmi.org/pmbok-guide-standards> (date of access: 10.11.2019).
33. ДСТУ 8302:2015. Інформація та документація. Бібліографічні посилання. Загальні положення та правила складання. — Чинний від 04.03.2016. — Київ: ДП "УкрНДНЦ", 2016. — 20 с.
34. Chalyi S. Доповнення вхідних даних рекомендаційної системи в ситуації циклічного холодного старту з використанням темпоральних обмежень типу "next" / S. Chalyi, V. Leshchynskyi, I. Leshchynska // Системи управління, навігації та зв'язку. Збірник наукових праць. — Полтава: ПНТУ, 2019. — Т. 4 (56). — С. 105-109. — doi:<https://doi.org/10.26906/SUNZ.2019.4.105>.

35. Chalyi, V. Leshchynskyi, I. Leshchynska. Method of forming recommendations using temporal constraints in a situation of cyclic cold start of the recommender system. EUREKA: Physics and Engineering Vol. 4. P. 34-40.

36. Levykin V., Chala O. (2018). Development of a method of probabilistic inference of sequences of business process activities to support business process management. Eastern-European Journal of Enterprise Technologies, 5/3(95), 16-24. DOI: 10.15587/1729-4061.2018.142664.

37. Leshchynskyi V., Leshchynska I. The use of active learning in a situation of a cyclical cold start of the recommender system // Вісник Національного технічного університету "ХПІ". Серія: Системний аналіз, управління та інформаційні технології — Харків : НТУ "ХПІ", 2018. — № 44 (1320) 2018). — С. 66-72.

38. Levykin V., Chala O. (2018). Method of determining weights of temporal rules in markov logic network for building knowledge base in information control system. EUREKA: Physics and Engineering, 5(18), 3-10.

39. Braunhofer M. (2014). Hybrid Solution of the Cold-Start Problem in Context-Aware Recommender Systems. UMAP 2014: User Modeling, Adaptation, and Personalization, 484-489.