

Хмельницький національний університет Факультет інформаційних технологій
Кафедра інженерії програмного забезпечення

ДИПЛОМНА РОБОТА МАГІСТРА

Метод створення рекомендаційних систем

Рівень вищої освіти Другий (магістерський) _____

Галузь знань _____ 12 - Інформаційні технології _____

Спеціальність _____ 121 - Інженерія програмного забезпечення _____

Освітня програма _____ Освітньо-професійна програма інженерія програмного
забезпечення

ДРІПЗ 2001112.20.02.05 ПЗ

Виконав: студент 2_курсу, група ІПЗм-20-1



підпис
підпис
підпис

А.І. Коваль
Ініціали прізвище

Керівник к.т.н, доцент _____

О.М. Яшина
Ініціали прізвище

Нормоконтроль к.т.н, доцент _____

О.М. Яшина
Ініціали прізвище

До захисту допускаю:

Зав. кафедри ІПЗ д-р фіз.-мат. наук, проф.



Л.П. Бедратюк

_____ 20 р.

Хмельницький 2021р.

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій
Кафедра Інженерії програмного забезпечення
Рівень вищої освіти Другий (магістерський)
Галузь знань 12 «Інформаційні технології»
Спеціальність 121 «Інженерія програмного забезпечення»
Освітня програма Освітньо-професійна програма «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Л. П. Бедратюк

01 09 2021 р.

**ЗАВДАННЯ
НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ)**

Коваль Андрій Ігорович

Прізвище, ім'я, по батькові студента

1. Тема проєкту (роботи) Метод створення рекомендаційних систем

Керівник проєкту (роботи) Яшина Оксана Миколаївна

к.т.н., доцент

Затверджена наказом ректора університету від 25.08.2021 р. № 102

2. Строк подання студентом проєкту (роботи) на кафедру 01.12.2021 р.

3. Вихідні дані до проєкту (роботи) Матеріали переддипломної практики

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)

1 Дослідження предметної області та постановка задачі

2 Концепції, моделі та методи вирішення задачі

3 Алгоритми та технології вирішення задачі

4 Реалізація та тестування програмної системи

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень)

Презентаційні матеріали (слайди)

6. Консультанти розділів дипломного проєкту (роботи)

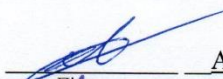
Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 01 вересня 2021 р.

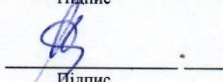
КАЛЕНДАРНИЙ ПЛАН

Назва етапів (розділів) дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1 Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження; визначення структури дипломної роботи	01.09 – 09.09.2021	
2 Робота над розділом 1 дипломної роботи – вивчення літературних джерел; аналіз відомих моделей, методів та засобів за темою роботи; висновки до розділу та постановка задачі	10.09 – 25.09.2021	
3 Робота над розділом 2 дипломної роботи – розробка моделей та методів вирішення поставленої задачі; висновки до розділу	26.09 – 10.10.2021	
4 Робота над науковими публікаціями	11.10 – 20.10.2021	
5 Робота над розділом 3 дипломної роботи – розробка алгоритмів та технологій, проектування для вирішення поставленої задачі; висновки до розділу	11.10 – 26.10.2021	
6 Робота над розділом 4 дипломної роботи – програмна реалізація спроектованих рішень, результати експериментів, їх аналіз; висновки до розділу	27.10 – 15.11.2021	
7 Узгодження постановки задачі, отриманих результатів та висновків; написання вступу, загальних висновків, оформлення джерел посилання та додатків; оформлення пояснювальної записки та графічних матеріалів згідно вимог стандартів	16.11 – 30.11.2021	
8 Попередній захист дипломної роботи	Листопад (згідно графіка)	
9. Перевірка роботи на наявність плагіату; нормоконтроль; брошурування пояснювальної записки; підготовка супровідних документів	01.12 – 04.12.2021	
10 Підготовка до захисту дипломної роботи	05.12 – 08.12.2021	

Студент


 Підпис А.І. Коваль
 Ініціали, прізвище

Керівник проєкту (роботи)


 Підпис О.М. Яшина
 Ініціали, прізвище

РЕФЕРАТ

Тема кваліфікаційної роботи: «Метод створення рекомендаційних систем».

Автор проекту: Коваль Андрій Ігорович.

Керівник роботи: Яшина Оксана Миколаївна.

Пояснювальна записка: 79 с., 5 рис., 2 дод., 85 джерел.

РЕКОМЕНДАЦІЙНІ СИСТЕМИ, ШТУЧНІ НЕЙРОННІ МЕРЕЖІ,
КЛАСИФІКАТОР, СИСТЕМИ ФІЛЬТРАЦІЇ.

Об'єктом дослідження є процеси пошуку товарів та послуг у мережі Інтернет з використанням рекомендаційних систем.

Предметом дослідження є методи створення рекомендаційних систем.

Мета роботи – розробка методу створення рекомендаційних систем для досягнення ефективності взаємодії користувача щодо пропонованих товарів та послуг в мережі інтернет.

У кваліфікаційній роботі проаналізовані методи створення рекомендаційних систем з використанням різних стратегій щодо надання релевантного пошуку на запит.

Розроблено метод створення рекомендаційних систем, який дає змогу досягти ефективності у взаємодії користувача щодо пропонованих товарів та послуг в мережі інтернет з використанням сучасних інформаційних технологій та методів розробки програмних продуктів. На відміну від відомих методів, враховує поєднання стратегій релевантного пошуку на запит та фільтрації даних в динамічному режимі.

ABSTRACT

Theme of qualification work: «Method of creating recommendation systems».

Author of the project: Koval Andriy Ihorovych.

Project manager: Yashina Oksana Mykolayivna.

Explanatory note: 79 pp., 5 figs., 2 appendice, 85 sources.

RECOMMENDATION SYSTEMS, ARTIFICIAL NEURAL NETWORKS, CLASSIFIER, FILTRATION SYSTEMS.

The object of the research is search processes for buyers on the Internet and sellers of services and goods using referral systems. The subject of research is methods of creating recommendation systems.

The aim of the work is to develop a method of creating recommendation systems to achieve the effectiveness of user interaction on the proposed goods and services on the Internet using modern information technology and software development methods.

The qualification paper analyzes the methods of creating recommendation systems using different strategies for providing relevant search on request.

The developed method of creating referral systems, which allows to achieve efficiency in user interaction on the proposed goods and services on the Internet using modern information technology and software development methods, in contrast to known methods, takes into account a combination of relevant search query and data filtering in dynamic mode.

ЗМІСТ

ВСТУП	8
1 РЕКОМЕНДАЦІЙНІ СИСТЕМИ	10
1.1 Основні поняття про рекомендаційні системи	10
1.2 Стратегії досягнення результату в рекомендаційних системах	15
1.3 Методи фільтрації рекомендацій для користувачів	19
1.4 Постановка задачі дослідження.....	30
1.5 Висновки до першого розділу	31
2 МЕТОДИ СТВОРЕННЯ РЕКОМЕНДАЦІЙНИХ СИСТЕМ З КОНТЕКСТНОЮ ОБІЗНАНІСТЮ ЗГІДНО ЖУРНАЛУ ВІДГУКІВ КОРИСТУВАЧІВ	32
2.1 Декларативна система рекомендацій з контекстної обізнаності	32
2.2 Контекстно-обізнана рекомендаційна система з декларативним користувачем та профілюванням елементів	41
2.3 Висновки до другого розділу.....	47
3 ІНТЕРАКТИВНА КОНТЕКСТНА РЕКОМЕНДАЦІЙНА СИСТЕМА	49
3.1 Вимоги до створення інтерактивної рекомендаційної системи.....	49
3.2 Метод створення інтерактивних рекомендаційних систем	54
3.3 Висновки до третього розділу	61
4 ДВОНАПРАВЛЕНА ПОСЛІДОВНА РЕКОМЕНДАЦІЙНА СИСТЕМА З КОНТЕКСТНОЇ ОБІЗНАНОСТІ ДАНИХ	62
4.1 Архітектура та метод створення двонаправленої послідовної рекомендаційної системи.....	62
4.2 Розробка, реалізація та оцінка прототипу рекомендаційної системи.....	67
4.3 Висновки до четвертого розділу	76
ВИСНОВКИ	77

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	80
ДОДАТОК А НАУКОВА СТАТТЯ.....	90
ДОДАТОК Б ПРЕЗЕНТАЦІЙНІ МАТЕРІАЛИ	95

ВСТУП

Рекомендаційна система – підклас системи фільтрації інформації, яка буде рейтинговий перелік об'єктів (фільми, музика, книги, новини, вебсайти), яким користувач може надати перевагу. Для цього використовується інформація з профілю користувача.

Існують дві основні стратегії створення рекомендаційних систем: фільтрація вмісту і колаборативна фільтрація. При фільтрації вмісту створюються профілі користувачів і об'єктів. Профілі користувачів можуть містити демографічну інформацію або відповіді на певний набір питань. Профілі об'єктів можуть містити назви жанрів, імена акторів, імена виконавців, тощо. Або якусь іншу інформацію в залежності від типу об'єкта. Цей підхід застосований у проєкті Music Genome Project: музичний аналітик оцінює кожну композицію за сотнями різних музичних характеристик, які можна використати для виявлення музичних уподобань користувача.

При колаборативній фільтрації використовується інформація про поведінку користувачів у минулому – наприклад, інформація про придбання або оцінки. В цьому разі не має значення, з якими типами об'єктів ведеться робота, але при цьому можна брати до уваги неявні характеристики, які складно було б врахувати при створенні профілю. Основна проблема цього типу рекомендаційних систем — «холодний старт»: відсутність даних про користувачів чи об'єкти, які нещодавно з'явилися у системі.

Потребують удосконалення методи створення рекомендаційних систем для покращення надання пропозицій користувач інтернет-сервісів.

Мета роботи – розробка методу створення рекомендаційних систем для досягнення ефективності взаємодії користувача щодо пропонованих товарів та послуг в мережі інтернет.

Для досягнення мети дослідження поставлено наступні завдання:

- здійснити аналіз відомих рішень з розробки рекомендаційних систем;
- здійснити моделювання системи та результати експериментів з моделлю;

- розробити метод та алгоритми створення рекомендаційних систем;
- здійснити реалізацію запропонованого рішення.

Об'єктом дослідження є процеси пошуку товарів та послуг у мережі Інтернет з використанням рекомендаційних систем.

Предметом дослідження є методи створення рекомендаційних систем.

У кваліфікаційній роботі проаналізовані методи створення рекомендаційних систем з використанням різних стратегій щодо надання релевантного пошуку на запит.

Під час виконання завдань дослідження були застосовані:

- методи обробки даних;
- методи класифікації.

Наукова новизна роботи: розроблений метод створення рекомендаційних систем, що дає змогу досягти ефективності у взаємодії користувача щодо пропонуванних товарів та послуг в мережі інтернет з використанням сучасних інформаційних технологій та методів розробки програмних продуктів. На відміну від відомих методів, враховує поєднання стратегій релевантного пошуку на запит та фільтрації даних в динамічному режимі.

За результатами кваліфікаційної роботи опубліковано наукову статтю у фаховому науковому виданні «Вісник Хмельницького національного університету»:

Коваль А.І., Яшина О.М., Радельчук Г.І. Порівняння об'єктно-орієнтованої та функційної парадигми в проектуванні програмного забезпечення. / Вісник ХНУ, серія Технічні науки, №3, 2021, с. 34-38.

1 РЕКОМЕНДАЦІЙНІ СИСТЕМИ

1.1 Основні поняття про рекомендаційні системи

Асортимент товарів у фізичних магазинах обмежений можливостями торговельних площ. Кількість товарів, що пропонуються, залежить від розміру магазину. У більшості випадків виникає потреба відвідувати декілька фізичних магазинів, щоб придбати те, що потребується. Цифрові технології дозволили споживачам отримати доступ до широкого спектру продуктів через Інтернет. Зі збільшенням розвитку мобільних пристроїв та онлайн-сервісів, онлайн-діяльність стала важливою частиною повсякденного життя людей. Користувачі отримують доступ до інтернет-сайтів для покупок, бронювання квитків та розважальних заходів. Незліченна кількість предметів доступна на інтернет-сайтах, залучаючи величезну кількість соціальних користувачів. За інформацією з повідомлень Amazon, близько 5 мільярдів людей в 2021 році наявні активними користувачами магазину з Інтернету, а Amazon Marketplace пропонує понад 353 мільйонів продуктів в Інтернеті, включаючи одяг, взуття, ювелірні вироби та предмети домашнього вжитку [1]. І ріст користувачів та асортимент товарів тільки будуть продовжувати швидко зростати [2]. За допомогою онлайн-покупок споживачі можуть порівнювати ціни і купувати товари у сотень різних продавців, усуваючи необхідність відвідати кілька фізичних магазинів. Але при цьому виникає нова проблема щодо спроможності інтернет - покупців знайти продукти, які відповідають їх потребам.

У реальному світі споживач підбирає товар на основі ряду характеристик та факторів. Наприклад, з врахуванням бренду та репутації магазину. Тим не менш, покупець, також, придбає товар з нового магазину, який пропонує якісний товар за нижчими цінами. Інші користувачі можуть вести себе так само або зовсім по-іншому. На кожну поведінку впливає важливість того, що індивідуальний споживач надає певному фактору, і жоден з двох інтернет – покупців не йде точно таким же шляхом до здійснення покупки. Таким чином, для того, щоб успішно

розвиватися на ринку, підприємства повинні спочатку зрозуміти поведінку онлайн-користувачів, а потім створити персоналізовані рекомендації для них. Рекомендації, що підготовлені для особи, можуть бути виконані за допомогою рекомендаційних систем.

Рекомендаційна система – це онлайн-система пошуку інформації, яка допомагає користувачам, відкриваючи та рекомендуючи відповідні елементи, які узгоджуються з уподобаннями користувача [3, 4].

Більшість компаній впроваджують рекомендаційні системи для ефективної персоналізації досвіду покупок. Їх стратегії варіюються від купівельної поведінки аналогічних користувачів, визначення поточних контекстів, а потім вивчення їх минулих транзакцій. Застосування рекомендаційної системи відрізняються для різних доменів і широко прийняті багатьма комерційними сайтами. Amazon, Netflix і YouTube. Серед інших відомих фірм, використовують підхід до рекомендації, оскільки він пропонує величезні переваги як для бізнес-провайдерів, так і для користувачів. Для бізнесу основними перевагами є підвищення лояльності клієнтів і доходів від бізнесу [5, 6]. Дослідження підтверджують, що на рекомендаційну систему припадало 35% доходу Amazon і 75% глядачів Netflix [7]. Для користувачів переваги включають більшу ефективність у пошуку найбільш релевантних пунктів, пошук нових товарів [6, 8], а також поліпшення процесу та якості прийняття рішень [9]. Більше 50% користувачів завдяки рекомендаційним системам висловили зацікавленість у придбанні та готовності платити більше за товари або послуги в деяких категоріях [10].

Для того, щоб поліпшити якість рекомендацій, системи рекомендацій повинні краще розуміти переваги користувачів, включаючи контекстне формування та персоналізацію їх для кожного користувача. Цей тип системи рекомендацій став відомий як рекомендаційна система з контекстною інформацією (РСКИ), метою якої є рекомендувати список елементів, які мають відношення до конкретної контекстної ситуації або неявні порядки послідовностей, кліків тощо. В [11] запропоновано загальні підходи до включення змісту та контекстів у рекомендації. Контексти можуть бути

використані трьома різними способами: для розділення даних перед рекомендацією; для фільтрування неконтекстуальних рекомендацій, які мають відношення до контексту користувача; для інтегрування контексту з процесом рекомендації. Рекомендації, що усвідомлюють контекст, повинні мати справу з кількома питаннями та проблемами, включаючи велику кількість контекстів та упередженість користувачів до цих контекстів; різне значення контексту для кожного користувача; динаміка контекстної інформації, яка може вплинути на подальшу поведінку користувача; великий обсяг даних; часте оновлення даних.

Розглянемо зміст поняття кількох контекстів. Велика кількість контекстів онлайн-медіа та залучення різних, але незрозумілих контекстів, коли користувач вирішує щось, роблять ідентифікацію персоналізованих уподобань користувачів більш складною. Елемент може мати сотні контекстів, і користувач може включати різні контексти для кожного прийняття рішень, які вона прийняла. Крім того, користувачі можуть використовувати різні контексти при виборі онлайн-елемента. Однак не всі контексти корисні. Надлишкові контексти введуть до додаткових витрат на обчислення, одночасно завищуючи якість рекомендацій. Попередні дослідження мали вбудовані різні контексти в рекомендаційні системи для підвищення ефективності, такі як поведінкові категорії [12], популярні статті [13], соціальна довіра [14], взаємодія з користувачами та діяльність [15-25]. Однак ці дослідження поділяють основний недолік – використання заздалегідь визначених контекстів у моделюючому середовищі даних, які не можуть всебічно охопити всі можливі тексти і можуть вплинути на переваги користувачів. Наприклад, рекомендаційна система, яка враховує тільки популярність товару при його рекомендації, не буде достатньою при необхідності для споживача зацікавленого тільки в покупці останнього продукту.

Персоналізована контекстна важливість враховує уподобання користувача. Крім різних контекстів, які розглядаються користувачами, користувач може також надавати більш високий пріоритет певному контексту більше, ніж іншим. Наприклад, споживач продуктів харчування оцінює ресторан на основі смаку їжі та

різноманітності меню, незалежно від ціни. З огляду на цей сценарій, факторами, які вплинуть на рекомендацію, в порядку важливості, є смак, різноманітність пунктів в меню. Такі фактори повинні бути прийняті до уваги; якщо ні, то якість рекомендації постраждає. Персоналізація користувацьких контекстів проводилася по-різному, одним з яких є створення підпрофілів користувачів для кожного контексту [22], підпрофілі для кожної групи користувачів [23] або моделей різних взаємодій між контекстами та рейтингами [26]. Однак всі ці вищезгадані підходи вибирають однаковий набір контекстів для всіх користувачів, і пріоритети користувача не добре змодельовані, що не може охоплювати реальні ситуації, як у наведеному вище сценарії.

Динамічна контекстна інформація впливає на поведінку користувача. У реальному світі онлайн-діяльність користувачів, як правило, постійно ведеться послідовно, а не є кількома незалежними подіями. Таким чином, ідеальний підхід до контекстно-свідомої послідовної рекомендації повинен бути чутливим, відповідати послідовним уподобанням користувачів і вміти відображати послідовність, зміни уподобань під час рекомендації наступної дії в режимі реального часу. Багато рішень, які підходять для контекстно-обізнаних рекомендаційних систем не можна застосувати до сценарію послідовної рекомендації, оскільки зміни поведінки користувачів не можуть бути визнані в послідовних справах. В роботах [27-33] запропонували контекстно-обізнані послідовні рекомендації, врахувавши контексти в послідовних рекомендаціях, таких як день тижня та година дня для транзакції, тобто брали до уваги обліковий запис тимчасового контексту, такого як часовий розрив між двома операціями. Однак основна увага цих робіт полягає в тому, щоб зафіксувати динамічні послідовні значення статичних контекстів, які повинні були бути визначені заздалегідь. Таким чином, вони не підходять для ситуацій, коли важливі контексти можуть динамічно змінюватися.

Розглянемо зміст впливу масштабних даних. Пошук відповідних рекомендацій для цільового користувача шляхом оцінки всіх доступних елементів неефективний, якщо є велика кількість елементів. Отже, пункти кандидатів повинні

бути визначені для того, щоб рекомендувати і оцінювати їх відповідно. Чисті системи на основі вмісту оцінюють всі доступні елементи, щоб знайти ті, які найкраще узгоджуються з профілем користувача. З іншої сторони, системи спільної фільтрації на основі пам'яті використовують взаємодію з користувачами та переваги інших користувачів для отримання предметів, що представляють інтерес. Ці методи мають недоліки через великі матриці, оскільки кількість взаємодії між користувачем і пунктом може бути занадто низькою для створення високоякісних кластерів. З огляду на ці обмеження, необхідно вирішити дві основні проблеми, щоб мати більш ефективну систему рекомендацій. По-перше, система повинна мати можливість фільтрувати відповідних кандидатів на предмет і обробляти дуже рідкісні дані. Методи спільної фільтрації на основі моделі вирішують цю проблему, створюючи рекомендації з використанням машинного навчання та методів інтелектуального аналізу даних, які можуть обробляти більшу кількість елементів для більшої кількості користувачів. Однак більшість підходів, що використовують цю техніку, не помічають впливу контекстів, щоб краще зрозуміти переваги користувачів.

Часті оновлення даних теж впливають на остаточний результат. Мільйони нових продуктів оновлюються щодня на ринку. Наприклад, в 2017 році в Amazon було додано 1,3 мільйона продуктів [3]. Таким чином, для того, щоб зберегти продуктивність рекомендації, модель повинна бути актуальною і адекватною кожного разу, коли надходять нові дані. Однак перепідготовка моделі буде дорогою з точки зору часу. Різні методи рекомендації були запропоновані для контексту-обізнаних рекомендаційних поправок. Тим не менш, існуючі моделі часто ігнорують різне значення контекстів для кожного користувача та елемента, застосовуючи аналогічні контексти та контекст до всіх користувачів. Проблема контекстно-обізнаної рекомендації з особою для ідентифікації контекстів користувача поряд з його важливістю у створенні профілю користувача є складною.

Метою цього дослідження є забезпечення ефективної рекомендаційної системи обізнаного контексту. По-перше, захопивши домінуючі контексти, з

огляду на різні характеристики даних користувачів, розгортаємо різні способи захоплення контекстів елементів користувача і рекомендуємо список елементів, які мають відношення до профілю користувача. По-друге, оновлюємо модель для підтримки продуктивності моделі, враховуючи часті оновлення даних електронної комерції.

Щоб подолати зазначені дослідницькі проблеми, досліджуємо більш ефективно моделювання контекстних факторів на уподобаннях користувачів над рекомендаційною системою так: використовуємо різні способи захоплення контекстів користувачів з урахуванням різних характеристик даних; беремо до уваги пріоритет, який такі системи надають контекстним факторам; дослідження моделює їх модель споживання для кожного елемента на основі того, як він споживається користувачами; включаємо користувачів або контексти елементів для пошуку відповідних елементів, які потрібно рекомендувати; оновлюємо профілі користувачів і предметів з урахуванням надходження нових даних. Ці чотири цілі призводять до формулювання чотирьох дослідницьких завдань цієї роботи.

1.2 Стратегії досягнення результату в рекомендаційних системах

Рекомендаційні системи з контекстною інформацією покращують розуміння користувача, віддаючи перевагу включеним контекстам, що впливають на попередні транзакції. Тим не менш, звичайні системи рекомендацій, що усвідомлюють контекст, розглядають всі контексти та їх пріоритети, які не відображають реальних ситуацій. Завданням роботи є захопити потенційні контексти для кожного користувача на основі характеристик історичних даних користувача. Ці дані дозволяють отримати контекстні ситуації користувача з декларативного відгуку користувача про елементи, враховуючи на приховані фактори користувача та елемента, явний відгук користувача або порядок цих контекстів, які з'явилися в транзакціях

користувача.

Враховуючи фактори, що лежать в основі поведінки кожного користувача, потрібні пріоритети користувачів або домінуючі контексти для кращого прогнозування потреби користувача повинні бути певним чином формалізовані та оцінені в моделі рекомендації. Оцінка всіх доступних елементів неефективна через велику кількість елементів. Щоб вирішити цю проблему, пропонується спочатку відфільтрувати елементи-кандидати користувача на основі їх перестановки згідно профілю користувача. Профіль користувача складається, але не обмежується, контекстами та вмістом користувача. Включатимемо вміст користувачів та контекстну інформацію з таких причин: для обробки великої кількості розріджених даних, які призводять до якісних груп користувачів і товарів, але на відміну від спільного фактору, включаємо групи користувачів і елементів в модель рекомендації, щоб краще дізнатися вплив кожного контексту на шаблон споживання користувача; фільтруватимемо пункт кандидатів для рекомендації, використовуючи перевагу аналогічних користувачів або елементів з подібними контекстами.

З появою нових даних модель не може відображати реальні контексти користувача або модель споживання, що може погіршити якість рекомендацій. Очевидним рішенням є перепідготовка моделі кожного разу, коли нові дані надходять. Однак це буде дорого і неефективно. Тому, врахуємо розміщення оновлення моделі для змінених профілів користувачів, використовуючи два різних підходи залежно від того, чи є характеристики програми не інтерактивними або інтерактивними. Для неінтерактивної програми рекомендацій спочатку перевіряємо, чи впливають нові дані на поточний профіль користувача. Коли це не так, то нам не потрібно оновлювати модель для цього користувача. Для інтерактивної рекомендаційної системи рекомендуватимемо користувачеві надати зворотний зв'язок, а потім вчитися на цьому, щоб уточнити налаштування користувача.

З огляду на питання дослідження та їх цілі, досліджуватимемо різні методи необхідні для захоплення контекстів користувачів та елементів для різних

характеристик даних та застосунків. Контексти користувача можуть бути відображені з текстів відгуку користувачів або послідовності транзакцій. Однак контексти користувачів розвиваються динамічно і не можуть бути повністю відображені з їх попередніх транзакцій. У цьому випадку розглядатимемо залучення користувачів до опису їх поточних контекстів шляхом взаємодії з системи. Виходячи з характеристик наявних даних, декларативна контекстно-свідома рекомендація може бути виконана трьома різними способами, розглядаючи всі проблеми, які необхідно вирішити.

Розглянемо можливу стратегію для рекомендації з контекстної обізнаності журналу відгуків користувачів. Враховуємо набір відгуків від користувача і всі відгуки на елемент, огляд історії користувача згідно контекстної рекомендації спрямованої на повернення набору елементів, яка стосується раніше визначених параметрів користувача та контекстного опису елемента. В подальшому, на наступному кроці, фіксуємо контекстну інформацію з текстів відгуку користувачів разом з оцінкою користувача та метаданими. Пропонуємо вікно користувача непосильної матричної факторизації для тематичне моделювання, щоб знайти домінуючі контексти користувача і домінуючі контексти пункту з товаром. Важливість цих контекстів у рекомендації моделюється двома різними способами: з використанням існуючої моделі; запропонованням нової рекомендації контекстної моделі, яка вивчає та включає важливість кожного контексту безпосередньо в моделі. Включаючи велику кількість елементів і користувачів, спочатку групуємо схожих користувачів і елементи на основі їх профілів, а потім знаходимо елементи кандидата, щоб розрізнити цільового користувача від інших користувачів, які знаходяться в тій же групі, що і користувач. Нарешті, щоб вирішити часті оновлення, пропонуємо оновити тільки профіль користувачів новими даними, а не повторно навчати модель з усіма даними. Теоретичний та емпіричний аналіз показує, що запропонована таким чином стратегія персоналізації користувачьких та домінуючих контекстів призводить до більш точної рекомендації, ніж звичайні системи рекомендацій з контексту. Інтерактивна контекстно-свідома рекомендація з врахуванням на

попереднього користувача з історією транзакції для пункту товарів, контексти, прагне використати механізм взаємодії для користувачів, щоб представити свої поточні потреби, включити цей прямий зворотний зв'язок до системи для уточнення моделювання уподобань користувача і повертає найбільш релевантні рекомендації для задоволення поточних потреб користувача. Захоплення контекстів, що лежать в основі поведінки користувача в попередніх діях, не завжди може працювати для всіх систем, особливо коли поточні потреби користувача відрізняються від тих, які вказуються регулярною поведінкою. Це може бути обумовлено зовнішніми ситуаціями, що впливають на користувача, такими як критична подія часу або спеціальне упорядкування. У цьому випадку пропонується рекомендація, яка полегшує взаємодію між системою та користувачами, щоб вони могли удосконалювати свої контексти до рекомендації. Для вирішення пропонується персоналізований зважений контекст – усвідомлена матрична факторизація, щоб знайти домінуючі контексти користувача, що дозволяє персоналізувати рекомендації важливих контекстів для кожного окремого користувача. Потім вводимо три нові стратегії вибору питань, які отримують переваги користувачів через відповіді користувачів на систему. Ці стратегії вибору питань також стосуються вибору елементів кандидатів, які найбільш відповідають поточним уподобанням користувача. Розглядаємо також часті оновлення за допомогою контекстного навчання, яке моделює відповідь користувача на питання, щоб повторно штрафувати налаштування користувачів, з метою повернути кращий набір рекомендацій для задоволення користувачів. Теоретичний та емпіричний аналіз показує, що наші поставлені інтерактивні та стратегії відбору питань призводять до точної рекомендації, ніж базові лінії.

Двонаправлена послідовна рекомендація з контекстної обізнаності з огляду на транзакційну послідовність цільового користувача, двонаправленого контексту рекомендація спрямована на повернення списку пунктів товарів, які мають найвищі ймовірності споживання користувачем на наступному кроці. Тому, включаємо тексти для перегляду користувача та / або метадані елемента для захоплення користувача. Для вирішення домінуючих контекстів пропонуємо

контекстний графік, а саме контекстно згрупований нормалізований точковий взаємний інформаційний граф, який фіксує властиві та різноманітні контексти користувача на основі семантичної подібності між контекстами у зборі даних. Потім розробляємо графік контексту транзакцій, щоб передбачити ймовірність транзакції, що відбувається в кожному контексті. Також, визначаємо важливість кожного контексту в прогнозуванні відповідних пунктів в рекомендаційних правках. Використовуємо стратегію оновлення моделі на основі структури кластера користувача для вирішення частих оновлень даних. Теоретичний і емпіричний аналіз показує, що запропонований метод перевершує існуючих конкурентів з точки зору послідовної рекомендації.

1.3 Методи фільтрації рекомендацій для користувачів

Було запропоновано багато підходів для рекомендації пункту. Спільна фільтрація, фільтрація на основі контенту і методи, які поєднуються з ними, є одними з найпопулярніших стратегій, реалізованих в рекомендаційних системах [31-34]. З огляду на користувача, пункт як рейтинг за користувачем спрямований на прогнозування оцінок необслуговуваних елементів на основі оцінок, які використовуються. Існуючі методи спільної фільтрації можна класифікувати на основі пам'яті і моделей [35]. Методи спільної фільтрації, які базуються на пам'яті, прогнозують оцінки, використовуючи взаємодію користувача та елемента на основі найближчого сусіда на основі користувача або елемента [36-40]. Зокрема, в підході на основі користувача, елемент рекомендується, коли він високо оцінений іншими користувачами з налаштуваннями, подібними до уподобань цільового користувача. У підході на основі елемента [41] рекомендується елемент, якщо він схожий на попередні елементи, які були високо оцінені цільовим користувачем [42]. Використовуючи метод на основі користувача як приклад, можна використовувати k-найближчий сусід, щоб знайти k найбільш схожого

користувача з цільовим користувачем, для якого показник подібності, такий як кореляція коефіцієнта Пірсона або векторна косинусоподібна схожість, може бути використана для вимірювання подібності між двома користувачами. Є також кілька поширених альтернатив для показників подібності в літературі, таких як скоригована косинус Схожість евклідовому відстань і жакардовий коефіцієнт [43–45].

Модель згідно спільної фільтрації тренує доступну рейтингову матрицю та вивчає параметричну модель, яка може бути використана для прогнозування оцінок обслуговуваних предметів. Модель прихованого фактора з низькорейтичною матричною факторизацією [46, 47] була однією з найпопулярніших. Латентний чинник моделі може передбачити приховані інтереси користувача, рейтинги за прихованими факторами користувача та елемента. Найпоширеніший моделі приймають в рахунок користувач і пункт упередження, який є ймовірно може бути. Параметри моделі можна оптимізувати, мінімізуючи помилку прогнозування. Модель оптимізації може проводитися зі змінними згідно методу найменших квадратів або стохастичного градієнтного спуску [47]. Однак, з огляду на велику кількість елементів і користувачів, цей рейтинг на основі спільної фільтрації не може уникнути розрідженості рейтингів в онлайн-спільнотах. Новий користувач повинен оцінити кількість елементів, перш ніж може бути повернуто точну рекомендацію, а новий елемент може бути рекомендований тільки після отримання певної кількості оцінок користувачів. Таким чином, надійність рекомендації не може бути гарантована. Щоб вирішити цю проблему, деякі дослідження виявляють і автоматично ставлять ключові відсутні дані [49], застосовують метод глибокого навчання, щоб зрозуміти, як використовувані переваги призводять до налаштувань після використання, або присвоюють різні ваги сусідам обраного користувача [50]. Тим часом, інші методи [51-56] включають рекламну інформацію про користувача та елемент, включаючи стать користувача, адресу, функції елемента, категорії, теги, тематичні інтереси та коментарі. Ця додаткова інформація називається вмістом у системі

рекомендацій. Цей метод, який включає вміст у систему рекомендацій, пізніше відомий як метод на основі вмісту спільної фільтрації.

Метод фільтрація на основі вмісту дає рекомендацію, коли вміст елемента-кандидата схожий на вміст споживаних користувачем елементів [54, 55]. Таким чином, елементи та профілі користувачів відіграють важливу роль у методі. Профіль користувача та елемента є структурованим представленням інтересів користувачів та описів елементів відповідно [56]. Профіль елемента можна визначити як вектор функцій елемента. З огляду на профіль користувача та елемента, метод спільної фільтрації проводить рекомендацію пункту, проаналізуючи профіль кожного елемента-кандидата з профілем активного користувача. Елемент з найбільшою схожістю буде рекомендований активному користувачеві. Схожість між профілями користувача і елемента можна виміряти за допомогою матриці подібності, такої як косинусоподібна схожість. Крім функцій елемента, профіль користувача може бути побудований з метаданих користувача, таких як адреса та стаття, або з тексту, такого як відгуки користувачів. Відмічений вміст з текстових особливостей можна використовувати, прийнявши методи обробки природної мови [57]. Такі методи, як латентний розподіл діріхле та подібні [58, 59], ненегативні матричні факторизації [60, 61] та частота зворотного документа [62, 63] надають непогані результати.

Тематичне моделювання корисно для вилучення функцій, коли користувачі мають мало елементів з рейтингом [64]. Частота перевернених термінів використовує статистику для визначення важливості слова для документа в колекції. Зокрема, важливість слова прямо пропорційна кількості разів, де воно з'являється в документі. Слово, яке з'являється майже у всіх документах, матиме значення, близьке до 0. У системі рекомендацій worda і документи в тематичному моделюючому мислені можуть бути представлені контентом і відгуками користувачів повторно. Зміст, який відбувається майже в кожному огляді, буде мати менше значення. Таким чином, наприклад, відгук про знижку може не бути пов'язаний з оглядом про угоду. Цей недолік може

бути вирішений прихованим розподілом Діріхле. Латентний розподіл Діріхле [59] є одним з найпопулярніших методів моделювання тем, що використовуються для визначення тем, до яких належить документ, на основі слів, які він містить. Наприклад, це може виявити, що огляд про знижку та угоду належить до тієї ж ціни теми. Таким чином, цей метод може обмежити відгуки, які мають одне з цих слів (знижка, угода) пов'язані між собою. У контексті моделювання теми ці тематичні ймовірності пропонують чіткий опис документа. Представлення даного методу має три рівні параметрів. Параметри рівня корпусів, які будуть відібрані один раз в процесі створення корпусу. Змінна рівня огляду, яка відбирається один раз за відгук. Змінні рівня вмісту і відбираються один раз для кожного контексту в кожному перегляді. Метод дає змогу отримати кластер вмісту, розподіл вмісту для кожної теми та розподіл тем для кожного огляду. Остаточні розподіли тем для кожного огляду в методі дозволяють згрупувати рецензію з пов'язаним вмістом під однією темою.

Не від'ємна матрична факторизація має ряд переваг перед звичайними підходами методу Діріхле, включаючи менше параметрів у процесі моделювання та можливість класифікувати більш складні теми, ніж [65]. З огляду на список документів, де документ складається зі списку слів, цей метод використовується для визначення того, наскільки добре кожен документ вирівнюється з кожним вмістом. У домені системи рекомендацій важливо, що важливий вміст від кожного користувача може бути зібраний з його / її рецензії користувача. Таким чином, документ може бути представлений як рецензія, і кожне слово, яке містить документ, є змістом рецензії. За цим методом представляються ці відгуки як рецензійна матриця і апроксимується матриця як добуток двох k -від'ємних не негативних латентних факторів. Наближення матриці проводиться шляхом випадкової ініціалізації обох матриць. Ця випадкова не негативна ініціалізація ваги зазвичай призводить до нестабільного результату для кінцевого фактора. В роботі [67] вирішити цю проблему запропонували через детерміновану ініціалізацію з від'ємним подвійним розкладанням значення, що підходить для розріджених матриць.

Вміст у системах рекомендацій проходить фільтрування. Фільтрування на основі вмісту використовує вміст елементів даних для прогнозування їх релевантності на основі профілю цільового користувача. Однак, як правило, рекомендація на основі системи фільтрації не враховує рейтинги користувачів пунктів у процесі рекомендації, що ускладнює визначення того, чи відповідає рекомендація очікуванням користувача. Наприклад, користувачеві може бути рекомендована пара неякісного взуття, подібного до придбаного користувачем в минулому. У зв'язку з цим недоліком було запропоновано ряд підходів до включення якості пунктів з точки зору рейтингів або популярності. Запропоновані підходи до поліпшення якості рекомендацій і виділення вмісту, який використовується в деяких найсучасніших системах рекомендацій на основі контенту, [68] рекомендують відповідні журнали або конференції знаки продавця на основі корисної інформації рукопису, включаючи назву, анотацію, автора та посилання статей. Дослідники пропонують новий метод вибору функцій, щоб отримати багаті інформативні особливості з кожного рукопису. Рекомендації потім генеруються за допомогою класифікатора регресії. Деякі роботи [69-71, 75] поєднують в собі як додаткові функції користувача / елемента, так і ступінь зацікавленості користувачів у елементах рекомендації. В роботі [69], було запропоновано розширити тип елементів, використовуючи відповідні функції і структури на основі графіків. Використовуючи ці функції, вони здатні збирати більш точну інформацію про предмети, тим самим покращуючи точність рекомендацій. В роботі [70] оброблялась розрідженістю даних система фільтрації, додаючи дані тегів та досліджуючи релевантність між тегами та функціями елемента. Користувачі вважають, що за краще буде розвиватися з плином часу і, отже, враховували тимчасовий вплив при визначенні налаштувань тегів користувача. В роботі [71] пом'якшив рідкісність рейтингів та проблем холодного старту, використовуючи тексти рецензій для висновку та підвищення рейтингів. Так само, в роботі [76] використовувався текстовий вміст раніше відвіданого об'єктів інтересів користувача для отримання більш точних уподобань користувача щодо місця. Дослідники пропонують дві

моделі, перша з яких вивчає тимчасові переваги користувачів, сегментуючи набір даних реєстрації на різні часові сегменти, а друга змінює метод Діріхле з урахуванням часу. Також, було запропоновано рекомендацію на основі класифікації щодо декількох атрибутів одягу, витягнутих із зображень одягу. Запропонований метод дозволяє рекомендацію одягу, яка відповідає атрибутам раніше шуканого зображення цільовим користувачем. В роботі [72], запропоновано використовувати як спільну, так і контент-інформацію у своїй системі рекомендацій щодо музики. Значущі особливості музичних предметів витягуються за допомогою графа знань, а контекстне утворення, таке як теги та текстові описи, витягуються, щоб збагатити семантичне значення ознак. У зображеннях рекомендують в роботі [73] використовувати згорткову нейромережу на зображеннях продуктів, щоб змоделювати людське відчуття візуального зв'язку між об'єктами. Запропонована модель може дізнатися, як різні візуальні розміри ознак продукту пов'язані один з одним у різних типах продуктів. Іншим прикладом того, що використання візуальних елементів має функції в рекомендаційних системах є робота [74], в якій витягають низькорівневі функції, такі як рух і освітлення з мультимедійного контенту, і доповнюють їх високорівневими функціями, такими як жанр, сюжет і режисер. Включення вмісту в рекомендаційну систему з фільтрацією дозволяє моделі рекомендувати елементи, які раніше споживалися будь-якими користувачами. Однак, система фільтрації припускає, що користувачі, які відвідали ті самі елементи будуть мати однакові профілі, незалежно від їх уподобань для цих елементів. Це припущення, однак, не може бути в реальності, коли кожен користувач має різні переваги для елементів.

Розглянемо системи рекомендацій з контекстною обізнаністю. Попередньо розглянуті рекомендаційні системи включали вміст для підвищення якості рекомендацій. На додаток до змісту, контекстуальна інформація також використовується в системних темах, щоб забезпечити кращі персоналізовані рекомендації користувачам [11, 77-79].

Рекомендаційні системи з контекстною обізнаністю визначають контекст

як «взаємопов'язані умови, в яких щось існує або відбувається», такі як навколишнє середовище та налаштування. У системі рекомендації контекст — це будь-яка інформація, яка може покращити якість рекомендації, яка не може бути отримана лише з оцінки елементів користувача [80] або «будь-якої інформації, що характеризує ситуацію, пов'язану з взаємодією між додатками та навколишнім середовищем» [81]. У роботі [82] контекст визначається як намір нас щодо діяльності з придбання товару або споживання товару. З огляду на ці визначення, насправді немає чіткої різниці між змістом і контекстами. Таким чином, в цьому дослідженні обидва ці терміни використовуються взаємозамінно.

Декларативні контексти є найбільш значущими контекстами, оголошеними кожним користувачем при проведенні транзакцій. Користувачі можуть представляти свої декларативні контексти через свої попередні транзакції користувачів, такі як відгуки, які супроводжують оцінки, або зворотний зв'язок, який добровільно вводиться користувачами під час системного взаємозв'язку. Наприклад, з відгуків користувачів можна зафіксувати, що певний користувач може слухати пісню через її «помітний ритм», в той час як інший користувач може слухати ту ж музику через її «поточний настрій». Ідентифікація цих декларативних контекстів дозволяє рекомендації зіставляти елементи з уподобаннями користувачів. Загалом, намір користувача можна отримати трьома способами від відповідних джерел до споживання користувачами: явно, неявно і висновок [11]. Наприклад, користувачі можуть заповнити опитування про свій намір покупки. Неявний підхід фіксує поведінку користувача під час транзакції за допомогою зовнішньої системи моніторингу, такої як журнал користувача. Наприклад, користувач неодноразово відкриває статтю про зміну клімату і залишає її відкритою на кілька хвилин, перш ніж закрити її. Тим часом, підхід до висновку витягує контексти, в яких користувач проводить транзакцію з наявних даних. З цих трьох підходів, метод висновку є більш ефективним, тому що наміри користувачів також доступні під час або після покупки товару.

Контекстні категорії в роботі [83] характеризують контекст на такі основні категорії: індивідуальна, час, діяльність та реляційна. Виходячи з цих поточних

категорій, в роботі [84] запропоновано сім контекстних категорій: обчислення, місце розташування, час, фізичні умови, активність, ресурс і користувач. Всі вони допомагають покращити ефективність системи.

Обчислювальні контексти, такі як мережа, апаратне та програмне забезпечення, можуть бути використані для рекомендації відповідних ресурсів споживачам, враховуючи пристрій, який вони використовували. Ці контексти були широко вивчені. Розташування контекстів відноситься до місця, пов'язаного з діяльністю користувача [85]. Наприклад, координати місцезнаходження користувача, близькість об'єктів в межах місця і орієнтація користувача, що може вказувати на те, в якому напрямку рухається користувач. Часовий контекст можна класифікувати як певний і невизначений. Певний часовий контекст визначається як часовий проміжок з конкретними початковими і кінцевими точками. Тим часом, невизначений час контексту не має встановленого періоду. Наприклад, контексти часу в сеансі користувача в онлайн-застосунку для покупок визначаються як періодичні події, які відбуваються під час цього сеансу. Фізичні умови контексту відносяться до зовнішніх середовищ або обставин, за яких знаходиться система або користувач. Наприклад, атмосфера, тепло, світло і звук. Захоплення фізичних контекстів допомагає у виборі відповідних місць, які рекомендували цільовому користувачеві, особливо в рекомендаціях щодо подорожей. Контекст діяльності відноситься до завдань або дій, що виконуються користувачем. Ці контексти можуть також включати тематичні інтереси користувача, виражені в коментарях користувача або явних взаємодіях користувачів. Фіксуємо контексти активності користувачів, особливо їхні тематичні інтереси, з відгуків користувачів, які супроводжували рейтинги елементів. Контекст ресурсів є загальним описом ресурсів. Ці контексти можуть бути вручну ідентифіковані експертом або витягнуті з анотацій інших користувачів [84]. Контексти користувача представляють характер та інтереси користувача, включаючи основну особисту інформацію (ім'я, стать, освіту), інтереси користувачів, які доступні в його коментарях або тегах [84]. Інтереси користувачів є ключовими характеристиками для персоналізованих рекомендацій, оскільки вони визначають переваги користувача.

Розглянемо інтерактивні системи рекомендацій. Хоча кілька рекомендаційних систем були розроблені і розгорнуті для відповідних елементів користувача, вони часто поводяться як «чорний ящик», так як вони не дають користувачам ніякої системної логіки або обґрунтування рекомендацій [53]. Це може призвести до проблем довіри, коли рекомендації не відповідають очікуванням користувача. Крім того, налаштування користувача можуть змінюватися в результаті конкретної домовленості або критичної в часі події, яка вимагає від користувача зробити щось відмінне від його звичайної поведінки. Однак більш ранні підходи до рекомендаційних систем обмежують користувачів від зворотного зв'язку з системою, що надає свої переваги, щоб отримати кращі рекомендації. Оскільки важко точно передбачити миттєвий інтерес користувача, існує необхідність встановити підходи, які дозволяють користувачам удосконалювати свої переваги, взаємодіючи з системами. Інтерактивні системи рекомендацій показують засіб адресації і цим розвиваються контексти користувача, дозволяючи взаємодію між користувачами і рекомендаційними системами через цикл зворотного зв'язку. Вони представляють основу візуалізації для інтерактивних систем рекомендувачів. Вузол даних користувача включає, але не обмежується рейтингами користувачів, кліками, пошуком або історією веб-перегляду, які використовуються як основа для створення персоналізованого рекомендаційного засобу. Вузол контексту містить контекстну інформацію, таку як місцезнаходження користувача, час, поточна активність та інтерес, всі з яких використовуються контекстно-обізнаними системами для кращого розуміння уподобань користувача. Обробник рекомендацій отримує і обробляє інформацію з даних користувача та / або контекстних даних, перетворюючи її в більш цінні знання, які зберігаються в середньому вузлі. Типовими прикладами даних середнього вузла є список користувачів, схожих на активного користувача, група елементів, які часто збираються разом, і список найпопулярніших елементів. Потім обробник рекомендацій використовує ці дані для створення списку рекомендацій для активного користувача на основі його / її інтересу. Наприклад, в елементі на основі систем

фільтрації, рекомендації двигун буде рекомендувати виправити список елементів, які активний користувач раніше купив. Цей список представлений вузлом рекомендації. На основі цього фреймворку активний користувач може взаємодіяти з будь-якими вузлами. Однак існуючі інтерактивні системи рекомендацій, як правило, зосереджуються лише на одному або декількох вузлах цієї структури.

Розглянуті кілька рекомендаційних систем, які дізнаються про переваги користувачів, самостійно ставлячись до їх історичних транзакцій. Таким чином, передбачається, що кожна угода не залежить від інших. Одним з недоліків цих рекомендаційних систем є те, що вони не можуть природно захопити еволюцію уподобань користувачів, якщо вони не включають додаткові дані, такі як контекстна інформація. Інтуїтивно, порядок попередніх транзакцій користувача *pro-vides* розуміння еволюції його / її уподобань.

Послідовна залежність транзакцій користувача враховується системами рекомендувача для захоплення поточних і останніх уподобань користувача з метою надання більш точної рекомендації [36]. Ці рекомендації стали відомі як послідовні системи рекомендацій. Взаємодія користувачів-елементів, як правило, базується на порядку, в якому вони відбуваються. У реальних ситуаціях попередні транзакції користувачів впливають на їх подальшу поведінку. Такі послідовні залежності зазвичай існують в даних транзакцій, але ігноруються або не добре захоплюються традиційними рекомендаційними системами на основі вмісту або спільною фільтрацією, що, по суті, сприяє розвитку.

Популярність продукції змінюється з часом. Наприклад, в останнє десятиліття багато клієнтів, які були шанувальниками одних систем, перейшли на більш інноваційні і якісні. Зміна популярності товару з плином часу також може призвести до зміни уподобань користувача. Таким чином, потрібно краще зрозуміти користувачів, вивішивши їх переваги з порядку, в якому відбуваються їх транзакції. Споживач може враховувати різні фактори при виборі онлайн-товару. Різні контексти, як правило, призводять до того, що споживач йде іншим шляхом до рішення, яке часто ігнорується традиційними системами

рекомендацій. А з іншої сторони, використовуйте попередню послідовність операцій як контекст для редагування наступного пункту, який споживач може придбати в найближчому майбутньому.

Розглянемо рекомендаційні системи та методи їх побудови з використанням моделі глибокої нейронної мережі. Побудова моделей глибоких нейронних мереж, як правило, базується або на періодичній нейронній мережі та графі мережі. Використовуючи ці підходи та використовуючи паралельний сеанс, міні-пакетний вибір вибірка виводу та функцію втрати ранжирування можна об'єднати декілька підходів у зваженому гібридному підході. Незважаючи на те, що методи на основі нейромереж показали вражаючу здатність моделювати послідовності, але їх продуктивність обмежена для довших послідовностей, які несуть високу вартість навчання.

Оцінка є важливою частиною дослідження рекомендаційних систем. Оцінки вивчають рекомендаційну систему з точки зору їх ефективності. У більшості рекомендаційних систем оцінка ефективності призначена для визначення точності прогнозів та / або відповідності рекомендованих елементів до уподобань користувача. З іншої сторони, оцінка ефективності вивчає систему рекомендацій з точки зору обчислювальної вартості або складності та її прийнятності для повернення рекомендацій протягом кінцевого, бажано короткого періоду часу. Запропоновано ряд методів оцінки для оцінки ефективності рекомендаційних систем. Ці оцінки можуть бути як в автономному режимі, так і онлайн. Онлайн-оцінка вимагає взаємодії в режимі реального часу з онлайн-користувачами для відстеження їх поведінки, такої як ставки кліків і конверсія покупок. Оскільки онлайн-оцінки є трудомісткими та дорогими, більшість існуючих досліджень покладаються на офлайн-експерименти. В автономній оцінці алгоритм використовується для прогнозування певних значень, відсутніх у доступних даних, а результати аналізуються за допомогою показників оцінки.

Автономні рекомендуючі системи мають три класи прогнозування: метрики прогностичної точності; показники точності класифікації; показники точності рангу. Показники прогностичної точності або завдання з

прогнозування, які передували, використовуються для оцінки необслуговуваних елементів з відсутніми оцінками. Точність оцінюється шляхом вимірювання того, наскільки близькі прогнозовані рейтинги до фактичних оцінок користувачів. Тим часом, завдання прогнозування ранжирування використовуються для рекомендації цільовому користувачеві зі списку кандидатів. Ефективність РС оцінюється за допомогою класифікаційних показників точності, які визначають, чи є класифікації правильними.

1.4 Постановка задачі дослідження

В результаті проведеного аналізу відомих рішень та проблематики предметної області дослідження необхідним і таким, що потребує подальших досліджень та є актуальним є продовження досліджень щодо розробки методів створення рекомендаційних систем.

Результатом таких досліджень має бути метод для створення рекомендаційних систем, який би усував недоліки попередніх використовуваних методів і стратегій.

Для досягнення поставленої мети необхідно:

- провести аналіз відомих методів і засобів для рекомендації користувачів товарів і послуг засобами інтернету;
- розробити модель рекомендаційної системи;
- розробити модель поведінки користувача з врахуванням особливостей проаналізованих недоліків в цій предметній області;
- здійснити моделювання системи та результати експериментів з моделлю;
- розробити метод та алгоритми створення рекомендаційних систем;
- здійснити концептуальну реалізацію запропонованого рішення.

1.5 Висновки до першого розділу

За результатами проведеного аналізу щодо поняття рекомендаційних систем, методів їх створення, необхідності використання, особливостей проектування з врахуванням потреба користувачів було встановлено межі предметної області дослідження, визначено основні поняття, також виділено недоліки у відомих рішеннях щодо створення рекомендаційних систем, стратегіях роботи з користувачами та пропонованими даними, особливостях розробки, що впливають на досягнення ефективності, поставлено задачі дослідження та способи вирішення цих задач.

2 МЕТОДИ СТВОРЕННЯ РЕКОМЕНДАЦІЙНИХ СИСТЕМ З КОНТЕКСТНОЮ ОБІЗНАНІСТЮ ЗГІДНО ЖУРНАЛУ ВІДГУКІВ КОРИСТУВАЧІВ

2.1 Декларативна система рекомендацій з контекстної обізнаності

Велика кількість онлайн-контенту зробила майже неможливим для соціальних користувачів ефективно навігуватися по ним. Багато компаній, такі як Amazon, Google News і Netflix, розробили персоналізовані рекомендації для вирішення цієї проблеми. Вважається, що контексти сильно впливають на якість рекомендацій. Різні контексти вбудовані в ці системи для підвищення їх ефективності. Однак різні цільові користувачі мають різні контексти, коли вони вибирають онлайн-елемент. Контексти ідентифікації та персоналізації користувачів дозволяють рекомендації відповідати елементам з уподобаннями користувачів. Аналогічним чином, контексти елементів, також, можуть по-різному характеризуватися кожним із користувачів.

Надмірні контексти введуть до додаткових витрат часу, одночасно зменшуючи якість рекомендацій. Варто дослідити більш персоналізовані контексти для кожного цільового користувача та кожного елемента в рекомендації, щоб забезпечити кращу продуктивність системи. Це дослідження вивчає користувацький огляд, історію, контекстно-обізнані рекомендації для додатків електронного бізнесу. З огляду на цільового користувача і набір елементів, декларативна рекомендація прагне автоматично вивчити домінуючі контексти і повернути список елементів, що мають найбільшу актуальність в домінуючому контексті підпростору. Він призначений для ефективного рекомендування онлайн-елементів для націлювання на користувачів більш персоналізованим способом. Це тому, що кількість онлайн-контекстів дуже велика, і є упередженість уподобань користувачів щодо цих контекстів. Наприклад, онлайн-елемент може мати сотні контекстів. Однак жодне з існуючих рішень не досліджувало вплив контекстної упередженості на якість рекомендацій. Традиційні системи

рекомендацій завжди виробляють відповідні елементи на основі оцінок користувачів [36, 38, 66]. Однак проблеми виникають, коли потрібно працювати з розрідженими рейтингами, в той час як з іншої сторони, розрідженість рейтингу є поширеною проблемою на інтернет-сайтах. Дослідники зробили великі спроби поліпшити якість рекомендацій. Існуючі контекст-обізнані рекомендаційні системи використовують контексти з репрезентативного подання, яке визначає контексти як попередньо визначений набір атрибутів, структура якого не змінюється з плином часу. Хоча попередні мають визначення контексту, які обслуговують програми, такі як поведінкові категорії [12], популярність статті [13], соціальна довіра [14], дії та діяльність користувачів . [15–17, 22, 23, 25], вони використовують попередньо визначені контексти в моделюючому середовищі даних, які не можуть адаптивно захопити домінуючі контексти кожного користувача, і, таким чином, не може гарантувати оптимальну якість рекомендаційної системи. Мотивовані ініціювання поточних рекомендаційних систем потребують двох систем рекомендацій, засновані на профілюванні, яке включатимемо в рекомендацію. Першою системою рекомендації є декларативний контекст – рекомендаційна система, яка включає декларативне профілювання користувача в рекомендацію. Тим часом, інший є декларативним користувачем – система рекомендації, що базується на профілі на основі контексту і включає як профілювання користувача, так і елемента в рекомендацію. Обидва підходи дозволяють персоналізувати контексти для кожного цільового користувача, автоматично вивчаючи його історію перегляду на інтернет-сайті.

В основі розробленого методу закладені дві описаних стратегій, які розглянемо детальніше.

Введемо нову тематичну модель базовану на вікні користувача, яка використовувати не негативну матричну факторизацію, щоб визначити домінуючі контексти, що впливають на рішення, що переслідує, досліджуючи причину рейтингів користувача. Використовуючи ці домінантні контексти, профіль цього цільового користувача описується як набір транзакцій у

домінуючому контексті підпростору. Для реалізації моделі розроблено новий алгоритм, який вивчає прогностичну модель цільового користувача на основі його профілю і проведено двофазну рекомендацію, спочатку фільтруючи кандидатів на основі оцінок користувачів, які мали домінуючі контексти, подібні до контекстів цільового користувача, а потім попередньо диктуючи перевагу останнього зі списку кандидатів предметів за допомогою прогностичної моделі. Алгоритм додаткового оновлення ефективно підтримує прогностичну модель. Таким чином, пропонується нова структура, яка використовує домінуючі контексти користувача, зміст елемента та користувача для декларативної рекомендації. Хоча декларативні контексти відображають актуальність елемента для користувача, ігноруючи контексти, які не викликають занепокоєння, можуть запобігти втраті великої кількості часу. Також, введемо нову графічну модель для вивчення домінуючих контекстів кожного цільового користувача, на основі яких декларативний профіль користувача представлений як набір транзакцій у домінуючому контексті підпростору. Вона добре фіксує наміри кожної транзакції. Крім того, розроблено нову двофазну декларативну рекомендацію, яка спочатку підказує кандидатів на предмет, а потім прогнозує остаточну перевагу, використовуючи нову прогностичну модель, яка включає профіль користувача. Це дозволяє уникнути непотрібних операцій. Розроблено новий алгоритм, який може ефективно підтримувати оновлення на інтернет-сайтах. Проведені великі випробування на двох великих реальних наборах даних для перевірки ефективності запропонованого рішення. Розширюємо функціональність тематичного моделювання для виявлення не тільки профілів користувачів, але і профілів елементів на основі відгуків користувачів про товари. Потім модель тестується, щоб переконатися, що вона може керувати профілями з обмеженою контекстною інформацією, співпрацюючи з групою користувачів та елементами з подібними профілями.

Отже, основні результати цієї роботи: запропоновано нову основу для рекомендації пункту, яка використовує персоналізовані та домінуючі контексти користувачів і елементів. Ці домінуючі контексти використовуються для створення

декларативних профілів користувачів і елементів. Зосереджуючись на найважливіших контекстах, запропонована модель знижує розрахункові витрати при виготовленні рекомендацій. Крім того, корпоративування профілю декларативного елемента, також, допоможе у генерації рекомендацій щодо елементів, які мають відношення до домінуючих контекстів цільового користувача. Запропоновано нову рекомендацію пункту, що знає контекст, яка враховує акаунт як профіль користувача, так і елемент. Цей новий контекстно-обізнаний елемент рекомендаційної системи, також, буде працювати над профілями з обмеженою контекстною інформацією, використовуючи групи користувачів та елементи на основі подібності профілю. Для підтвердження запропонованих рішень проводимо великі експерименти над двома загальнодоступними наборами даних, щоб перевірити рішення з точки зору його ефективності.

Розглянемо складові компоненти історії відгуків користувачів. Вона складається з трьох компонентів: створення декларативних профілів користувачів; розробка рекомендацій згідно декларативних профілів користувачів; оновлення рекомендаційної систем новими даними. З врахуванням користувача, його декларативний профіль визначається для кожного пункту товару, які він оцінив як рецензований. Зокрема, генерується декларативно з відгуків, а також змісту, так і пункту товару. Відповідна частина з відгуків визначається як контекст користувача (прихований). Тематичні моделі розгортаються для виявлення цих прихованих контекстів. Однак, найпростіше моделювання тем, таке як латентне співвідношення Діріхле є недостатнім для цієї мети, оскільки воно не розкриває приховані теми з кожного огляду самостійно. Тому, пропонується створення рекомендаційної системі згідно з методом фільтрації даних, яка працюватиме на «документах», які визначаються в цьому дослідженні як набір всіх відгуків, написаних користувачем. Отже, для користувачів створюється пункт користувача з оглядовим вікном. Отже, враховуючи набір відгуків, і відповідний набір термінів огляду будується матриця, в якій міститиметься ця відповідна інформація. Функція вагового зважування визначається у форматі частотно-зворотної частоти документа. Для декларативного профілювання

користувачів, ми стверджуємо, що кожен користувач має різні важливі контексти при споживанні предметів. Таким чином, вибираючи топові терміни для кожної теми, здійснюємо уникнення встановлення однакових факторів для оцінки та огляду, а також визначення однакових контекстів для всіх користувачів. Наприклад, якщо треба знайти п'ять тем з відгуків користувачів, встановивши $k = 5$, то для кожної теми додатково вибираємо топ-5 рейтингових термінів, які представляють цю тему. З огляду на наведений вище сценарій, матимемо максимум 25 термінів, які будуть розглядатися як контексти для відповідного користувача. Неможливо очікувати переваги користувача у всіх обраних 5 темах у кожному рецензованому оціненому пункті, що означає, що є рідкісним. Потім визначаємо усічену матрицю, зберігаючи в ній лише вибрані терміни, які називатимемо далі контекстами. Зокрема, для кожного терміну огляду перевіряємо їх появу у згенерованих контекстах користувача. Якщо термін існує в контекстах користувача, то отримуємо значення терміну з матриці. Припущення, що стоїть за цим визначенням, полягає в тому, що коли користувач включає контекст в огляд, то він розглядає аналогічну важливість цього контексту у всіх своїх відгуках. Крім того, також так отримується вміст користувача та елемента з метаданих. Нарешті, маючи функції користувача, визначаємо декларативний профіль користувача для користувача на елемент. Вивід цієї частини рекомендаційної системи є тематичною моделлю вікна користувача, яка охоплює контекстні налаштування користувача. Об'єднавши все разом з вмістом користувача та елемента, матимемо декларативний профіль користувача.

Розроблені персоналізовані декларативні профілі користувачів, які охоплюють контекст користувача, користувацький контент та вміст відповідних рейтингових / рецензованих елементів потребують для ефективного створення нового алгоритму декларативного контексту, рекомендувавши елементи, які відповідають декларативному профілю користувача. Він, наприклад, включатиме такі компоненти: вивчення користувача і моделей споживання на основі його, її декларативних профілів на всіх пунктах оцінки рецензування, а також рейтинги надані відповідним пунктам. Зокрема, визначатимемо це як задачу регресії,

розглядаючи незалежну змінну. В розширенні дерева можна визначити як середнє значення для навчального набору декларативні профілі для користувача на всіх рецензованих / рейтингових пунктах та середній внесок з кожної функції. Модель намагається зводити до мінімуму помилку прогнозування. Вивід – це модель споживання або прогностична модель користувача з урахуванням пункту товару, яка здатна передбачити рейтинг. Однак, враховуючи це як правило, це не можливо реалізувати і це не ефективно, щоб зробити прогноз для кожного пункту. Тому, пропонується здійснювати вибір підмножини елементів, які будуть розглядатися як кандидати на елементи, вибравши кількість подібних користувачів і ретельно оцінити їх рейтингові елементи. Вибір схожих сусідів, які знають контекст. Групування користувачів у групу можна зробити різними способами. Припустимо, що контексти були прикріплені кожного разу, коли користувач споживає елемент, який може бути захоплений через компонент контексту користувача. Потім групуємо користувачів на основі схожості їх контекстів. З огляду на активного користувача спочатку отримаємо декларативне профілювання користувача для всіх користувачів. Потім визначаємо групу користувачів, схожих відповідно до їх контекстів. Далі, визначаємо елементи кандидата, які потрібно рекомендувати. Всі попередні рейтингові елементи визначаємо і елементи, які не були оцінені користувачем як елементи кандидати. Щоб передбачити рейтинги ймовірного кандидата пункту, включатимемо зміст кожного кандидата пункту, щоб зробити тимчасовий декларативний профіль, який включатиме контекст користувача, зміст користувача та зміст елемента-кандидата. Таким чином, розміщується вся інформація, яка може вплинути на перевагу користувачів елементів. Таким чином, кожен з кандидатів пункту, буде мати той же профіль, за винятком рейтингової інформації для елемента кандидата, яка все ще відсутня. Однак це буде передбачено за допомогою навченої наступної моделі. Проектована рекомендаційна система працюватиме згідно профілювання декларативного користувача. Таким чином, необхідно перекваліфікувати модель, коли з'явилися нові дані, якщо продуктивність не погіршиться. Однак перепідготовка моделі

буде витратна протягом терміну виконання. Щоб вирішити цю проблему, розгортатимемо додаткове оновлення моделі рекомендаційної системи для налаштування з новими даними прибуття. Ці данні використовували для оновлення оффлайн моделі рекомендаційної моделі. При навчанні нової моделі використовуватимемо залишкову похибку попереднього дерева для підвищення продуктивності поточного дерева. Інтуїція цього додаткового оновлення полягає в тому, що нові дані прибуття принесуть нову перспективу до поточної моделі. Наприклад, зміни домінуючих особливостей користувача, щоб покрити цю потребу, потрібно лише перерахувати важливість кожної функції з точки зору покриття та прибутку.

Розглянемо ефективність декларативної рекомендаційної системи, в проведеному комплексному експерименті над двома групами даних. Експерименти проведено над даними з двох великих реальних наборів даних. Перший набір даних є підмножиною набору бенчмарків, який включав огляд ресторану користувачами через інтернет-сайт. Другий набір - набір даних відгуків готелю. Обидва набори даних були створені шляхом збереження користувачів, які оцінили більше, ніж середні відгуки. Було вибрано 40% користувачів з найбільшою кількістю відгуків в якості цільових користувачів. В таблиці 2.1 представлено результати таких наборів.

Таблиця 2.1 - Два набори експериментальних даних

Набір даних	Користувачів	Елементи	Перший набір	Другий набір
Кількість для першого набору	570	16	430	245
Кількість для другого набору	3250	81	345	81

В результаті проведених експериментів було оцінено ефективність та достовірність декларативної системи рекомендацій. Перед проведенням такого

оцінювання здійснимо її методологічне обґрунтування. При отриманні такого обґрунтування можна буде застосовувати її до конкретного експерименту. По-перше, оцінюємо параметри, які вирішують ефективність роботи системи. Існує два параметри, що впливають на продуктивність системи: оптимальний номер теми; кількість найкращих термінів, що зберігаються для кожної теми. Доведено, що оптимальний номер теми для виявлення теми – 10. Після встановлення номера теми оцінюємо лише ефект числа верхніх термінів, перевіряємо чутливість моделі і вирішуємо оптимальне значення. Потім оцінюємо ефективність рекомендаційних систем, використовуючи оптимальні значення. Для кожного набору даних використовуємо наступні шість підходів, щоб рекомендувати елементи одному набору цільових користувачів і порівняти запропонований рекомендаційний підхід з конкурентами. Після цього ми порівнюємо наші результати з рекомендаційної системи з альтернативою, щоб ще більше довести, що спроектована рекомендаційна система є кращим вибором для рекомендації. Після цього оцінюємо вплив оновлення інкрементної моделі на ефективність системи та порівнюємо ефективність нашої системи з існуючими рекомендаційними підходами та методами. Запропоновані альтернативні методи та існуючі конкуренти для порівняння такі:

1) декларативна контекстно-обізнана рекомендаційна система, в якій модель користувача була створена на основі бажань користувача для вмісту або властивості елементів;

2) декларативна контекстно-обізнана рекомендаційна система, в якій відміну від попередньої, включено як зміст елемента, так і різні контексти для кожного користувача;

3) інкрементні рекомендаційні системи, які підтримують додаткове оновлення моделі кожного разу, коли надходять нові дані про часове вікно;

4) рекомендаційна система, в якій використано техніку моделі матричної факторизації для прогнозування рейтингів для кандидатів на позиції, виходячи з історії поглядів користувачів, тому спочатку отримуються рекомендаційні кандидати за схожістю часу, типом та спільною фільтрацією між елементами;

5) рекомендаційні системи, які створені з використанням методів, що намагаються використовувати як рейтингову інформацію, так і інші неявні відгуки;

6) рекомендаційні системи, які створені з використанням методів, що відстежують поведінку клієнта, що змінює час протягом усього життя даних;

7) рекомендаційні системи, які створені з використанням методів, що повертають рекомендацію елемента на основі подібності споживання товарів користувачів;

8) рекомендаційні системи, які створені з використанням методів, що використовують лише рейтингову інформацію.

Ефективність системи оцінюється за допомогою метричної квадратної похибки засобів, в ній представлена інформація, що вказує на середнє значення квадратів помилок між передбаченнями і отриманим. Менша різниця вказує на кращу продуктивність системи прогнозування рейтингу. Оцінюємо ефективність системи з точки зору часової вартості обслуговування моделі за усім набором даних. Всі експерименти проводяться на сервері за допомогою використання потужностей процесора.

Спроектвана декларативна система рекомендацій, яка обізнана з контекстом, працює краще, ніж існуючі конкуренти, з двох причин. Вона фіксує більш персоналізовані контексти для кожного цільового користувача, які відображають наміри кожної з його транзакцій, і може краще захопити перевагу користувача в рекомендації. А також, вона видаляє неважливі контексти в рекомендації цільовому користувачеві, що є процесом операцій над нижчим контекстом підпростору, таким чином, побічний ефект надлишкових контекстів може бути усунений.

В результаті проведеної роботи запропоновано метод створення декларативної системи рекомендацій, яка може ефективно рекомендувати елементи на інтернет-сайтах і може мати кращу часову ефективність. Це досягається за рахунок декларативного підходу до профілювання користувачів, який витягує персоналізовані контексти для кожного користувача на основі його

або її текстів відгуків. В основі цього методу адаптовано двофазний алгоритм рекомендації пункту для вибору відповідних елементів щодо домінуючого контексту підпростору для цільового користувача ефективним способом. Також, в методі закладено алгоритм оновлення додаткової моделі, щоб підтримувати модель в динамічному середовищі. Було проведено експерименти, щоб оцінити запропонований рекомендаційний підхід. Експериментальні результати показали, що розроблений метод створення рекомендаційних систем покращує існуючі з точки зору ефективності для 40% активних користувачів. Розроблена рекомендаційна система була призначена для націлювання на найактивніших користувачів. Вона не є відповідним рішенням для рекомендацій щодо товарів, коли метою є прийняття ефективних рекомендацій для всіх користувачів, незалежно від їх залучення. Для виконання цієї вимоги розглянемо детальніше проблемні невіршені задачі і їх розв'язання імплементуємо в неї.

2.2 Контекстно-обізнана рекомендаційна система з декларативним користувачем та профілюванням елементів

Розглянемо нову розроблену контекстно-обізнану рекомендаційну систему з декларативним користувачем та профілюванням елементів, яка охоплює конструкції та генерацію рекомендацій користувача і елемента профілювання.

Розроблений метод містить введення метаданих елемента користувача, а також транзакції історії користувача, який включає текст відгуку користувача, що пояснює оцінку, а саме у форматі якого користувач оцінює, який елемент з текстом огляду. За такої побудови рекомендаційної системи досягнення покращення результатів буде здійснене через прагнення розкрити користувача і прихований елемент контексти на основі відгуків користувача для набору елементів і пунктів товарів чи послуг з набору користувачів. З огляду на декларативні профілі користувача, так і на елементи, система буде включати ці профілі в систему рекомендацій, щоб генерувати елемент, який, швидше за все, буде відповідати

перевагам користувача. Така рекомендаційна система повертає найбільш релевантний пункт цільовому користувачеві. В ній є два компоненти. Структура декларативної рекомендації профілю користувача-елемента зображена на рис. 2.1.

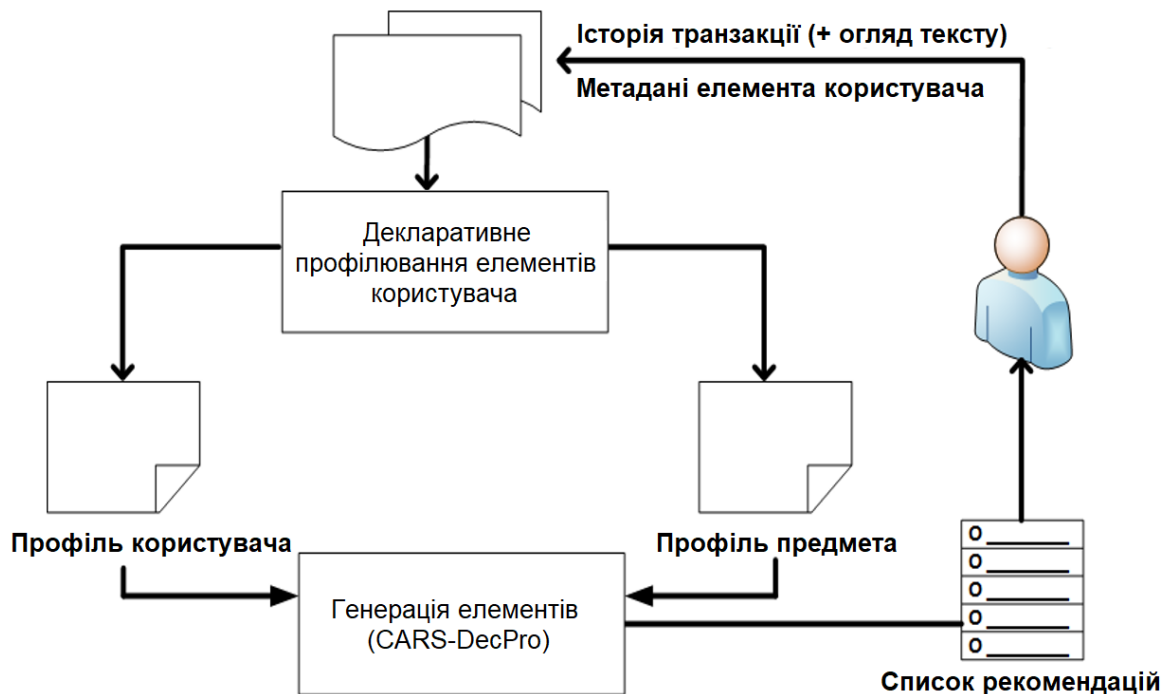


Рисунок 2.1 – Структура декларативної рекомендаційної системи з профілем користувача-елемента

Декларативний користувач і профілювання елементів визначає найбільш домінуючі контексти як для користувача, так і для елемента на основі попередніх транзакцій користувачів, особливо для текстових оглядів користувачів. Припустимо, що користувач буде писати тільки важливі контексти в своєму огляді елемента. Таким чином, елемент буде розглянутий їх домінуючими контекстами, а також декларативне профілювання як для користувача, так і для елемента враховує різні контексти для кожного користувача та кожного елемента. Рекомендація пункту генерує рекомендації пункту включати декларативний профіль як користувача, так і елемента. Інформація користувача і елемента з найближчим профілем використовується для підвищення ефективності рекомендації, особливо при роботі як з користувачем, так і з предметом з поганою контекстною

інформацією.

При виборі онлайн-елементів кожен користувач враховує різні контексти, які варіюються від особистих контекстів користувача або домінуючих контекстів, які має елемент. Користувач враховує контексти / аспекти елемента при виборі елемента, і кожен елемент має різні домінуючі контексти в порівнянні з іншими. Кожен користувач має різні важливі контексти при споживанні предметів, то кожен елемент має різні домінуючі контексти. Також ідентифікуємо профіль елемента на основі відгуків, отриманих від користувачів. Пункт декларативного профілю визначається кожним користувачем, який дає швидкість і огляд допункту. Зокрема, генерується декларативно з відгуків. Відповідна частина з відгуків визначається як контекст прихованого елемента. Отже, для того, щоб визначити профіль елемента, для елементів, є вікно огляду елемента. Отже, з огляду на набір відгуків для пункту, і відповідний набір термінів огляду будуватимемо огляд матриці термінів для пункту, і визначатимемо вагу терміну в огляді.

Функція вагового зважування визначається у форматі частоти термінів - обернена частота документів. Потім застосовуватимемо норму, щоб зменшити упереджений внесок як з загальних, так і з рідкісних термінів. Вона застосовуватиметься в кожному вікні елемента, яке розкладається на два k -мірних латентних фактори з властивістю, що всі три матриці не мають від'ємних елементів. Профілювання елементів охоплює домінуючі контексти елемента. В профілі елемента не визначаємо вміст елемента та контексти користувача як частину профілю, щоб уникнути надмірності з декларативним профілем користувача. Декларативне профілювання пункту за допомогою функції можна реалізувати у відповідному алгоритмі. Персоналізований профіль декларативного елемента, який охоплює домінуючі контексти елементів та профілі користувачів, які складаються з контексту користувача, користувацького вмісту та відповідного номінального / рецензованого вмісту елемента. Включатимемо як профілі користувачів, так і елементи в рекомендацію. Потрібно врахувати важливість контексту як для користувача, так і для елемента в рекомендації. Однак він виправдовує оцінку користувача або темами користувача, або темами елементів і

приймає подібні контексти над користувачами та елементами. Тим часом включає лише профілі користувачів, але не елементи. Розроблений для цього випадку алгоритм генерує рекомендацію елемента, зіставляючи профіль елемента з профілем користувача. Зокрема, на основі матричної факторизації потрібно включити декларативний профіль користувача до прихованих факторів користувача та декларативний профіль елемента до пункту латентних факторів. З цього моменту профіль користувача та елемента включається для покращення прихованих уявлень користувача та елемента відповідно. Коли два користувачі віддають перевагу одним і тим же елементам, то можна припустити, що обидва користувачі поділяють схожі інтереси. Таким чином, якщо користувачу сподобався елемент, який не був спожитий другим користувачем, то можна рекомендувати пункт першому користувачеві. Аналогічним чином, можна припустити, що користувач буде, як правило, споживати інший пункт, який поділяє аналогічні контексти або функції до пункту, що перший користувач раніше користувався. Групування користувачів на основі схожості їх профілю та елементів з подібністю їх домінуючих контекстів може здійснюватися різними способами. Припустимо, що контексти були прикріплені кожен раз, коли користувач споживає елемент і захоплені через користувача. Потім буде групування користувачів і елементів на основі подібності їх контекстів. В результаті буде отримано декларативне профілювання елементів користувача для всіх користувачів і елементів. Потім визначено групу користувачів, які схожі і є групою елементів, які схожі відповідно з їх контекстами. Використовуємо обов'язково пошук подібних користувачів та подібні елементи на основі їх профілів. Модель намагається звести до мінімуму помилку прогнозування. Проектована рекомендаційна система навчається на основі історичних даних транзакцій користувачів (наприклад, рейтингів) і визначає профіль як для користувача, так і для елемента для створення рекомендацій.

Розглянемо ефективність так проєктованої рекомендаційної системи, проводячи комплексні експерименти над двома реальними наборами даних. Ці два набори є загальнодоступними наборами даних. Всі набори даних були створені

шляхом збереження користувачів, які оцінили більше, ніж середні відгуки. Основні дані істини для цільового користувача - це дані, які оцінюються цим користувачем у тестовому наборі. Оцінюємо рекомендацію на успішність для кожного користувача, обчислюючи значення між прогнозованим рейтингом та рейтингом даних отриманим під час експерименту. Застосовуватимемо 5-кратну перехресну перевірку, і всі базові лінії налаштовані відповідно до роботи.

Розглянемо методологічні основи постановки експерименту. Проведені експерименти для оцінки ефективності проєктованої рекомендаційної системи. Дотримуючись оптимального параметра визначатимемо 10 як оптимальний номер теми і кількість найкращих термінів, що зберігаються для кожної теми. Оцінюємо ефективність рекомендаційної системи, використовуючи оптимальні значення. Для кожного набору даних порівнюємо запропоновані з п'ятьма існуючими конкурентами, які можуть бути розділені на дві групи. Це традиційна спільна фільтрація та контекстні рекомендації, а саме:

1) рекомендаційна система, яка використовує як рейтингову інформацію, так і інший неявний зворотний зв'язок;

2) рекомендаційна система, яка відстежує поведінку клієнта, що змінює час протягом усього життя даних;

3) рекомендаційна система, яка повертає рекомендацію елемента на основі подібності споживання елементів користувачів і використовує певні функції продукту в інтересах користувача та приховані функції, отримані для рекомендацій;

4) рекомендаційна система, яка має найкращий серед методів, які розглядають різні домінуючі контексти для кожного користувача, включаючи вміст елемента та користувача та текст відгуку користувача для визначення профілю користувача.

Експериментальні дослідження повинні використовувати більше двох наборів даних для однакових наборів цільових користувачів. Встановлюватимемо всі методи на їх оптимальні параметри параметрів, і рекомендуватимемо прогнозовані елементи цільовим користувачам. В такому випадку всі

запропоновані рекомендаційні системи показують кращу продуктивність, повертаючи найнижчу похибку в прогнозуванні елементів. Рекомендаційні системи з традиційною рекомендацією спільного фільтрування має аналогічну продуктивність, як контекстно-обізнані рекомендаційні системи з декларативним профілем користувача. Лише включення профілю користувача недостатньо для підвищення продуктивності рекомендаційної системи для цих двох наборів даних. Проектована рекомендаційна система може перевершити всі базові лінії, включивши як профіль користувача, так і елемент у рекомендацію пункту. Проектована рекомендаційна система має багато параметрів для вивчення. Оцінимо ефект параметра фактора елемента, щоб побачити ефект значення для рекомендації елемента. З результатів встановлено, що так спроектована система підвищує точність прогнозування. Однак, чим менше значення, то процес стає довше.

Оцінюватимемо ефективність включення профілювання як користувачів інтернет ресурсу, так і товарів ресурсу у рекомендації пункту, яка включає подання користувача в Інтернеті. Для обох рекомендаційних методів вибираємо адаптовані під вимоги рекомендовані елементи та несприятливо рекомендовані елементи. Після цього порівнюватимемо профіль елемента цих елементів з профілем поточного користувача. У зв'язку з обмеженням простору, відображаємо рекомендації пункту для двох найбільш активних користувачів, щоб представляти характеристичні ознаки: високо (***) і несприятливо (*) рекомендовані елементи. Спроектована рекомендаційна система рекомендує товари користувачу з більш схожим профілем на його профіль в порівнянні з рекомендаційною системою, представленою в попередньому параграфі, яка таких можливостей немає. Це показує, що включення профілю користувача та елемента в рекомендації елемента підвищують якість рекомендації, повертаючи більше схожих елементів до профілю користувача.

2.3 Висновки до розділу

В другого результаті виконання завдання з усунення недоліків відомих рекомендаційних систем було запропоновано спроектовану рекомендаційну систему з контекстної обізнаності, яка декларативно визначає як контексти користувача, так і елемента та ефективно рекомендує елементи користувачам на інтернет-сайтах. В ній запропоновано декларативного користувача та профілю елемента, який використовує попередні транзакції для вилучення персоналізованих контекстів для кожного користувача та елемента. При моделюванні тем рекомендаційної системи домінуючі контексти як користувачів, так і елементів використовуються для представлення профілю користувача та елемента. В подальшому групуємо схожих користувачів і елементи на основі схожості їх профілю. Далі, інтегруємо декларативного користувача і профілювання елементів, групи подібних користувачів і елементи в модель, щоб поліпшити здатність моделі повертати найбільш релевантні елементи кожному користувачеві.

Над спроектованою системою було проведено комплексні експерименти, щоб оцінити запропоновану рекомендаційну систему на двох реальних наборах даних. Експериментальні результати показують, що запропонований метод до створення рекомендаційних систем згідно елементів «декларативний користувач – контекст профілювання елементів – система рекомендаційних засобів», перевершує існуючі методи в термінах ефективності.

Програмні реалізації обох рекомендаційних систем, які спроектовані, доводять, що захоплення персоналізованих уподобань користувачів через їх заявлені інтереси в оглядах користувачів можуть підвищити якість рекомендацій щодо товарів для обох кадрів. Ці методи, з іншої сторони, не призначені для захоплення миттєвих змін у налаштуваннях користувача, які можуть виникнути в результаті спеціальних домовленостей або критичних у часі подій.

3 ІНТЕРАКТИВНА КОНТЕКСТНА РЕКОМЕНДАЦІЙНА СИСТЕМА

3.1 Вимоги до створення інтерактивної рекомендаційної системи

Велика кількість онлайн-товарів і користувачів створюють великі проблеми для навігації правильними продуктами для своїх зацікавлених користувачів у частих онлайн-транзакціях. У відповідь на цю вимогу, рекомендації системи призначені для кращого розуміння уподобань користувачів шляхом інтерполяції контекстів за транзакціями користувачів. У відповідь на великі проблеми для навігації правильними продуктами для своїх зацікавлених користувачів через велику кількість онлайн-товарів і користувачів в частих онлайн-транзакціях, використання цих систем рекомендується зацікавленим користувачам на основі історії активності користувачів. Однак у реальних застосунках на переваги користувачів впливають контексти / фактори, що беруть участь у прийнятті рішень, що не може відобразитися на їх поведінці історії. Таким чином, традиційні системи рекомендацій, що рекомендують контекст, не в змозі зафіксувати цю миттєву зміну уподобань, що може знизити якість рекомендації, введе його в оману, щоб прийняти невідповідну рекомендацію при прийнятті рішень. Але, соціальні користувачі прагнуть взаємодіяти з системою, коли вони проводять онлайн-діяльність. В результаті, стає багатообіцяючим підвищення якості рекомендацій, дозволяючи користувачам удосконалювати свої контексти за допомогою нової взаємодії з системою. Спроекуємо та розглянемо інтерактивну рекомендаційну систему для електронного бізнесу. З огляду на цільового користувача і набір елементів, інтерактивна система рекомендацій спрямована на використання взаємодії між системою для уточнення домінуючих контекстів і переваг, а також для повернення списку елементів, які мають найкраще значення в домінуючому контексті підпростору. В якості критерія доцільності і визначення переваги приймемо ефективність інтерактивної контекстно-обізнаної рекомендації щодо онлайн-платформ для застосунків електронного бізнесу. Щоб вирішити цю проблему, необхідно вирішити три

ключові завдання. По-перше, потрібно побудувати модель, яка дозволяє персоналізувати переваги користувачів і розставляє пріоритети різним факторам в остаточній рекомендації. Нездатність захопити персоналізовані переваги користувачів та його вплив на прийняття рішень, може ввести систему в оману, щоб рекомендувати правильно. По-друге, потрібно розробити стратегію вибіркового вибору питань, щоб отримати зворотний зв'язок користувачів і максимізувати ефект взаємодії. Кожне питання повинно представляти поточні міркування користувачів і ініціювати їх у наданні зворотного зв'язку для уточнення своїх уподобань. Без ефективної стратегії вибору питань система може вибрати такий варіант, що знижує ефект питання, що ставить ітерацію. Нарешті, потрібно розробити розширений підхід до навчання, який включає відгуки про рейтинг користувачів в модель. У зв'язку з впливом поточних контекстів / факторів, переваги користувача можуть розвиватися в процесі рекомендації. Система повинна визнати, що саме шукає користувач, а не інші варіанти, які він регулярно відвідує, для перегляду. Без розвитку розглянутих уподобань користувачів система може рекомендувати елементи на основі застарілих уподобань користувача.

Відомі різні способи полегшення взаємодії в рекомендаційних системах. Наприклад, присвоєння більш високого пріоритету для нових взаємодій, отримання преференцій з використанням абсолютного та відносного зворотного зв'язку або пари ознак-значень зворотного зв'язку. Однак жоден з них не розглядає контексти як частину реакції користувача, що негативно позначається на персоналізації результатів рекомендацій. Запропоновано підходи для використання соціальних контекстів для ідентифікації інтересів користувачів, таких як місце розташування та запити або різне значення контекстів. Однак вони визначають важливість контексту до покоління рекомендаційних систем з поправками і припускають, що перевага користувачів не змінюється. Робота над цими елементами системи не може відповідати системній меті, оскільки контексти кожного користувача можуть розвиватися під час рекомендації. Спроекуємо інтерактивну рекомендаційну систему з контекстної обізнаності, яка включає контексти під час взаємодії з

користувачами. Визначатимемо контексти як наміри, що лежать в основі покупок або споживання користувачем предметів. Ці контексти можна дізнатися з декларативного огляду користувача на елементи, неявних прихованих факторів користувача та елемента, або явно викладені на основі поточної ситуації користувача. Рекомендаційна система інтерактивно змушує користувачів розкривати свої останні переваги, задаючи питання, що представляє їх попередні транзакції та розгляд контексту. Зокрема, спочатку пропонуватимемо персоналізовану рекомендацію контекстної обізнаності, засновану на матричній факторизації, яка називається персоналізованою зваженою контекстом факторизацією усвідомлень користувача щодо уподобань.

В проєктованій рекомендаційній системі закладемо можливість вивчати індивідуальну важливість і пріоритет контекстів для користувачів на основі їх попередніх транзакцій. Потім пропонуємо втілити ряд стратегій вибору елементів, які ефективно вивчають переваги користувача, вибираючи найбільш релевантне питання в кожній новій взаємодії. На завершення проєктування здійснюватимемо контекстне навчання, щоб адаптувати модель з кожною відповіддю користувача. Розширюючи попередню інтерактивну рекомендацію, розширюватимемо модель, вирішуючи проблему надмірності, яка може статися під час вибору питання. Тут, з огляду на набір елементів розміру, мета полягає в тому, щоб знайти релевантне питання, щоб задати, для того щоб зменшити невизначеність поточних уподобань користувача, запобігаючи надмірності інформації серед обраного пункту в кожному новому інтерактивному пошуку. Ця проблема важлива в інтерактивних системах рекомендацій, оскільки чим швидше ми отримаємо точні налаштування користувачів, тим менше ймовірність того, що користувач втомиться давати зворотний зв'язок. Порівняно з двома попередніми проєктованими рекомендаційними системами в цій при проєктуванні зроблено кілька нових доповнень: вибір пункту є NP-повною задачею, тому пропонується нова стратегія вибору жадібних алгоритмів, яка працює в лінійний час; пропонується новий метод створення рекомендаційних систем з новою стратегією вибору пункту, щоб ефективно прогнозувати найбільш релевантну

рекомендацію для цільового користувача в кожній взаємодії; проводимо всі експерименти над наборами даних, включаючи раніше використані два набори даних і п'ять нових великих наборів даних, для перевірки продуктивності запропонованого рішення в обробці великих, рідкісних даних і користувачів; проводимо нові експерименти, щоб порівняти наші запропоновані стратегії вибору питань з базовою базою на популярності елементів, порівняти персоналізовану модель рекомендацій з двома попередніми рекомендаційними системами з розділу 2 на основі нейронних мереж, і порівняти проєктовану систему з недавньою інтерактивною рекомендацією; перевірятимемо ефективність запропонованого рішення з точки зору складності часу та простору; проведитимемо дослідження ефективності для оцінки впливу кожного основного компонента в моделі, включаючи контекст персоналізації та стратегії вибору питань. Архітектура запропонованої інтерактивної контекстно-обізнаної рекомендаційної системи зображена на рис. 3.1.

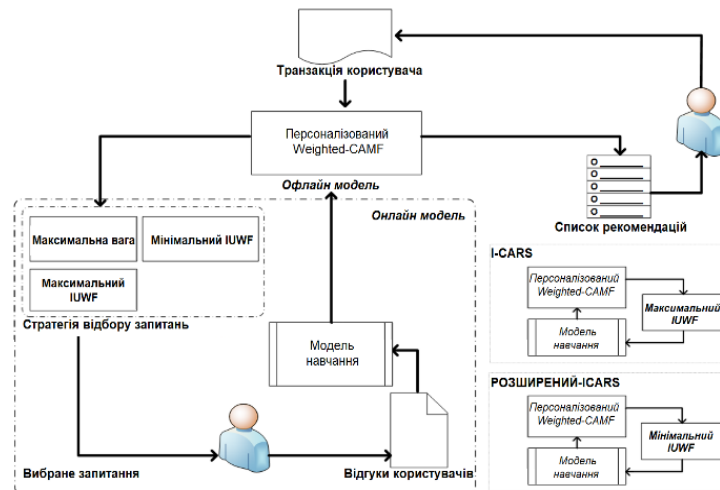


Рисунок 3.1 – Архітектура запропонованої інтерактивної контекстно-обізнаної рекомендаційної системи

Розроблений метод створення рекомендаційних систем має на меті забезпечити механізм взаємодії для користувачів, щоб представити свої поточні потреби, тоді система буде включати в себе їх прямиий зворотний зв'язок від взаємодії для уточнення моделювання уподобань користувача.

Рекомендаційна система повертатиме найбільш актуальну рекомендацію для задоволення своїх поточних потреб. У ній є три компоненти:

1. Модель вивчає переваги користувачів з попередньої транзакції користувача, полегшуючи різне значення та пріоритет контекстів та прихованих факторів у остаточній рекомендації.

2. Вибір питань направляє користувача, щоб уточнити її недавній інтерес, задаючи питання під час взаємодії. Питання вибирається в терміні елемента, який, швидше за все, представляє попередній інтерес користувача.

3. Контекстне навчання моделює відгуки користувачів із поставленого питання для уточнення уподобань користувачів, прагнучи повернути кращий набір рекомендацій для задоволення потреб користувачів. Запуск користувачем питання і модель оновлення зворотного зв'язку здійснюється послідовно під час взаємодії з користувачем.

Огляд запропонованого алгоритму на високому рівні можна представити так:

1. Ініціалізувати параметри та дізнатися параметри контексту користувача за допомогою офлайн-моделі.

2. Спілкування з користувачами, щоб перевизначити свої поточні інтереси, задаючи кілька запитань. Питання вибирається в терміні елемента, який, швидше за все, представляє попередні інтереси користувача.

3. Для інтерактивної рекомендаційної системи вибиратимемо питання, яке потрібно задати.

Крок 2 представляє взаємодію з користувачами. Система буде оцінювати прямий зворотний зв'язок від взаємодії для уточнення моделювання уподобань користувача. Нарешті, рекомендаційна система повертає найбільш релевантну рекомендацію для задоволення поточних потреб користувача. Перш, ніж детально описувати кожен елемент інтерактивна рекомендаційна система визначатиме визначення контекстів.

Таким чином, визначаємо контекст як намір користувача щодо діяльності з придбання товару або споживання товару, які можуть бути отримані з будь-якої інформації, крім оцінок елементів користувача. Різні контексти відіграють важливу

роль у різних ситуаціях. Наприклад, стан руху або настроїв водія можуть бути важливими контекстами, коли в автомобілі пропонується відтворення музичної доріжки. З огляду на набір контекстів і оцінок користувачьких елементів, для створення вектора ваги контекстів для користувача, спочатку дізнаємося його прихованих факторів користувача, а потім обчислюємо середній розмір всіх користувачів прихованих факторів, які вказують на важливість контекстів для них. Потім для кожного елемента в його історії представляємо контексти цього елемента як контекст-вектор, а потім подаємо в модель. Вплив кожного контексту для неї досягається шляхом множення вагового вектора на контекстний вектор. Припустимо, у нас є набір контекстів і оцінки користувачів, тоді можна створити вектор ваги користувача та його впливу на кожен контекст.

3.2 Метод створення інтерактивних рекомендаційних систем

Користувачі, як правило, надають пріоритет контекстам при виборі онлайн-елементів або наданні зворотного зв'язку під час взаємодії. При цьому розглядають персоналізацію контекстів для кожного користувача в попередньому підході з фільтрації. Однак він припускає, що після виявлення контексти рівномірно сприяють генерації остаточних рекомендацій. Тому, видаляємо це припущення за допомогою контекстного підходу моделювання. Контекстна інформація, зокрема, включена в алгоритми, що генерують рекомендації.

Зокрема, на основі матриці факторизації визначаємо контекст персоналізації для кожного користувача, вводячи різні параметри ваги як для контекстної інформації, так і для прихованих факторів користувача. З огляду на користувача і пункт визначаємо домінуючі контексти по відношенню до пунктів, і вивчаємо ваги всіх цих контекстів. Тоді можемо передбачити рейтинг користувача на елементі на основі продукту між вектором ваги контексту та прихованими факторами користувача, а також між вектором ваги контексту та контекстуальним фактором для кожної категорії товару. Формально прогнозований рейтинг користувача на елемент обчислюється. Таким чином,

враховуємо пріоритет кожного користувача з точки зору контекстів при моделюючому їх перевагах. Таким чином, інтерактивна рекомендаційна система намагатиметься звести до мінімуму наступну об'єктивну функцію, мінімізуючи передбачення. Як і більшість рекомендаційних систем, запропонований метод для створення рекомендаційних систем надає їм можливість навчатись на основі історичних даних транзакцій користувачів (наприклад, рейтингів) для створення рекомендацій. Пізніше він періодично оновлює модель, щоб включити нові дані.

Розглянемо варіанти вибору відповідних елементів, щоб сформувані запитання, для того щоб запитати користувачів про їхні відгуки про них. За допомогою цих прямих взаємодій з користувачами інтерактивна рекомендаційна система прагне максимізувати розуміння контекстних уподобань користувачів, задаючи питання, яке зменшить невизначеність поточних уподобань користувачів. З огляду на набір елементів прагнутимемо вибрати оптимальну підмножину елемента фіксованого розміру, яка мінімізує невизначеність уподобань користувача. Завдяки зворотному зв'язку користувача на сформоване питання про товар, отримуємо деяку інформацію, щоб ще більше зрозуміти переваги користувача. Використовуватимемо спочатку дві стратегії, щоб знайти елементи, для того щоб запитати користувачів. З огляду на набір всіх кандидатів, за цими стратегіями виконують вибір за двома кроками алгоритму: обчисліть значення елемента, щодо решти набору; порівняти вагу двох кандидатів і видалити мінімальне значення ваги. Обидва етапи проводяться рекурсивно, поки не буде знайдено один елемент з максимальним отриманням інформації. Це відповідно описано так:

1. Стратегія відбору: Максимальні тригери коефіцієнта ваги елемента користувача відгуки користувачів, вибравши елемент. Це відноситься до найважливішого контексту та прихованих факторів для користувача. Оскільки оновлюємо з градієнтом крок, то також це полегшує в контексті цей параметр. Максимізація намагається привести до балансу важливість контексту з векторами користувача та елемента і розраховується. Інтерактивна

рекомендаційна система замовляє питання, щоб задати користувачеві від найвищого до найнижчого контексту і факторів, важливих від попередньої транзакції користувача. Розставляючи пріоритети в усіх цих факторах при кожній взаємодії з користувачами, інтерактивна рекомендаційна система буде більш ефективно вивчати свої поточні переваги.

2. Максимальна вага стратегії відбору дозволить досягти вибору питання максимальної ваги контексту. Стратегія опирається тільки на пріоритет. Параметр надає змогу вибирати пункт до триггеру користувача для відповіді. Максимальна вага вибору є стратегія, яка визначена як сума ваг над усіма прихованими факторами користувача. Ця стратегія дає можливість вивчити і визначити пріоритети відповідного користувача і пункту латентних факторів, заснованих на відповіді користувача.

Питання вибору стратегії включають контексти під час вибору питання, і можуть задати те ж питання користувачеві, але в іншому контексті. Коли запитується елемент, то припускаємо, що користувач надасть зворотний зв'язок з системою, тоді, таким чином, використовуємо це для моделювання своїх уподобань. Однак, хоча інформаційний внесок від кожного елемента, як правило, відрізняється, то він, ймовірно, буде зайвим щодо декількох пунктів (питань), які потрібно задати. Для вирішення цієї проблеми пропонуємо спільну концепцію питання (пункту) внеску для вимірювання інформаційного внеску, що ставить кожне питання, де є набором предметів, які потрібно запитати; і є пунктом, який потрібно додати. Загальна схожість між усіма пунктами множини визначатиметься двохкомпонентним вектором. З огляду на набір всіх кандидатів пунктів і кількість питань, які треба задати, прагнутимемо вибрати оптимальну підмножину. Це кількість питань. Пошук оптимальної підмножини питання є складною проблемою NP-повною. Таким чином, оптимальний вибір питання є NP-повним. Оскільки оптимальний вибір елемента підмножини не може бути досягнутий, вирішуватимемо цю проблему за допомогою жадібних стратегій, які вибирають елементи, які потрібно запитати, послідовно вивчивши їх у наступних кроках: визначити, який

елемент почати досліджувати для формування питання, і покласти його в поточний вибраний набір; обчислювати внесок питання для кожного з решти пункту; вибрати елемент, який має найменшу цінність внеску питання, щоб додати його; кроки 2-3 запускаються рекурсивно, поки не залишиться предметів або відповідатиме максимальній кількості питань. На кроці 1 використовуватимемо обчислення максимального значення, щоб вибрати перший елемент, який треба запитати.

Контекстне навчання спрямоване на полегшення безперервного вивчення уподобань користувачів шляхом включення відповіді користувача в рекомендації. Воно визначає, подобається користувачу або не подобається елемент відповідно. Цей підхід не включає контексти як у взаємодії, так і в моделі. Нагадуючи про важливість контекстів у рекомендації, включатимемо контексти як у виборі, так і в моделюючому навчанні. Оскільки рекомендаційна система включає відгуки користувачів, тим компактніше система розуміє переваги останнього користувача. Зокрема, контексти розглядаються при виборі питання, щоб задати користувачам. З огляду на відповідь користувача в форматі чотири компонентного вектора, це нове спостереження буде включено в модель, і параметри активного користувача будуть оброблені. Відповідь користувача безпосередньо не відштовхує ставлення користувача до контекстів або елемента, але відповідає багатьом невідомим факторам, які сприяють вибору елемента. На відміну від існуючої літератури з бінарними рейтингами, дані мають рейтинги за шкалою 1-5. Ці оцінки 1-5 шкал дозволяють рецензентам висловлювати свою думку з пунктів у 5 варіантах («відмінно», «добре», «добре», «погано» або «незадовільно») замість лише позитивних або негативних, як у бінарних рейтингах. Таким чином, оскільки користувачі висловлюють свою думку в оцінках 1-5 шкали, потрібно визначити рейтинг зворотного зв'язку нового користувача. Новий рейтинг відгуків користувачів повинен відображати тенденції користувача або елемента в наданні або отриманні більш високих / нижчих ставок, ніж інші, як це було. Після відповідного опрацювання, визначатимемо, що користувачеві подобається елемент, коли він

оцінює його вище 3 з 5 рейтингової шкали, і відповідь на запитаний елемент залежить від існування елемента в наборі даних групової істини користувача. Зокрема, коли запитаний елемент знаходиться в основі істини користувача, рейтинг зворотного зв'язку поверне середню оцінку всіх елементів, які раніше були оцінені користувачем як вищі за 3, або поверне середню негативну оцінку.

Визначатимемо фіксований максимум взаємодій між користувачами та системою, перш ніж користувачі втомлюються давати зворотний зв'язок. Це здійснюватимемо з використанням окремого алгоритму, що уточнюватиме моделювання уподобань користувача, коли з'являється новий зворотний зв'язок від кожного користувача взаємодії. Зокрема, зворотній зв'язок взаємодії з користувачем адаптовані до моделі шляхом оновлення відповідного користувача чотири параметри. Оскільки система постійно перевіряє відповідні параметри, змінюється також ступінь міри в користувача, елемент і контекст. Ці зміни впливають на питання що воля бути вибрані під час наступної взаємодії. Коли взаємодії завершені, елементи з найвищим рейтингом генеруються на основі останньої моделі. З огляду на модель навчання та контекст навчання, час складності для такої інтерактивної рекомендаційної моделі є задовільним і може бути прийнятним. Вивчення моделі здійснюється шляхом оновлення користувача, елемента та контекстуальних параметрів після кожної взаємодії, де більшості налаштувань є невеликим числом, порівняно з кількістю користувачів та елементів. Оскільки система включає відгуки користувачів, тим компактніше система розуміє переваги останнього користувача.

Розглянемо ефективність спроектовано інтерактивної рекомендаційної системи проводячи комплексні експерименти над реальними наборами даних. Метою експериментів є прагнення дослідити вплив контекстної персоналізації на якість рекомендацій у застосунках електронної комерції. Крім того, набір використовуваних даних є типовим набором даних, що використовується в інтерактивних системах рекомендацій. У різних контекстних ситуаціях використовувався набір даних, який збирається для оцінки впливу контексту на уподобання. Також, ті які містять багаті метадані, такі як назва, категорії та

бренд. Дотримуючись практики, щоб уникнути перенавчання ділимо набір даних на набір даних для навчання, тестування та зворотного зв'язку. Зокрема, ми сортуємо дані на основі часу транзакції, використовуючи перші 70% як навчальний набір, наступні 15% як набір тестів і останні 15% як набір даних зворотного зв'язку. Навчальний набір даних використовується моделлю для вивчення уподобань користувачів на основі їх історичних даних. Набір даних зворотного зв'язку розглядається як нові дії від користувача до системи, в той час як тестовий набір даних використовується для оцінки продуктивності моделі. Результат отримується з набору тестових даних, де рейтинг вище 3 для наборів даних з 5 різними оцінками. Для бінарного набору даних елементи зі значенням оцінки 1 вважаються істиною. Приймаючи стратегію експерименту гарантуватимемо, що для кожного користувача існує хоча б один позитивний рейтинг. Проводимо експерименти з метою відповіді на наступні дослідження:

1. Запропонована офлайн-модель інтерактивної рекомендаційної системи порівняно традиційну спільну фільтрацію включає глибоке навчання та контекстну рекомендацію щодо більшого та розрідженого набору даних матиме кращі показники за результатами використання.

2. Запропонована офлайн-модель інтерактивної рекомендаційної системи підвищує ефективність рекомендації як у малому, так і в більшому наборі даних.

3. Запропонована офлайн-модель інтерактивної рекомендаційної системи враховує впливи різних стратегій вибору питань.

4. Контекст навчання з відгуків кожного користувача може збільшити ефективність інтерактивної рекомендаційної системи у поверненні рекомендації верхнього пункту.

5. Змінюються поточні контекстні параметри користувача під час процесу рекомендації.

6. Запропонована офлайн-модель інтерактивної рекомендаційної системи є чутливою до різного ступеня розрідженості даних.

7. Запропонована офлайн-модель інтерактивної рекомендаційної системи

виконується на холодному запуску користувача.

8. Наскільки ефективна інтерактивна рекомендаційна система з точки зору складності часу та простору порівняно з іншими інтерактивними / розмовними системами рекомендацій.

9. Вплив основних компонентів моделі, таких як контекстна персоналізація, взаємодія з користувачами та стратегія вибору питань, на ефективність моделі.

На результат впливатимуть такі події та особливості:

1. Використання подібності споживання товарів користувачами для рекомендації.

2. Використання як рейтингової інформації, так і інших неявних даних для рекомендації.

3. Архітектура нейронної мережі для спільної фільтрації, яка поєднує в собі лінійність і нелінійність глибоких нейронних мереж для моделювання латентних структур користувача-елемента.

4. Метод спільної фільтрації, який використовує ідею автоматичного кодера для реконструкції повних налаштувань елемента користувача від спостережуваних дій між елементами користувача.

5. Зважений результат дозволяє різним вагам прихованих факторів впливати на остаточні рейтинги.

6. Моделювання взаємодії контекстних факторів з оцінками елементів.

7. Використання певних функцій продукту в інтересах користувача та прихованих функцій, отриманих для рекомендацій.

8. Визначення найкращих запитань, які потрібно задати, щоб швидко вивчити налаштування користувачів.

9. Зважування безперешкодних взаємодій неоднорідно з популярністю елемента і миттєво оновлює модель з новим зворотним зв'язком.

10. Метод рекомендації на основі питань, який допомагає користувачеві знаходити елементи в інтерактивному режимі.

Таким чином, розроблено метод створення інтерактивної рекомендаційної системи та на його основі систему. Проведеними експериментами доведено

ефективність запропонованих рішень.

3.3 Висновки до третього розділу

В результаті запропоновано інтерактивну рекомендаційну систему з контекстною обізнаністю, яка ітераційно вивчає переваги користувачів за допомогою зворотного зв'язку, наданого під час взаємодії з системою. Спочатку пропонується персоналізована контекстна система рекомендацій для використання в якості методу навчання. Потім, крім того, пропонується нова інтерактивна рекомендаційна система з механізмом зворотного зв'язку. Щоб викликати зворотний зв'язок користувачів, пропонуються стратегії вибору питань, щоб задати користувачеві під час взаємодії, а потім включити відповідь в систему. Для перевірки інтерактивної рекомендаційної системи було здійснено оцінювання ефективності методу в різних метриках до і після кожної нової взаємодії. Результати підтверджують, що розроблений метод та система перевершує аналогічні системи.

4 ДВОНАПРАВЛЕНА ПОСЛІДОВНА РЕКОМЕНДАЦІЙНА СИСТЕМА З КОНТЕКСТНОЇ ОБІЗНАНОСТІ ДАНИХ

4.1 Архітектура та метод створення двонаправленої послідовної рекомендаційної системи

Платформи електронної комерції стали інструментом для клієнтів для проведення онлайн-заходів, таких як онлайн-покупки, бронювання квитків і навіть освіта. У реальному світі онлайн-діяльність користувачів зазвичай відбувається послідовно, замість того, щоб формувати ряд паралельних подій. Наприклад, перед початком відпочинку за кордоном користувач може спочатку придбати авіаквитки, забронювати готель, купити квиток в парк розваг, замовити таксі і купити сувеніри в кінці свята. У цьому випадку всі ці дії послідовно залежать, навряд чи хтось бронює таксі перед покупкою авіаквитка. Ці сценарії застосування підвищили попит на ефективні рекомендаційні системи. У відповідь на цю вимогу деякі платформи електронної комерції, такі як Alibaba і eBay, створили свої послідовні системи рекомендацій, які дозволяють рекомендацію наступного пункту на основі послідовної поведінки користувачів в Інтернеті. На практиці налаштування користувачів, як правило, чутливі до контексту і можуть змінюватися з часом. Це вимагає розробки контекст-обізнаних послідовних рекомендаційних підходів. Розглядатимемо ефективні рішення для послідовної рекомендації в режимі реального часу на платформах онлайн-сервісів. Послідовна рекомендація, що усвідомлює контекст, спрямована на використання послідовних залежностей щодо взаємодії між користувачем і елементом і контекстом, які впливають на ці взаємодії, і повернення наступного елемента з найкращим значенням поточних уподобань цільового користувача. За винятком загальних характеристик контекстно-свідомих рекомендаційних систем, таких як контекстно-чутливі переваги історії користувачів, послідовний рекомендувач контексту має особливі вимоги динамізму щодо послідовної поведінки користувача, контекстів користувачів та реакції в режимі реального часу на цю динаміку. Природно, ідеальний підхід до контекстно-обізнаних послідовних рекомендацій повинен бути

чутливим до послідовних уподобань користувачів і вміти відображати послідовні зміни уподобань в рекомендації наступної дії в режимі реального часу. Але, багато рішень, які підходять для рекомендацій щодо елементів, що знають контекст, не застосовується до сценарію послідовних рекомендацій через відсутність визнання змін поведінки користувачів у послідовній діяльності. Основна увага полягає в тому, щоб зафіксувати динамічні послідовні значення статичних контекстів, які повинні бути визначені, такі як часовий інтервал, місце розташування та час доби. Вони не підходять для ситуацій, коли важливі контексти можуть динамічно змінюватися. Відрізняється від існуючих робіт, запропонована проєктована послідовна рекомендаційна система, що усвідомлює контекст, адаптивно фіксує послідовну контекстну інформацію з метаданих елемента та відгуків користувача, які самостійно описуються з транзакцій користувачів.

Основною проблемою існуючої системи послідовної рекомендаційної системи для контексту є динаміка контекстної інформації, яка може вплинути на поведінку користувача. Особливо важливий час транзакції в рекомендації. Як тільки користувачі проведуть транзакцію в різних періодах, ці моделі більше не зможуть представляти переваги користувача. Покладатися на припущення, що всі транзакції, які існують більш, ніж 30 днів будуть мати однакові тимчасові контексти, є неправильним. Це припущення ігнорує важливість і динаміку впливу контекстів у кожній транзакції. Після того, як одна з 30-денних транзакцій знову відбудеться в останній історії діяльності користувачів, модель буде прогнозувати наступний елемент на основі контекстів, які закріплені для історії операцій, проведених 30 днів тому, що призводить до низької якості переваги на наступний пункт. Практична система послідовної рекомендаційної системи, що усвідомлює контекст, повинна бути в змозі добре розпізнавати динаміку поведінки користувачів над послідовністю діяльності та динамікою контекстної інформації, що стоїть за цими послідовностями. Існуючі підходи, що рекомендують послідовні елементи, які можна розділити на три типи: моделі послідовностей; латентні моделі представлення; моделі глибоких нейронних мереж. Методи на основі послідовностей моделюють залежності послідовностей між взаємодією

користувача-елемента шляхом майнінгу шаблонів частоти даних або використання марківських моделей. У той час як послідовна рекомендація на основі частоти даних часто має тенденцію рекомендувати тільки популярні предмети, послідовні рекомендації марківського ланцюга добре відображають короткострокові переваги, але не довгострокові. Латентні підходи до представлення вивчають взаємодію користувача і елемента за допомогою матриці або тензорної факторизації або кодування всіх взаємодій в послідовності в прихований простір. Крім своєї простоти та ефективності ці методи спираються на припущення, що переваги користувачів мають жорсткий порядок послідовності. Глибокі нейронні мережі, засновані на послідовних рекомендаціях, здатні моделювати всебічні відносини в послідовних даних. Однак всі підходи, засновані на відомих моделях нейронних мереж, припускають, що послідовна поведінка користувача є жорсткою і може бути змодельованою тільки в одному тимчасовому напрямку. Механізм уваги дозволяє моделі захоплювати відповідні деталі при прогнозуванні наступного пункту незалежно від положення. Однак ці підходи ігнорують важливість контекстів у послідовній поведінці користувачів або спираються на припущення, що поведінка послідовності користувача суворо впорядкована.

Тому, враховуючи наведені недоліки відомих методів створення рекомендаційних систем, пропонується розробка двонаправленої послідовної системи рекомендацій, яка ефективно прогнозує переваги наступного елемента цільового користувача. Контекстний графік, який фіксує властиві і різноманітні контексти користувача, спочатку побудований на основі семантичної подібності між контекстами в наборі даних. Потім графік контексту транзакції призначений для прогнозування ймовірності транзакції, що відбувається в кожному контексті. Крім того, механізм застосовується до контекстного графіка та графіка контексту транзакції, щоб забезпечити вивчення двонаправленої послідовної поведінки користувачів для більш гнучкої рекомендації наступного пункту. Також, включатимемо стратегію динамічного оновлення моделі, щоб модель відображала зміни даних на інтернет-сайтах в

режимі реального часу.

Кроки методу створення двонаправлених рекомендаційних систем:

1. Створити двонаправлену контекстно-свідому послідовну рекомендаційну систему, яка використовує глобальну та місцеву контекстну інформацію для прогнозування наступного пункту двонаправленими способами. Хоча локальна контекстна інформація може передбачати наступний елемент, який відповідає поведінці користувача, глобальні контексти можуть визначити пов'язані елементи, які відповідають притаманним цьому користувачеві інтересам.

2. Створити новий контекстний графік та новий графік контексту транзакцій. Він фіксує актуальність між контекстами та цільовими уподобаннями користувачів, тоді як попередній зменшує ймовірність транзакції в різних контекстах. Використовуючи ці графіки, динаміка поведінки користувача і контекстів може бути захоплена з метаданих елемента, відгуків користувачів і самоописаних транзакцій користувачів, що є надійним для динамічних змін контексту.

3. Використовуємо стратегію оновлення моделі на основі структури кластера користувача, яка адаптивно оновлює модель глибокої нейронної мережі. Модель оновлюється лише тоді, коли змінюється структура кластера. Це дає змогу відображати поточні послідовні налаштування користувача та гарантує оновлення моделі в режимі реального часу.

4. Проводимо експерименти над трьома реальними наборами даних для перевірки ефективності запропонованого рішення з точки зору різних показників ранжирування та проводимо дослідження для аналізу внесків ключових компонентів у запропонованій моделі. Результати експериментів використовуємо для здійснення самонавчання нейронної мережі.

Згідно розроблено методу створення двонаправлених рекомендаційних систем розроблено архітектуру такої системи, яка зображена на рис. 4.1.

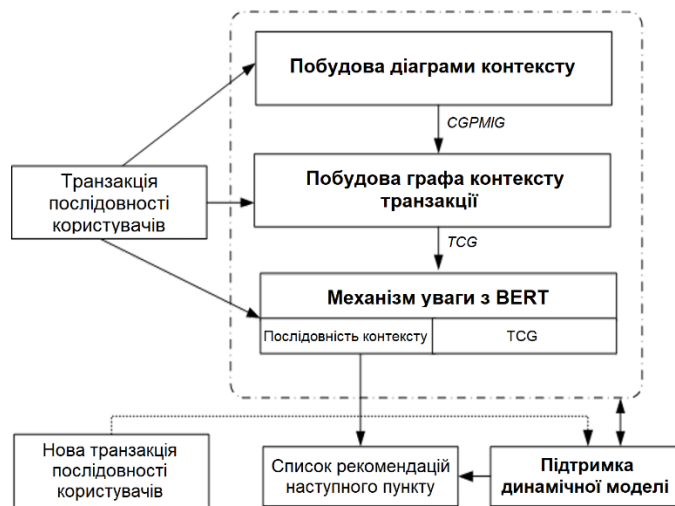


Рисунок 4.1 - Архітектура системи

У цій системі наявна послідовність історичних транзакцій від цільового користувача, а саме у форматі якого цільовий користувач споживає елемент і дані контекстів або вмісту елемента. З огляду на послідовність транзакцій цільового користувача, система повертає список елементів, які мають найвищі ймовірності споживання користувачем на наступному кроці. Система складається з чотирьох важливих частин. По-перше, глобальний зв'язок між контекстами фіксується за допомогою графіка. Потім фіксується ймовірність транзакції, що відбувається в кожному контексті. Після цього система дізнається про представлення поведінки локального користувача та поведінки глобального користувача за допомогою механізму самоцілі, щоб передбачити список наступних пунктів. Нарешті, враховуючи оновлення динаміки на інтернет-сайтах, поступово оновлюється модель, щоб відобразити переваги нового послідовного користувача.

Таким чином, розроблений метод створення двонаправлених контекстно-обізнаних послідовних рекомендаційних систем надав змогу розробити архітектуру такої системи. В подальшому необхідні представлення її компонентів.

4.2 Розробка, реалізація та оцінка прототипу рекомендаційної системи

Розглянемо компоненти двонаправлених контекстно-обізнаних послідовних рекомендаційних систем, представлених її архітектурою на рис. 4.1.

Контексти користувачів є динамічними і можуть змінюватися з плином часу, що призводить до динаміки послідовної поведінки користувачів. Через ці характеристики заздалегідь визначені статичні контексти, такі як день тижня або проміжок часу транзакцій, не будуть точно відображати реальні контекстні переваги користувачів. Краще уявлення про послідовні контексти реального користувача та його поведінку потрібно для високоякісної послідовної рекомендації, обізнаної з контекстом. Інтуїтивно рецензії користувачів і метадані елемента, як правило, відображають послідовні контексти та поведінку користувачів. Наприклад, коли користувачі переглядають фільм, їх відгуки можуть вказувати на те, що одному користувачеві подобається конкретний актор, сюжетна лінія, щасливий кінець і музика, а іншому подобається приємний кінець, звукова лінія і сюжет. Ця інформація буде розглядатися як відповідні контексти для відповідного користувача. Порівняно з заздалегідь визначеними статичними контекстами, ці відгуки вказують на переваги користувача більш чітко. Однак покладатися виключно на відгуки, щоб обмежити контекстні переваги користувача, недостатньо. Потрібно, також, визначити поведінку зв'язку цих контекстів, визначивши зв'язки між ними. Наприклад, «сюжетна лінія» є важливим контекстом у перегляді користувачем, модель повинна бути в змозі визначити сюжет, розповідь, контур і сценарій як важливі контексти також. На основі цієї інтуїції будемо контекстний графік, щоб дізнатися зв'язки між контекстами в зборі даних, щоб забезпечити всебічне і ефективне вивчення поведінки користувачів.

Розглянемо вилучення контексту. Як зазначалося раніше, контексти вказують на аспекти, за якими користувач проводить транзакцію, а також поведінку або наміри користувача щодо цієї транзакції. Різні користувачі можуть мати різні контексти під час каналювання певної транзакції. Крім того, користувач може проводити різні транзакції в ряді фіксованих контекстів. Враховуючи унікальність

контекстів між користувачами та їх динаміку, адаптивно персоналізуємо контексти для певного збору даних. З огляду на набір даних, витягуємо контексти транзакцій користувачів, аналізуючи всі метадані елемента, такі як заголовок і жанр, і всі коментарі всіх транзакцій на елементах цього набору даних. Процес вилучення контексту виконується наступними кроками. По-перше, очищаємо всі коментарі та метадані, видаляючи спеціальні символи, такі як розділові знаки та всі слова, такі як слова «це», «були» та «маючи». Потім всі очищені коментарі та метадані розбиваються на слова за допомогою токен аналізатора. Нарешті, всі ці слова утворюють набір контекстів, які використовуються для побудови контекстного графа. Прагнутимемо ефективно прогнозувати наступні елементи з найвищими ймовірностями, рекомендованими цільовому користувачеві. Для цього потрібно визначити всі елементи, які мають контексти, що мають відношення до поточних контекстів цільового користувача. Щоб досягти цього, потрібно побудувати модель, яка фіксує поведінку зв'язок всіх контекстів, витягнутих з даного збору даних. Очевидно, що простий набір незалежних контекстів не може надати жодної інформації про поведінку контекстів. Тому, пропонується модель контекстного – групового точкового взаємного інформаційного графа над набором витягнутих контекстів. Кожен вузол в нашому випадку є контекст, а край між двома контекстами вказує на кореляцію між ними. Для того, щоб добувати кореляцію між контекстами, простий спосіб полягає в тому, щоб обчислити слово двох пов'язаних контекстів. Однак спільне виникнення не фіксує семантичну схожість між контекстами. Наприклад, спів і конкурс не мають значення, але неправильно визначені як релевантна, оскільки вони часто з'являються як контекстна пара в транзакціях. Тому, пропонуємо згруповану нормалізовану точкову взаємну інформацію для обчислення країв між контекстами. Всі контексти в зборі даних згруповані в кілька груп. Кожна група включає в себе набір семантично пов'язаних контекстів. Усі контексти в групі будуть взаємозамінними в обчисленні країв.

З огляду на набір транзакцій в наборі даних, перетворюємо кожну транзакцію в вектор. Потім всі ці транзакції кластеризуються за допомогою алгоритму K-Means на основі відстані між кожною транзакцією. З огляду на два контексти, межа між

ними розраховується на основі ймовірності появи контекстної пари в 1 і тих же транзакціях і ймовірності появи цієї контекстної пари в 1 і тих же кластерах. Система включає в себе семантичну схожість між контекстами, що дає більш пояснювану інформацію про контекстове виникнення. Використовуючи спроектовану рекомендаційну систему, спів і конкурс в прикладі визначаються як неактуальні на основі їх семантичної подібності. В основі система закладено алгоритм щодо пари контекстів, які з'являються в загальному скупченні, як правило, мають схожість до певної міри. Контекстна пара в невеликій кількості загальних кластерів вказує на високу схожість їх, в той час як пара контекстів у великій кількості загальних кластерів менш впевнені, щоб бути схожими через низьку їх кластеризацію. Значення діапазону становить $[-1,1]$. Позитивне значення має на увазі високу семантичну кореляцію між контекстами, в той час як від'ємне значення вказує на мало або взагалі відсутню семантичну кореляцію. У нашій роботі створюється край для з'єднання двох контекстів, коли значення цих контекстів більше 0. За допомогою контекстного графа можна захопити всі контексти, які мають відношення до поточних контекстів цільового користувача. На рисунку 4.2 зображено приклад 1 контекстного графіка. Тут кожен вузол є контекстом, а вага між двома вузлами - це значення між ними, що вказує на ймовірність того, що ці два вузли будуть актуальними. Наприклад, контексти розповіді і сюжету мають ймовірність релевантності 0,95.

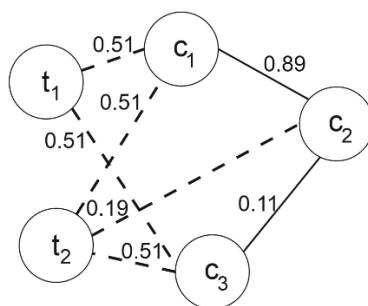


Рисунок 4.2 - Приклад графіка 1

Використовуючи запропонований метод можемо представляти відносини між контекстами. Однак, щоб захопити контекстні переваги користувача,

необхідні додаткові послідовні контексти користувача, щоб визнати ситуацію, в якій проводиться транзакція, але не досяжні лише за допомогою контекстного графа. Ідеальна модель повинна бути в змозі охопити як властиві загальні контексти користувача, так і її миттєві контексти щодо конкретної транзакції. Таким чином, для вирішення цієї проблеми включаємо всі транзакції користувачів у певному зборі даних у контекстний графік, щоб сформувати розширений графік, який називають графіком контексту транзакцій, який побудований нейромережею. Щоб переконатися, що отримаємо однаковий формат введення, використовуємо токенизатор для вилучення контекстів з кожної транзакції користувача, застосовуючи кроки вилучення контексту. Потім витягнуті контексти з транзакцій використовуються для створення пропозиції.

Будуємо графік контексту транзакцій, щоб зафіксувати зв'язки кожної транзакції та всі контексти даного збору даних. В системі є алгоритм, який дає доступ до ще одного шару моделі глибокої нейронної мережі. Вихід його необхідний для прогнозування транзакцій, що відбуваються. Розширюючи графік, тобто двошарову мережеву структуру, включаємо два типи вузлів і два типи ребер. Нижній шар - це графік, де вузол є контекстом, а край вказує на відношення між двома різними контекстами. Верхній шар - це двопартійний графік, де вузол є контекстом або транзакцією. Край у верхньому шарі представляє зв'язок між транзакцією та контекстом, що вказує на ймовірність цієї операції, що відбувається в цьому конкретному контексті. Однак у контекстному графі транзакції потрібно обчислити релевантність між транзакцією, описаним як її ідентифікатор, і контекстом. На рис. 4.3 наведено приклад 2 такого графіка, який має 2 вузли транзакцій t_1 , t_2 і 3 контекстні вузли c_1 , c_2 і c_3 .



Рисунок 4.3 – Приклад графіка 2

Потрібно розробити нову метрику для розрахунку краю. З огляду на вузол транзакції і контекстний вузол, вага краю між ними може бути обчислена з використанням різних показників, таких як коефіцієнт кореляції Пірсона. Кореляція Пірсона має припущення про нормальний розподіл змінних, що може бути не правильно для деяких реальних даних. Порівняно з кореляцією Пірсона, він не має жодних припущень для статистики слів і є ефективним і простим алгоритмом зіставлення слів у запиті до документів, які мають відношення до запиту. З огляду на це, вибираємо його для побудови краю між транзакціями та контекстами. За допомогою нього межа між транзакцією користувача та контекстом визначає, наскільки важливим є контекст для транзакції в зборі даних.

Це дозволяє моделі захопити всі відповідні елементи кандидата, які будуть евакуюватися для наступного пункту прогнозування. Вивсд цього шару є сукупність n -мірних векторів транзакцій користувача, де кожне значення виміру вектору показує важливість відповідного контексту в цій транзакції користувача. Ці вектори транзакцій подаються в шар самооцінки для наступного елемента ідентифікації. Використовуючи спроектовану систему можна представити всі відповідні контексти для кожного контексту в транзакціях користувача. Однак модель розглядає всі контексти як набір контекстів, який ігнорує важливість контекстного порядку в контекстних параметрах користувача. Порядок контекстів важливий для розуміння

поведінки користувача, оскільки користувач зазвичай проводить діяльність послідовно. Наприклад, користувач, який хоче відпочити за кордоном, вважає за краще спочатку заповнити свою туристичну візу перед бронюванням готелю в місті призначення. Для досягнення цієї мети потрібно побудувати модель, яка точно фіксує порядок послідовностей контекстів для всіх транзакцій користувача в зборі даних, одночасно вивчаючи різноманітні пріоритети, пов'язані з кожним контекстом серії. Простими методами представлення та генерації двонаправленого представлення порядку всіх транзакцій у зборі даних є використання двонаправленого списку. Щоб визначити найбільш підходящий підхід до визначення важливості контексту в налаштуваннях користувачів, проводимо набір експериментів з використанням набору даних огляду фільмів для перевірки точності різних функцій. Представлені результати кожного підходу з точки зору точності є більш надійними і точними, ніж у визначенні значення кожного контексту в послідовності, використовуючи шари самої середини двох направленийими способами. Це узгоджується з експериментом, проведеним для визначення значення контексту в послідовності. Таким чином, застосовуємо систему для створення представлення порядку всіх транзакцій і для вивчення важливості кожного контексту в двонаправлених шляхах. Як правило, система отримує пропозиції в документі як вхід. Адаптуємо її до представлення графіків. Включаємо графік і послідовність транзакцій користувачів з трансформатором. Після оцінки уваги для кожного контексту розраховується їх значення. Загальні переваги та миттєві налаштування користувача повністю інтегровані після багатошарової взаємодії в 12-шаровий та 12-головний кодер самооцінки. Використовуємо навчальний оптимізатор моделі та втрату перехресної ентропії для вимірювання продуктивності моделі при проведенні прогнозування.

Модель навчена з урахуванням набору даних і очікується, що вона буде добре працювати на інших нових наборах даних в тому ж домені. Однак динамічні оновлення даних можуть ввести нові параметри користувача. У цій ситуації модель повинна бути скоригована, щоб уникнути значних втрат якості рекомендацій.

Простий підхід до підтримки моделі полягає в перепідготовці моделі кожного разу, коли надійдуть нові дані. Однак це дорого коштує з точки зору часу. Щоб вирішити цю проблему, пропонуємо додаткове оновлення моделі для підтримки якості рекомендації моделі. Використаємо для досягнення мети алгоритм інкрементного оновлення моделі. З огляду на попередньо підготовлену модель, графік контексту транзакцій, контекстну спільну фільтрацію, транзакційно-кластерний центр та набір нових даних в партіях потокового передавання, алгоритм виконуватиме в 3 кроки для поступового оновлення поточної моделі: використовувати тригер для виявлення потреб в оновленнях; оновлення спільної фільтрації та графіку новими даними з використанням попередньо навчених; оновити модель, тренувавши її з новими графіками.

Використовуємо транзакції кластерів інформації, щоб запустити тригер оновлення моделі. По-перше, кластеризуємо нові дані за допомогою K-Means до тієї ж кількості груп, що і при зборі даних, і пізній кожен кластерний центр. Потім обчислюємо відстань нового кластера до найближчого кластера в зборі даних і знаходимо максимальну відстань. Відстань вимірюється на основі відстані K-L. Після цього порівнюємо максимальну відстань до максимального внутрішньо кластерного збору даних, щоб вирішити стан транзакційних кластерів. Нарешті, оновлення моделі проводиться, коли одна з нових відстаней кластера вище, ніж максимальна внутрішньо кластерна. Інтуїція цього додаткового оновлення полягає в тому, що нові дані прибуття, особливо з новим контекстом, можуть принести нову перспективу до поточної моделі. Модель може захопити цю нову інформацію, вбудовуючи нові дані до вхідних даних моделі, графіку та вбудовування контексту. Як тільки буде оновлена система даними і вбудовування контексту, то можемо оновити модель, щоб відобразити нові переваги користувачів.

З огляду на складність часу, яка значною мірою залежить від кількості транзакцій та контекстів, оновлення з меншою кількістю даних зробить навчання швидшим. З оновленням інкрементної моделі навчання проводиться тільки над новими вхідними даними, які потребують набагато нижчої вартості обслуговування через набагато менший розмір навчальних даних у порівнянні з

усім набором даних. Таким чином, використовуючи цю додаткову стратегію, модель може бути добре збережена.

Експерименти проводимо над наборами даних, включаючи набори даних Amazon. Поділимо набір даних на набори даних для навчання, тестування та перевірки. Зокрема, сортуємо рейтинги на основі послідовності часу, використовуємо 80% як навчальний набір, 10% як тестовий набір і 10% як набір даних перевірки. Визначаємо, що користувачеві подобається елемент, коли він оцінює його вище 3 з 5 рейтингової шкали і змінює їх на 1 і 0 в іншому випадку. Порівнюємо запропоновані методи з трьома типовими найсучаснішими базовими лініями, які поділяються на три групи: один класичний метод рекомендації на основі ранжирування; один послідовний метод рекомендації; один двонаправлений послідовний метод рекомендування. Тут бассовий персоналізований рейтинг є класичним методом, який включає неявні відгуки для вивчення персоналізованих рангу. Згортова послідовність вбудовування є популярною послідовною рекомендацією, яка заснована на методі шляхом застосування згорткових операцій на матриці вбудовування. Рекомендаційна система використовує глибоку самооцінку до моделі послідовностей поведінки користувачів, яка вважається найкращою практикою існуючих систем рекомендувачів. Оцінюємо ефективність порівнюваних методів за допомогою нормалізованого дисконтованого кумулятивного прибутку, враховуючи список найкращих рекомендованих елементів і останню дію в послідовностях користувача.

Приймаємо вбудовані вектори контексту та вектори документів з шарів, побудованих в одному навчальному процесі. У дослідженні встановлюємо всі значення параметрів за замовчуванням для цієї роботи, дотримуючись налаштувань. Спеціально використовуємо стек кодера, загальну кількість епох для навчання, розмір партії. Використовується оптимізатор та модель попередньої підготовки. Для базових ліній використовуємо код авторів і добре налаштовуємо параметри для кожного набору даних.

Також, перевіряємо вплив розміру кластера на якість рекомендації. Потім

порівнюємо отримане рішення з найсучаснішими конкурентами. Нарешті, перевіряємо вплив динамічного оновлення моделі на якість рекомендацій. Далі оцінюємо ефективність системи над різними розмірами кластерів і повідомляємо про результати випробувань за трьома контрольними наборами даних. Максимальний розмір отриманих кластерів за допомогою цих наборів даних становить 4, оскільки контекстна інформація цих наборів даних (назва і жанр) досить схожа в різних транзакціях. Враховуючи, що кількість корпусів, які використовуємо в дослідженні, є коротким текстом (назва та жанр елемента), а продуктивність кластера з 4 кластерами значно краща, то використовуємо цей розмір кластера для проведення решти тестів для всіх наборів даних.

Також, перевіряємо ефект стратегії оновлення послідовної моделі, представляючи зміну продуктивності моделі з огляду на новий прихід даних. Зокрема, ділимо набір даних на поїзд, тестування, перевірку та нові дані як 80%, 10%, 5% та 5% відповідно. Для нових потокових даних оцінюємо ефективність запропонованого методу в 5 партіях потокових даних: вони 20%, 40%, 60%, 80% і 100% нових даних. Поступово оновлюємо модель з урахуванням нових даних і використовуємо нову модель для прогнозування тестових даних. Зі збільшенням оновленого розміру даних, точність моделі дещо зменшуються. Порівняння витрат часу для рекомендації показують, що відрізняється в різних методах, незалежно від кількості функцій, включених моделлю.

Запропонована в рекомендаційній системі модель спрямуванням контекст-обізнаної послідовної рекомендації зберегла високу ефективність рекомендації, досягнувши при цьому набагато більш високої ефективності, ніж у існуючих конкурентів. Це продемонструвало перевагу нашого послідовного рішення щодо послідовних рекомендацій перед конкурентами та довело ефективність контекст-обізнаної послідовної рекомендації у динамічному середовищі.

4.3 Висновки до четвертого розділу

У результаті дослідження проектування рекомендаційних систем та розробки методу їх створення за спрямуванням контекст-обізнаної послідовної рекомендації. пропонується новий графік для визначення релевантності контекстів. Потім пропонуємо новий графік для видобутку контекстів, що стоять за тим, що відбувається транзакцією. Крім того, трансформатор застосовується до графіків новим способом, який фіксує послідовну поведінку користувача двонаправленими способами. Також, пропонуємо додаткове оновлення моделі алгоритму для обробки змін уподобань користувача в динамічному середовищі. За результатами проведених досліджень та спроектованої системи проведено експерименти, щоб оцінити ефективність запропонованого, а також провели експерименти для оцінки впливу ключових компонентів моделі. Експериментальні результати показали, що результат перевершує існуючих конкурентів за послідовною рекомендацією.

ВИСНОВКИ

Отже, на основі здійснення аналізу праць вітчизняних та зарубіжних дослідників та проведеного дослідження можна зробити такі висновки.

1. Системи рекомендацій спрямовані на те, щоб допомогти користувачам у виявленні та отриманні певних елементів з великої колекції елементів, які мають відношення до їх уподобань. Переваги користувачів, як правило, вивчаються з їх попередніх транзакцій, тобто для їх ефективного застосування повинен накопичуватись попередній результат. Уподобання користувача важко зрозуміти, оскільки на них впливають різні незрозумілі ситуації або контексти, які існують, коли користувач приймає рішення. Контексти динамічні, і значення кожного контексту кожного користувача змінюється, що призводить до динамізму уподобань користувача.

2. Було доведено, що система рекомендацій з контекстної обізнаності (СРКО) підвищує якість рекомендацій шляхом включення контекстної інформації та персоналізації їх для кожного користувача. Персоналізація контекстів для кожного користувача та включення їх у процес рекомендацій є двома основними задачами СРКО. Було розглянуто ці два питання з метою підвищення продуктивності рекомендаційної системи з точки зору ефективності та продуктивності. Зокрема, залежно від характеристик наявних даних, застосовано три різні підходи для вирішення цих завдань. Перша характеристика даних пов'язана з ситуацією, коли налаштування користувачів доступні через відгуки користувачів про придбані товари. Як правило, ці відгуки описують рейтинг користувача товару. У цьому дослідженні запропоновано використовувати вікна користувача згідно від'ємної матриці (ВКВМ), нову модель для визначення контекстів користувачів з їх заявлених переглядів. ВКВМ персоналізує та визначає найбільш релевантні контексти для кожного користувача. Потім включаємо персоналізовані контекстом у процес генерації рекомендацій. Введемо першу модель - декларативно-контекстну систему рекомендацій (ДКСР). ДКСР може підвищити якість рекомендацій для найбільш активних користувачів за допомогою комплексних

досліджень. Також, введемо другу модель для декларативного користувача з елементом профілювання згідно контекстної обізнаності рекомендацій системи (ДКПКОРС). Модель явних факторів ДКПКОРС, яка включає не тільки профілі користувачів, але і профілі товарів в модель рекомендації має на меті працювати над усіма користувачами, незалежно від того, чи активні вони чи не активні. ДКПКОРС відповідає профілям користувачів і елементів, щоб рекомендувати користувачеві найбільш релевантні елементи.

3. У більшості випадків налаштування користувачів можуть бути взяті з історії активності. Через динамічний характер контекстів у прийнятті рішень, може з'явитися нова характеристика, яка ще не була захоплена, що є другою особливістю даних. З огляду на цю характеристику, розроблено новий метод захоплення користувацьких проміжних переваг, використовуючи взаємодію користувачів із системою. Кожен контекст має різноманітний пріоритет для кожного користувача. Спочатку введемо новий підхід за помилкою середнього значення (СЗ) за персоналізованою зважено-контекстною матрицею факторизації (ПЗКМФ), щоб дізнатися важливість контекстів для кожного користувача. Ми збираємо миттєві налаштування користувачів у верхній частині персоналізованої зважено-контекстною матриці факторизації, використовуючи їх для надання входу в систему, використовуючи один з трьох розглядуваних варіантів. Відповідь користувача повертається в модель за допомогою контекстного навчання, а потім найбільш релевантні елементи повертаються користувачеві. Третя характеристика даних виникає, коли порядок транзакцій впливає на переваги користувача. Рішення приймається в цьому випадку на основі попередніх або майбутніх дій користувача. Тому, пропонується контекстний графік контекстно згрупованої нормованої взаємної точкової інформації (КЗНВТІ) та контексту транзакції (КТ) для послідовного захоплення контекстів користувачів. Зокрема, КЗНВТІ моделює взаємозв'язок між контекстами, що дозволяє нам видобувати всі відповідні контексти в даному контексті. Контекст транзакції моделює важливість контекстів у кожній транзакції користувача, фіксуючи ситуації, в яких користувач транзакцій. Включаємо вивчені переваги користувача в систему рекомендацій за допомогою

двох направлених представлень кодерів трансформерів (ДПКТ). Це надає змогу розглядати декларативні контексти, в яких користувачі проводять транзакції, особливо з точки зору відгуків користувачів та / або попередньо визначених неявних або явних текстів.

4. Пропонується три нові рекомендаційні підходи, які базуються на характеристиках наявних даних. Результати експериментів показують, що запропоновані підходи можуть вирішити цілі запропонованих рішень і покращити існуючі аналоги конкурентів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Joseph Johnson. Worldwide digital population. URL: <https://Worldwidedigitalpopulation>.
2. ScrapeHero. How many products does amazon sell worldwide. URL: <https://www.scrapehero.com/how-many-products-does-amazon-sell-worldwide-october-2017/>.
3. Ian MacKenzie, Chris Meyer, and Steve Noble. How retailers can keep up with consumers. URL: <https://www.mckinsey.com/industries/retail/our-insights/how-retailers-can-keep-up-with-consumers>.
4. The Deloitte Consumer Review. Made-to-order: The rise of mass personalisation. URL: <https://www2.deloitte.com/ch/en/pages/consumer-business/articles/made-to-order-the-rise-of-mass-personalisation.html>.
5. Weihua Yuan, Hong Wang, Xiaomei Yu, Nan Liu, and Zhenghao Li. Attention-based context-aware sequential recommendation model. *Information Sciences*, 2020. 122–134p.
6. Lakshmanan Rakkappan and Vaibhav Rajan. Context-aware sequential recommendations with stacked recurrent neural networks. In *The World Wide Web Conference*, 2019. 3172–3178p.
7. Mehrdad Jalali, Norwati Mustapha, Md Nasir Sulaiman, and Ali Mamat. Webpum: A web-based recommendation system to predict user future movements. *Expert Systems with Applications*, 2010 37(9):6201–6212.
8. Dong Qin, Xiangmin Zhou, Lei Chen, Guangyan Huang, and Yanchun Zhang. Dynamic connection-based social group recommendation. 2020. volume 32, pages 453–467.
9. Alejandro Bellogín, Pablo Castells, and Iván Cantador. Neighbor selection and weighting in user-based collaborative filtering: a performance prediction approach. *ACM Transactions on the Web (TWEB)*. 2014. 8(2):1–30.
10. Yuchen Zhang, Amr Ahmed, Vanja Josifovski, and Alexander Smola. Taxonomy discovery for personalized recommendation. In *Proceedings of the 7th ACM international conference on Web search and data mining*. 2014. pages 243–

252.

11. Hamed Jelodar, Yongli Wang, Chi Yuan, Xia Feng, Xiahui Jiang, Yanchao Li, and Liang Zhao. Latent dirichlet allocation (lda) and topic modeling: models, applications, a survey. *Multimedia Tools and Applications*. 2019. 78(11): 15169–15211.
12. Bin Ren, Laurent Pueyo, Guangtun Ben Zhu, John Debes, and Gaspard Duch[^]ene. Non-negative matrix factorization: robust extraction of extended structures. *The Astrophysical Journal*. 2018. 852(2):104.
13. Jure Leskovec, Anand Rajaraman, and Jeffrey David Ullman. *Mining of massive data sets*. Cambridge university press, 2020.
14. Derek O’callaghan, Derek Greene, Joe Carthy, and P’adraig Cunningham. An analysis of the coherence of descriptors in topic modeling. *Expert Systems with Applications*. 2015. 42(13):5645–5657.
15. Donghui Wang, Yanchun Liang, Dong Xu, Xiaoyue Feng, and Renchu Guan. A content-based recommender system for computer science publications. *Knowledge-Based Systems*. 2018. 157:1–9.
16. Cataldo Musto, Pasquale Lops, Marco de Gemmis, and Giovanni Semeraro. Semantics-aware recommender systems exploiting linked open data and graph-based features. *Knowledge-Based Systems*. 2017. 136:1–14.
17. Ling Luo, Haoran Xie, Yanghui Rao, and Fu Lee Wang. Personalized recommendation by matrix co-factorization with tags and time information. *expert systems with applications*. 2019. 119:311–321.
18. Li Chen, Guanliang Chen, and Feng Wang. Recommender systems based on user reviews: the state of the art. *User Modeling and User-Adapted Interaction*. 2015 25(2):99–154.
19. Sergio Oramas, Vito Claudio Ostuni, Tommaso Di Noia, Xavier Serra, and Eugenio Di Sciascio. Sound and music recommendation with knowledge graphs. *ACM Transactions on Intelligent Systems and Technology*, 8(2), October 2016. ISSN 2157-6904. doi: 10.1145/2926718. URL: [https:// doi.org/10.1145/2926718](https://doi.org/10.1145/2926718).
20. Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hen-

- gel. Image-based recommendations on styles and substitutes. In Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval. 2015. 43–52p.
21. Mehdi Elahi, Yashar Deldjoo, Farshad Bakhshandegan Moghaddam, Leonardo Cella, Stefano Cereda, and Paolo Cremonesi. Exploring the semantic gap for movie recommendations. In Proceedings of the Eleventh ACM Conference on Recommender Systems. 2017. 326–330p.
 22. Dandan Sha, Daling Wang, Xiangmin Zhou, Shi Feng, Yifei Zhang, and Ge Yu. An approach for clothing recommendation based on multiple image attributes. In International conference on web-age information management, 2016. 272–285p. Springer.
 23. Shuiqiao Yang, Guangyan Huang, Yang Xiang, Xiangmin Zhou, and Chi-Hung Chi. Modeling user preferences on spatiotemporal topics for point-of-interest recommendation. In 2017 IEEE International Conference on Services Computing (SCC), 2017, 204–211p. IEEE.
 24. Norha M Villegas, Cristian S´anchez, Javier D´iaz-Cely, and Gabriel Tamura. Characterizing context-aware recommender systems: A systematic literature review. Knowledge-Based Systems, 2018. 140:173–200p.
 25. SM Seyednezhad, Kailey Nobuko Cozart, John Anthony Bowllan, and Anthony O Smith. A review on recommendation systems: Context-aware to social-based. arXiv preprint arXiv:1811.11866, 2018.
 26. Bouchra Bouihi and Mohamed Bahaj. An ontology-based architecture for context recommendation system in e-learning and mobile-learning applications. In 2017 International Conference on Electrical and Information Technologies (ICEIT), 2017. pages 1–6. IEEE.
 27. Yao-Ting Sung, Han-Yueh Lee, Je-Ming Yang, and Kuo-En Chang. The quality of experimental designs in mobile learning research: A systemic review and self-improvement tool. Educational Research Review, 2019. 28:100279.
 28. Michele Amoretti, Laura Belli, and Francesco Zanichelli. Utravel: Smart mobility with a novel user profiling and recommendation approach. Pervasive and

mobile computing, 2017. 38:474–489p.

29. R Logesh, V Subramaniaswamy, V Vijayakumar, and Xiong Li. Efficient user profiling based intelligent travel recommender system for individual and group of users. *Mobile Networks and Applications*, 2019. 24(3):1018–1033p.
30. Mozhgan Karimi, Dietmar Jannach, and Michael Jugovac. News recommender systems—survey and roads ahead. *Information Processing & Management*, 2018. 54(6):1203–1227p.
31. Gina George and Anisha M Lal. Review of ontology-based recommender systems in e-learning. *Computers & Education*, 2019. 142:103642.
32. Lenin Erazo-Garzón, Andrés Patinõ, Priscila Cedillo, and Alexandra Bermeo. Calms: A context-aware learning mobile system based on ontologies. In *2019 Sixth International Conference on eDemocracy & eGovernment (ICEDEG)*, 2019. pages 84–91. IEEE.
33. Fatima Ezzahraa Louhab, Ayoub Bahnasse, and Mohamed Talea. Towards an adaptive formative assessment in context-aware mobile learning. *Procedia Computer Science*, 2018. 135:441–448p.
34. Nur Baiti Afini Normadhi, Liyana Shuib, Hairul Nizam Md Nasir, Andrew Bimba, Norisma Idris, and Vimala Balakrishnan. Identification of personal traits in adaptive learning environment: Systematic literature review. *Computers & Education*, 2019. 130:168–190p.
35. Xiangmin Zhou, Dong Qin, Xiaolu Lu, Lei Chen, and Yanchun Zhang. Online social media recommendation over streams. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, 2019. pages 938–949. IEEE.
36. Xiangmin Zhou, Dong Qin, Lei Chen, and Yanchun Zhang. Real-time context-aware social media recommendation. *The Very Large Data Bases (VLDB) Journal*, 2019. 28(2):197–219.
37. Lei Mei, Pengjie Ren, Zhumin Chen, Liqiang Nie, Jun Ma, and Jian-Yun Nie. An attentive interaction network for context-aware recommendations. In *Proceedings of the 27th ACM international conference on information and knowledge management*, 2018. pages 157–166.

38. Chen He, Denis Parra, and Katrien Verbert. Interactive recommender systems: A survey of the state of the art and future research challenges and opportunities. *Expert Systems with Applications*, 2016. 56:9–27p.
39. Yueming Sun and Yi Zhang. Conversational recommender system. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, 2018. pages 235–244.
40. Konstantina Christakopoulou, Filip Radlinski, and Katja Hofmann. Towards conversational recommender systems. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016. pages 815–824.
41. Xuhui Ren, Hongzhi Yin, Tong Chen, Hao Wang, Nguyen Quoc Viet Hung, Zi Huang, and Xiangliang Zhang. Crsal: Conversational recommender systems with adversarial learning. volume 38, 2020. pages 1–40. ACM New York, NY, USA.
42. Jie Zou, Yifan Chen, and Evangelos Kanoulas. Towards question-based recommender systems. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020. pages 881–890.
43. Xu Chen, Hongteng Xu, Yongfeng Zhang, Jiayi Tang, Yixin Cao, Zheng Qin, and Hongyuan Zha. Sequential recommendation with user memory networks. In *Proceedings of the eleventh ACM international conference on web search and data mining*, 2018. pages 108–116.
44. Wang-Cheng Kang, Mengting Wan, and Julian McAuley. Recommendation through mixtures of heterogeneous item relationships. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 2018. pages 1143–1152.
45. Ruining He, Wang-Cheng Kang, and Julian J McAuley. Translation-based recommendation: A scalable method for modeling sequential behavior. In *IJCAI*, 2018. pages 5264–5268.
46. Avram Pilch. Why lenovo is the best laptop brand and apple fell off a cliff.

URL: <https://www.tomshardware.com/news/lenovo-best-laptop-brand,36927.html>.

47. Ruining He and Julian McAuley. Fusing similarity models with markov chains for sparse sequential recommendation. In 2016 IEEE 16th International Conference on Data Mining (ICDM), 2016. pages 191–200. IEEE.
48. Bal'azs Hidasi and Domonkos Tikk. General factorization framework for context-aware recommendations. *Data Mining and Knowledge Discovery*, 2016. 30(2):342–371p.
49. Shoujin Wang, Liang Hu, Longbing Cao, Xiaoshui Huang, Defu Lian, and Wei Liu. Attention-based transactional context embedding for next-item recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.
50. Jiarui Qin, Kan Ren, Yuchen Fang, Weinan Zhang, and Yong Yu. Sequential recommendation with dual side neighbor-based collaborative relation modeling. In *Proceedings of the 13th international conference on web search and data mining*, 2020. pages 465–473.
51. Dietmar Jannach and Malte Ludewig. When recurrent neural networks meet the neighborhood for session-based recommendation. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, 2017. pages 306–310.
52. Chao-Yuan Wu, Amr Ahmed, Alex Beutel, Alexander J Smola, and How Jing. Recurrent recommender networks. In *Proceedings of the tenth ACM international conference on web search and data mining*, 2017. pages 495-503.
53. Yuyun Gong and Qi Zhang. Hashtag recommendation using attention-based convolutional neural network. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI'16*, 2016. pages 2782-2788. AAAI Press. ISBN 9781577357704.
54. Yi Tay, Anh Tuan Luu, and Siu Cheung Hui. Multi-pointer co-attention networks for recommendation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018. pages 2309–2318.

55. Fajie Yuan, Alexandros Karatzoglou, Ioannis Arapakis, Joemon M Jose, and Xiangnan He. A simple convolutional generative network for next item recommendation. In Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, 2019. pages 582–590.
56. Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. volume 1, 2020. pages 57–81. Elsevier.
57. Zhibin Lu, Pan Du, and Jian-Yun Nie. Vgcn-bert: Augmenting bert with graph embedding for text classification. In Joemon M. Jose, Emine Yilmaz, Joaõ Magalhães, Pablo Castells, Nicola Ferro, Mário J. Silva, and Flávio Martins, editors, Advances in Information Retrieval, 2020. pages 369–382, Cham. Springer International Publishing.
58. Hui Fang, Guibing Guo, Danning Zhang, and Yiheng Shu. Deep learning-based sequential recommender systems: Concepts, algorithms, and evaluations. In International Conference on Web Engineering, 2019. pages 574–577. Springer.
59. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018.
60. Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer, 2019.
61. Tian Wang and Yuyangzi Fu. Item-based collaborative filtering with bert. In Proceedings of The 3rd Workshop on e-Commerce and NLP, 2020. pages 54–58, Seattle, WA, USA. Association for Computational Linguistics. doi: 10.18653/v1/2020.ecnlp-1.8. URL: <https://www.aclweb.org/anthology/2020.ecnlp-1.8>.
62. Wanvimol Nadee. Modelling user profiles for recommender systems. PhD thesis, Queensland University of Technology, 2016.
63. Diogo V Carvalho, Eduardo M Pereira, and Jaime S Cardoso. Machine

learning interpretability: A survey on methods and metrics. *Electronics*, 2019. 8 (8):832p.

64. Khalid Haruna, Maizatul Akmar Ismail, Suhendroyono Suhendroyono, Damiasih Damiasih, Adi Cilik Pierewan, Haruna Chiroma, and Tutut Herawan. Context-aware recommender system: A review of recent developmental process and future research direction. *Applied Sciences*, 2017. 7(12):1211p.

65. Zohreh Dehghani Champiri, Adeleh Asemi, and Salim Siti Salwah Binti. Meta-analysis of evaluation methods and metrics used in context-aware scholarly recommender systems. *Knowledge and Information Systems*, 2019. 61 (2):1147–1178p.

66. Rocío Can˜amares, Pablo Castells, and Alistair Moffat. Offline evaluation options for recommender systems. *Information Retrieval Journal*, 2020. pages 1–24.

67. Emaad Manzoor, Rui Li, Dhananjay Shrouthy, and Jure Leskovec. Expanding taxonomies with implicit edge semantics. In *Proceedings of The Web Conference 2020*, pages 2044–2054.

68. A Vineela, G Lavanya Devi, Naresh Nelaturi, and G Dasavatara Yadav. A comprehensive study and evaluation of recommender systems. In *Microelectronics, Electromagnetics and Telecommunications*, 2021. pages 45–53. Springer.

69. Gary Brassington. Mean absolute error and root mean square error: which is the better metric for assessing model performance? In *EGU General Assembly Conference Abstracts*, 2017. page 3574.

70. Sau˜l Vargas. Novelty and diversity enhancement and evaluation in recommender systems and information retrieval. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, 2014. pages 1281–1281.

71. Saurabh Kulkarni and Sunil F Rodd. Context aware recommendation systems: A review of the state of the art techniques. *Computer Science Review*, 2020. 37:100255.

72. Rosni Lumbantoruan, Xiangmin Zhou, Yongli Ren, and Zhifeng Bao. D-

- cars: A declarative context-aware recommender system. In 2018 IEEE International Conference on Data Mining (ICDM), 2018. pages 1152–1157. IEEE.
73. Daniel James. 8 essential live chat customer service support statistics. URL: <https://www.furstperson.com/blog/8-essential-live-chat-customer-support-statistics>.
74. Rosni Lumbantoruan, Xiangmin Zhou, Yongli Ren, and Lei Chen. I-cars: an interactive context-aware recommender system. In 2019 IEEE International Conference on Data Mining (ICDM), 2019. pages 1240–1245. IEEE.
75. Merit Valdsalu. Customer reviews. URL: <https://www.klausapp.com/blog/customer-service-rating-scales/>.
76. Yongfeng Zhang, Xu Chen, Qingyao Ai, Liu Yang, and W Bruce Croft. Towards conversational search and recommendation: System ask, user respond. In Proceedings of the 27th acm international conference on information and knowledge management, 2018. pages 177–186.
77. F Maxwell Harper and Joseph A Konstan. The movielens datasets: History and context. volume 5, 2015. pages 1–19. ACM New York, NY, USA.
78. Afshin Gholamy, Vladik Kreinovich, and Olga Kosheleva. Why 70/30 or 80/20 relation between training and testing sets: A pedagogical explanation. 2018.
79. Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In Proceedings of the 26th international conference on world wide web, 2017. pages 173–182.
80. Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. Deep learning based recommender system: A survey and new perspectives. ACM Computing Surveys (CSUR), 2019. 52(1):1–38.
81. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Advances in neural information processing systems, 2017. pages 5998–6008.
82. Lakshmipathi N. Imdb dataset of 50k movie reviews. URL: <https://www.kaggle.com/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews>.
83. Fiddler AI. ML model performance management, redefined. URL:

<https://www.fiddler.ai/>.

84. Sinong Wang, Belinda Li, Madian Khabza, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. arXiv preprint arXiv:2006.04768, 2020.

85. Коваль А.І., Яшина О.М., Радельчук Г.І. Порівняння об'єктно-орієнтованої та функційної парадигми в проектуванні програмного забезпечення. / Вісник ХНУ, серія Технічні науки, №3, 2021, С. 34-38.

ДОДАТОК А

НАУКОВА ПУБЛІКАЦІЯ

Technical sciences

ISSN 2307-5732

DOI 10.31891/2307-5732-2021-297-3-34-38

УДК 004.9

А. І. КОВАЛЬ, О. М. ЯШИНА, Г. І. РАДЕЛЬЧУК, Ю. В. ФОРКУН

Хмельницький національний університет

ПОРІВНЯННЯ ОБ'ЄКТНО-ОРІЄНТОВАНОЇ ТА ФУНКЦІЙНОЇ ПАРАДИГМ ПРОГРАМУВАННЯ У ПРОЕКТУВАННІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

У статті описано та досліджено дві парадигми проектування – об'єктно-орієнтовану та функційну. Проаналізовано доречність використання кожної з них з посиланням на їх відмінності та переваги для конкретних цілей. Незважаючи на змінні тенденції популярності цих парадигм, зробити висновок стосовно актуальності котроїсь з них залишається неможливим.

Хоча як об'єктно-орієнтоване програмування, так і функціональне програмування є суттєвими парадигмами, які мають одну і ту ж мету – розробити зрозумілі та безпомилкові програми, їх підходи різні. ООП дотримується імперативної моделі програмування, яка базується на наборі примітивів, яку надає мова програмування. Функціональна парадигма, навпаки, тісно пов'язана з декларативним стилем, що означає, що визначається лише те, що потрібно виконати, не вказуючи, як це зробити.

Здається, загальний консенсус полягає в тому, що об'єктно-орієнтована парадигма та функціональна парадигма ефективні в будь-якій певній ситуації, тому розробники завжди повинні вибрати парадигму програмування, яка робить процес продуктивним та простим.

Об'єктно-орієнтовані мови хороші, коли є фіксований набір операцій над речами, і коли код розвивається, в першу чергу додаються нові речі. Цього можна досягти, додавши нові класи, що реалізують існуючі методи, а існуючі класи залишаються самі.

Функціональні мови хороші, коли є фіксований набір речей, і коли код розвивається, в першу чергу додаються нові операції над існуючими речами. Цього можна досягти, додавши нові функції, які обчислюються з існуючими типами даних, а існуючі функції залишаються в спокої.

Об'єктно-орієнтоване програмування, так само як і функційне, має своє місце у сучасній розробці програмного забезпечення. В той час, як розвиток технологій Big Data дає новий поштовх для використання функційного програмування, об'єктно-орієнтоване програмування, у свою чергу, залишається актуальним задля роботи для відображення в коді об'єктів реального світу.

Ключові слова: парадигма програмування, об'єкт, функція, об'єктно-орієнтоване програмування, функційне програмування.

A. I. KOVAL, O. M. YASHYNA,
G. I. RADELCHUK, Y. V. FORKUN
Khmelnitskyi National University

COMPARISON OF OBJECT-ORIENTED AND FUNCTIONAL PROGRAMMING PARADIGMS IN SOFTWARE DESIGN

This article describes two types of paradigms – object-oriented and functional paradigms. Paradigm stands for a style and an approach to perform any kind of coding activities. Relevance of usage for each of them as well as their differences and benefits were analyzed. Regardless of ever-changing tendencies in popularity of both paradigms it is impossible to acknowledge any of them to deprecate another one. Although object-oriented programming as well as functional programming are essential in their approaches they have the same goal – to make comprehensive programs without possible mistakes. OOP follows imperative programming model which is based on a set of primitives the given language provides. Functional paradigm is intertwined with declarative style which imply what is to be done, but not how to do it. We may come to conclusion that the consensus is that object oriented paradigm and functional paradigm can be effective in a peculiar situation. Therefore, developers are to choose and pick the programming paradigm for a given task to make a process as simple and productive as it can be.

Object-oriented programming languages are a good choice when you have a fixed set of operations on things and you add new things for your code to evolve. You can achieve it by adding new classes that implement existing methods while existing classes remain the same. Functional languages may be a better choice if you have a fixed set of things and you add new operations on existing things for your code to evolve. You can achieve it by adding new functions which are to be computed with existing data types while existing functions remain the same. Object-oriented programming as well as functional programming has its place in modern approaches to software development. Meanwhile the functional programming is being reconsidered and used much more often due to Big Data Technologies while object-oriented programming remains popular to perform representation of real-life objects in the code.

Keywords: programming paradigm, object, function, object-oriented programming, functional programming.

Вступ

Парадигма програмування – це стиль або «спосіб» програмування. Іншими словами – це система ідей і понять, які визначають стиль написання комп'ютерних програм, а також спосіб мислення програміста [1]. Парадигми програмування відрізняються одна від одної залежно від особливостей та стилю, який вони підтримують. Існує декілька особливостей, що визначають парадигму програмування, а саме: модульність, об'єкти, переривання або події, керування потоком тощо. Кожна парадигма програмування має власні переваги, про які варто знати перед тим, як її використовувати.

Дві найпопулярніші парадигми програмування при розробці програмного забезпечення (ПЗ) – це об'єктно-орієнтоване програмування (ООП) та функційно-орієнтоване програмування (ФООП). Саме ці дві парадигми найчастіше використовують розробники, дизайнери та проектувальники при розробці програмного забезпечення.

Об'єктно-орієнтовані мови хороші, коли у розробника є фіксований набір операцій; і коли програмний код розширюється, він у першу чергу додає нові речі. Цього можна досягти, додавши нові класи, які реалізують існуючі методи.

Функційні мови хороші, коли у розробника є фіксований набір речей; і коли код розвивається, він у першу чергу додає нові операції над існуючими речами. Цього можна досягти, додавши нові функції, які обчислюються з існуючими типами даних. Також є можливим використання обох парадигм (відповідно до власних потреб) за допомогою мультипарадигмових мов програмування, які підтримують як об'єктно-орієнтовану концепцію, так і функційну.

Метою статті є дослідження і порівняльний аналіз особливостей об'єктно-орієнтованої та функційної парадигм програмування на предмет їх ефективності та зручності використання при проектуванні ПЗ.

Виклад основного матеріалу

Об'єктно-орієнтоване програмування. Об'єктно-орієнтоване програмування – одна з парадигм програмування, яка розглядає програму як множину «об'єктів», що взаємодіють між собою. Основу ООП складають чотири основні концепції: інкапсуляція, успадкування, поліморфізм та абстракція. Відповідно до парадигми ООП кожен об'єкт здатний отримувати повідомлення, обробляти дані та надсилати повідомлення іншим об'єктам. Кожен об'єкт – це своєрідний незалежний автомат з окремим призначенням та відповідальністю.

Парадигма ООП використовує об'єкти для репрезентації речей у програмі (до прикладу, структури даних). Інколи ці об'єкти можуть бути і реальними речами. Кожний об'єкт має атрибути, що містять дані, якими можна маніпулювати за допомогою властивих йому методів чи функцій. Наприклад, існує об'єкт під назвою «Особа», який має різні атрибути людини, такі як вага, зріст, колір шкіри, колір волосся тощо. Окрім атрибутів, об'єкт також має функції. Наприклад, об'єкт «Людина» може мати такі функції, як їжа, сон, прогулянка тощо. Ці функції використовують дані, які об'єкт зберігає як атрибути.

Об'єктно-орієнтоване проектування – це стиль програмування, який допомагає програмістам моделювати реальні сценарії, а отже, при ООП існує прямиий перехід від реальних об'єктів до фактичного коду. Деякі приклади мов програмування, які використовуються в ООП, – C++, Java, Python, C# та ін. [2].

Функційне програмування. Функційне програмування – парадигма програмування, яка розглядає програму як обчислення математичних функцій та уникає станів і змінних даних. Іншими словами, функційне програмування є способом створення програм, в яких єдинною дією є виклик функції, єдиним способом розбиття програми є створення нового імені функції та задання для цього імені виразу, що обчислює значення функції, а єдиним правилом композиції є оператор суперпозиції функцій.

Важливим поняттям у функційному програмуванні є референційна прозорість, яка означає, що для заданої функції та вхідного значення вона поверне той самий результат, незалежно від порядку програми (тобто незалежно від того, коли і з якої точки програми вона викликається) [3]. У функційному програмуванні вихід функції повністю покладається на аргументи функції. Так, наприклад, якщо викликається функція `sum()`, яка обчислює суму двох змінних як вхідних даних і повертає цю суму, то вихідний результат завжди буде однаковим. Отже, у функційно-орієнтованому програмуванні функції програми є максимально передбачуваними.

У цій парадигмі програмування є невеликі функції, які виконують лише свою частину, а отже, код у програмуванні, орієнтованому на функції, є модульним і чистим. Також до програм, написаних у межах ФООП, доволі легко застосувати модульне тестування.

До функційно-орієнтованих мов програмування відносять мови Lisp, Haskell, Clojure, F# та деякі інші.

Недоліки об'єктно-орієнтованої та функційної парадигм програмування. Перша проблема для об'єктно-орієнтованого програмування полягає в тому, що деякі функції залежать від свого класу, і, отже, важко використовувати ці функції з іншим класом [4]. Також відомо, що об'єктно орієнтоване програмування є менш ефективним, але складнішим для роботи програмістів.

З іншого боку, є низка недоліків, пов'язаних із функційним програмуванням. Перш за все, функційно-орієнтоване програмування не є загальним методом. Програмістам легше мислити з точки зору об'єктно-орієнтованого програмування порівняно з ФООП. Об'єктно-орієнтоване програмування полягає в моделюванні об'єктів реального життя в кодї, з іншого боку, маніпулювання даними є найважливішим у функційному програмуванні. Рівень складності функційно-орієнтованого програмування гарантує, що небагато програмістів використовують цей стиль програмування. Це також означає, що стає менше розробників, які вносять свій внесок у спільноту, а отже, менше інформації та контенту про функційно-орієнтоване програмування.

Відмінності об'єктно-орієнтованого та функційного програмування. В обох парадигмах програмування кінцевою метою є створення програм та уможливлення легкої і швидкої розробки з мінімальною кількістю помилок. Проте для спільних цілей ці дві парадигми застосовують різні методи зберігання та обробки даних. В об'єктно-орієнтованому програмуванні дані зберігаються в атрибутах об'єктів, і маніпулювання ними здійснюється за допомогою функцій об'єкта. Функційне ж програмування полягає у перетворенні даних шляхом створення нових версій цих даних та маніпулювання ними.

Фундаментальною є різниця концепцій між двома парадигмами. У функційно-орієнтованому дизайні функція програми є найбільш суттєвою. З іншого боку, в об'єктно-орієнтованій програмі основна

увага приділяється даним та їх маніпулюванню, а не функціям. Об'єкти в об'єктно-орієнтованому програмуванні – це незалежні сутності, які зберігають усі стани та можуть швидко змінюватися.

Ще одна відмінність між парадигмами програмування полягає в тому, що функційно-орієнтоване програмування використовує ітеративну процедурну декомпозицію, що є стратегією зверху вниз [5]. Функційно-орієнтована програма трактується як ієрархія зростаючих рівнів деталей, де існує першочерговий опис функції, і з кожним наступним етапом він уточнюється. Це означає, що програма розробляється, починаючи від висококонцептуальної моделі, а далі – переходячи до деталей нижчого рівня. Цей процес повторюється до тих пір, поки не буде досягнуто рівень атомарності окремої функції.

З іншого боку, ООП більше пов'язане з реальними речами, які переносяться на об'єкти в об'єктно-орієнтованому програмуванні. Кожен об'єкт має характеристики з точки зору його атрибутів та поведінки. Ці об'єкти можуть бути розподілені, і, отже, вони виконуються послідовно або паралельно [6].

Ще однією темою для обговорення є абстракція. У ФООП абстракція є функціями реального світу, тоді як в ООП абстракція даних – це сутності реального світу. Крім того, у функційно-орієнтованому програмуванні функції групуються разом, за допомогою чого отримується функція вищого рівня, тоді як в об'єктно-орієнтованому, функції групуються разом на основі своїх даних, а класи асоціюються з їх методами. У ФООП інформація про стан представлена в централізованій спільній пам'яті, а в об'єктно-орієнтованому програмуванні інформація про стан реалізується або розподіляється між об'єктами.

Поряд з усіма іншими відмінностями ще однією різницею між обома стилями програмування є їх використання. Функційно-орієнтоване програмування є популярним в обчислювальних додатках, тоді як об'єктно-орієнтоване програмування використовується в системі, що розвивається, імітує бізнес чи бізнес-кейси.

На основі результатів аналізу основні характеристики об'єктно-орієнтованої та функційної парадигми програмування зведені у порівняльну таблицю 1.

Таблиця 1

Порівняння основних характеристик об'єктно-орієнтованої та функційної парадигми програмування

Характеристика	Парадигма	
	Об'єктно-орієнтована	Функційна
Фокус	На понятті об'єктів	На оцінці функції
Дані	Використовує змінні дані	Використовує незмінні дані
Модель	Імперативна модель	Декларативна модель
Паралельне програмування	Не підтримує	Підтримує
Виконання	Оператори виконуються в певному порядку	Твердження може виконуватися в будь-якому порядку
Ітерація	В основному використовує цикли	Використовує рекурсію
Основний елемент	Об'єкти та методи	Функції та змінні

Тенденції використання об'єктно-орієнтованого та функційного програмування. Хоча функційному програмуванню і надають перевагу в аспекті паралельності та імутабельності коду, твердження стосовно витіснення ним об'єктно-орієнтованої парадигми є невірним. Багаторазове використання коду, абстрагування даних, ефективне вирішення проблем та гнучкість поліморфізму досягаються лише за допомогою об'єктно-орієнтованого програмування. Що стосується системної безпеки, то функційне програмування перевершує об'єктно-орієнтоване програмування, особливо беручи до уваги, що великі компанії залежать від машинного навчання та штучного інтелекту у своїх бізнес-додатках. Неможливо переоцінити значення функційного програмування при написанні коду для розробки та тренування моделей машинного навчання.

Динаміка популярності об'єктно-орієнтованого та функційного програмування наочно продемонстровані на рис. 1 та 2 [7].

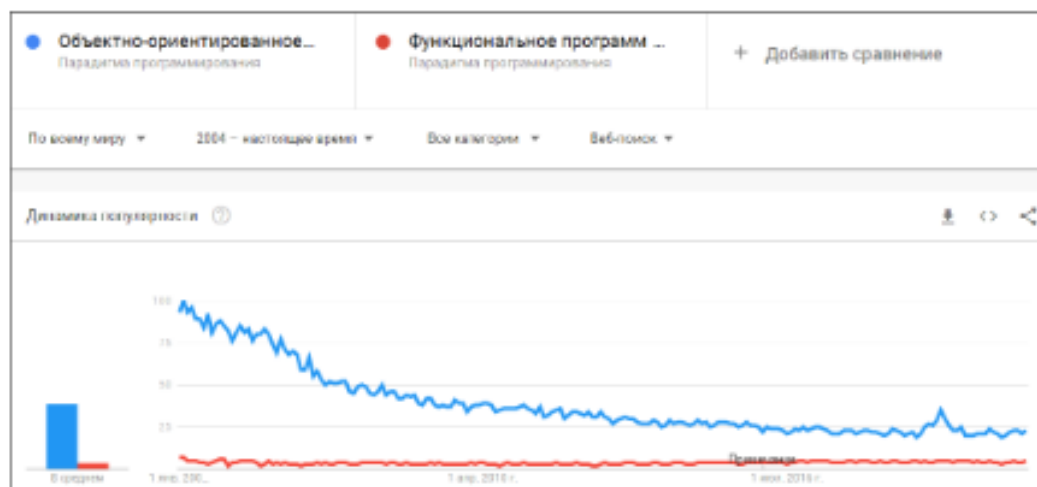


Рис. 1. Діаграма частоти Google-запитів за 2004 року по теперішній час [7]

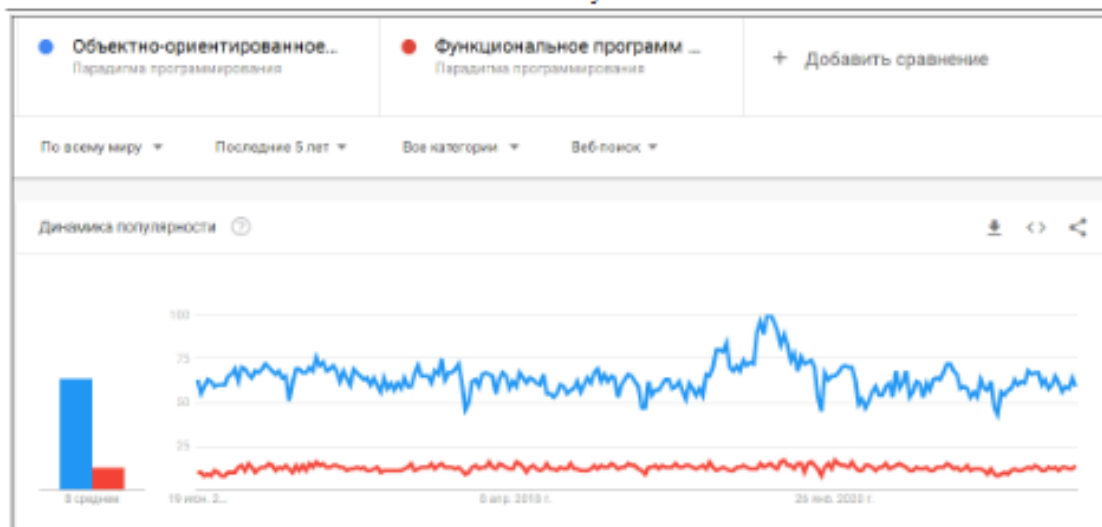


Рис. 2. Діаграма частоти Google-запитів за останні п'ять років [7]

Висновки

На основі опрацювання літератури та проведеного дослідження можна зробити наступні висновки.

Описані парадигми вимагають різних підходів до їх використання. Для вирішення проблем функційно-орієнтоване програмування використовує підхід зверху вниз, тоді як об'єктно-орієнтоване використовує підхід знизу догори. Ще один момент, на який слід звернути увагу, це те, що розробка за функційною парадигмою розпочинається з визначення сценаріїв та діаграм використання, а об'єктно-орієнтоване програмування розпочинається з визначення об'єктів і класів. Декомпозиція також відрізняється в обох парадигмах: у функційно-орієнтованому програмуванні декомпозиція здійснюється на рівні функції/процедури, тоді як в об'єктно-орієнтованому програмуванні – на рівні класів.

Незважаючи на те, що функційне та об'єктно-орієнтоване програмування є досить протилежними поняттями, вони переслідують одну й ту ж мету – створення простих для розуміння, без помилок та ефективних програм, – і обидва роблять свою роботу добре.

Хоча і ООП, і ФОП є двома важливими парадигмами програмування, які мають спільну мету створення зрозумілих, гнучких та працездатних програм, вони використовують два різні підходи до створення цих програм. ООП дотримується імперативної моделі програмування, що спирається на набір примітивів, які надає відповідна мова програмування. За допомогою ООП визначається, ЯК її потрібно впроваджувати, не вказуючи, ЩО потрібно виконати. З іншого боку, функційне програмування тісно пов'язане з декларативним стилем програмування, який лише визначає, ЩО потрібно виконати, не вказуючи, ЯК.

Отже, правильне питання полягає не в тому, який метод кращий, а в тому, який з них найкраще вирішує поставлену проблему. І навіть якщо використовується оптимальна парадигма, розроблена програма ризикує «розвалитися», якщо вона стане занадто складною. Проблема полягає в тому, що досить важко уникнути складності у великих проектах, у яких бере участь багато людей. У таких випадках може знадобитися гібридний підхід, який об'єднує переваги обох парадигм, і, відповідно, мультипарадигмова мова програмування.

Література

1. Programming paradigm. Wikipedia. the free encyclopedia. URL: https://en.wikipedia.org/wiki/Programming_paradigm
2. Гамма Э. Приемы объектно-ориентированного проектирования. Паттерны проектирования / Э. Гамма, Р. Хельм, Р. Джонсон, Дж. Влиссидес ; пер. с англ. – СПб : Питер, 2017. – 368 с.
3. Берд Р. Жемчужины проектирования алгоритмов. Функциональный подход. С примерами на языке Haskell / Р. Берд. – М. : Изд-во «ДМК-Пресс»? 2015. – 330 с.
4. Вайсфельд М. Объектно-ориентированное мышление / М. Вайсфельд ; пер. с англ. – СПб : Питер, 2018. – 304 с.
5. Окасаки К. Чисто функциональные структуры данных / К. Окасаки ; пер. с англ. – М. : Изд-во «ДМК-Пресс», 2016. – 252 с.
6. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений / Г. Буч, Роберт, А. Максимчук и др. ; пер. с англ. – 3-е изд. – М. : ИД «Вильямс», 2020. – 720 с.
7. Динамика популярности объектно-ориентированного и функционального программирования [Электронный ресурс] // Google Trends. – Режим доступа : <https://trends.google.ru/trends/explore?date=today%205-y&q=%2Fm%2F05prj,%2Fm%2F02ykw>

References

1. Programming paradigm. Wikipedia. the free encyclopedia. URL: https://en.wikipedia.org/wiki/Programming_paradigm
2. Gamma E. Priemy obektno-orientirovannogo proektirovaniya. Patterny proektirovaniya / E. Gamma, R. Helm, R. Johnson, Dzh. Vlissides ; per. s angl. – SPb : Piter, 2017. – 368 s.
3. Berd R. Zhasnchuzhiny proektirovaniya algoritmov. Funktsionalnyj podhod. S primerami na yazyke Haskell / R. Berd. – M. : Izd-vo «DMK-Press», 2015. – 350 s.
4. Vajsfeld M. Obektno-orientirovannoe myshlenie / M. Vajsfeld ; per. s angl. – SPb : Pitar, 2018. – 304 s.
5. Okasaki K. Chisto funktsionalnye struktury dannyh / K. Okasaki ; per. s angl. – M. : Izd-vo «DMK-Press», 2016. – 252 s.
6. Buch G. Obektno-orientirovannyj analiz i proektirovanie s primerami prilozhenij / G. Buch, Robert, A. Maksimchuk i dr. ; per. s angl. – 3-e izd. – M. : ID «Vilyams», 2020. – 720 s.
7. Dinamika populyarnosti obektno-orientirovannogo i funktsionalnogo programirovaniya [Elektronnyj resurs] // Google Trends. – Rezhim dostupa : <https://trends.google.ru/trends/explore?date=today%205-y&q=%2Fm%2F05pej,%2Fm%2F02ykw>

А. І. КОВАЛЬ
 О. М. ЯШИНА
 Г. І. РАДЕЛЬЧУК
 Ю. В. ФОРКУН

ORCID ID: 0000-0001-7816-1662
 ORCID ID: 0000-0002-7906-4191

andrey1998t29@gmail.com
 ksusha.ja@gmail.com
 gal_2015@ukr.net
 forkun@ridne.net

Рецензия/Peer review : 04.05.2021 р. Надрукована/Printed :30.06.2021 р.

ДОДАТОК Б

ПРЕЗЕНТАЦІЙНІ МАТЕРІАЛИ

Метод створення рекомендаційних систем

Коваль А.І.

Науковий керівник
к.т.н., доцент Яшина О.М.

Хмельницький
2021

Актуальність роботи.

Рекомендаційна система — підклас системи фільтрації інформації, яка буде рейтинговий перелік об'єктів (фільми, музика, книги, новини, вебсайти), яким користувач може надати перевагу. Для цього використовується інформація з профілю користувача. Існують дві основні стратегії створення рекомендаційних систем: фільтрація вмісту і колаборативна фільтрація. При фільтрації вмісту створюються профілі користувачів і об'єктів. Профілі користувачів можуть містити демографічну інформацію або відповіді на певний набір питань.

Профілі об'єктів можуть містити назви жанрів, імена акторів, імена виконавців, тощо. Або якусь іншу інформацію в залежності від типу об'єкта.

Цей підхід застосований у проекті Music Genome Project: музичний аналітик оцінює кожну композицію за сотнями різних музичних характеристик, які можна використати для виявлення музичних уподобань користувача.

При колаборативній фільтрації використовується інформація про поведінку користувачів у минулому — наприклад, інформація про придбання або оцінки. В цьому разі не має значення, з якими типами об'єктів ведеться робота, але при цьому можна брати до уваги неявні характеристики, які складно було б врахувати при створенні профілю. Основна проблема цього типу рекомендаційних систем — «холодний старт»: відсутність даних про користувачів чи об'єкти, які нещодавно з'явилися у системі. Потребують удосконалення методи створення рекомендаційних систем для покращення надання пропозицій користувач інтернет-сервісів.

Мета роботи – розробити метод створення рекомендаційних систем для досягнення ефективності взаємодії користувача щодо пропонованих товарів та послуг в інтернеті з використанням сучасних інформаційних технологій та методів розробки програмних продуктів.

Для досягнення мети дослідження поставлено наступні завдання:

Здійснити аналіз відомих рішень з розробки рекомендаційних систем;
здійснити моделювання системи та результати експериментів з моделлю;
розробити метод та алгоритми створення рекомендаційних систем;
здійснити реалізацію запропонованого рішення.

Об'єктом дослідження є процеси пошуку товарів та послуг у мережі Інтернет з використанням рекомендаційних систем.

Предметом дослідження є методи створення рекомендаційних систем.

У кваліфікаційній роботі проаналізовані методи створення рекомендаційних систем з використанням різних стратегій щодо надання релевантного пошуку на запит.

Під час виконання завдань дослідження були застосовані:

методи обробки даних;
методи класифікації.

3

Наукова новизна одержаних результатів полягає в наступному:

розроблений метод створення рекомендаційних систем, що дає змогу досягти ефективності у взаємодії користувача щодо пропонованих товарів та послуг в інтернеті з використанням сучасних інформаційних технологій та методів розробки програмних продуктів, на відміну від відомих методів, враховує поєднання стратегій релевантного пошуку на запит та фільтрації даних в динамічному режимі.

За результатами кваліфікаційної роботи опубліковано наукову статтю у фаховому науковому виданні «Вісник Хмельницького національного університету»:

Коваль А.І., Яшина О.М., Радельчук Г.І. Порівняння об'єктно-орієнтованої та функційної парадигми в проектуванні програмного забезпечення. / Вісник ХНУ, серія Технічні науки, №3, 2021, с. 34-38.

4

Асортимент товарів у фізичних магазинах обмежений можливостями торговельних площ. Кількість товарів, що пропонуються, залежить від розміру магазину. У більшості випадків виникає потреба відвідувати декілька фізичних магазинів, щоб придбати те, що потребується. Цифрові технології дозволили споживачам отримати доступ до широкого спектру продуктів через Інтернет. Зі збільшенням розвитку мобільних пристроїв та онлайн-сервісів, онлайн-діяльність стала важливою частиною повсякденного життя людей. Користувачі отримують доступ до інтернет-сайтів для покупок, бронювання квитків та розважальних заходів. Незліченна кількість предметів доступна на інтернет-сайтах, залучаючи величезну кількість соціальних користувачів. За інформацією з повідомлень Amazon, близько 5 мільярдів людей в 2021 році наявні активними користувачами магазину з Інтернету, а Amazon Marketplace пропонує понад 353 мільйонів продуктів в Інтернеті, включаючи одяг, взуття, ювелірні вироби та предмети домашнього вжитку [1]. І ріст користувачів та асортимент товарів туди будуть продовжувати швидко зростати [2]. За допомогою онлайн-покупок споживачі можуть порівнювати ціни і купувати товари у сотень різних продавців, усуваючи необхідність відвідати кілька фізичних магазинів. Але при цьому виникає нова проблема щодо спроможності інтернет-покупців знайти продукти, які відповідають їх потребам.

5

Для досягнення поставленої мети необхідно:

- розробити модель рекомендаційної системи;
- розробити модель поведінки користувача з врахуванням особливостей проаналізованих недоліків в цій предметній області;
- здійснити моделювання системи та результати експериментів з моделлю;
- розробити метод та алгоритми створення рекомендаційних систем;
- здійснити реалізацію запропонованого рішення.

6

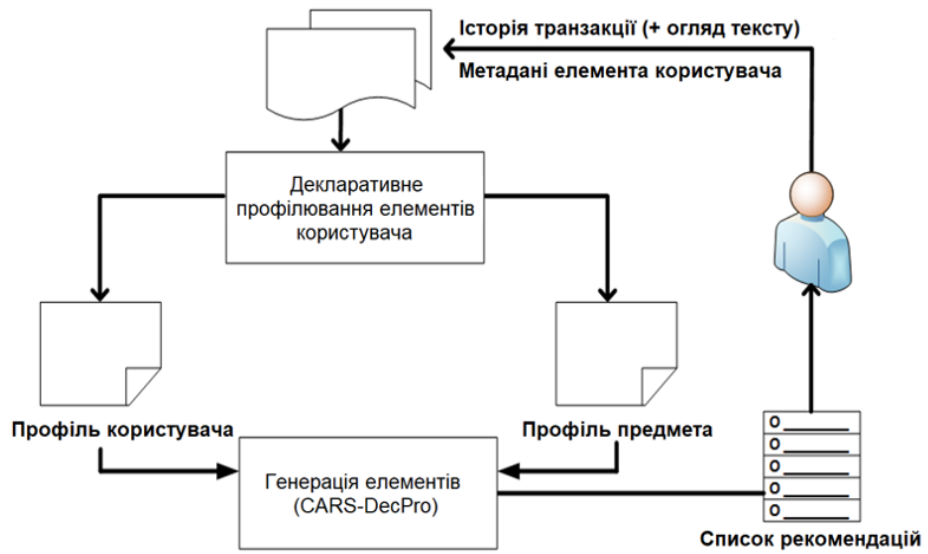


Рисунок 2.1 - Структура декларативної рекомендаційної системи з профілем користувача-елемента

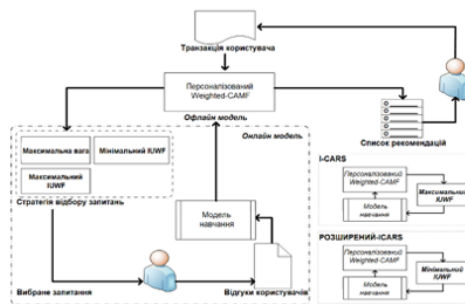


Рисунок 3.1 - Архітектура запропонованої інтерактивної контекстно-обізнаної рекомендаційної системи

Кроки методу створення двонаправлених рекомендаційних систем:

1. Створити двонаправлену контекстно-свідому послідовну рекомендаційну систему, яка використовує глобальну та місцеву контекстну інформацію для прогнозування наступного пункту двонаправленими способами. Хоча локальна контекстна інформація може передбачати наступний елемент, який відповідає поведінці користувача, глобальні контексти можуть визначити пов'язані елементи, які відповідають притаманним цьому користувачеві інтересам.

2. Створити новий контекстний графік та новий графік контексту транзакцій. Він фіксує актуальність між контекстами та цільовими уподобаннями користувачів, тоді як попередній зменшує ймовірність транзакції в різних контекстах. Використовуючи ці графіки, динаміка поведінки користувача і контекстів може бути захоплена з метаданих елемента, відгуків користувачів і самоописаних транзакцій користувачів, що є надійним для динамічних змін контексту.

3. Використовуємо стратегію оновлення моделі на основі структури кластера користувача, яка адаптивно оновлює модель глибокої нейронної мережі. Модель оновлюється лише тоді, коли змінюється структура кластера. Це дає змогу відображати поточні послідовні налаштування користувача та гарантує оновлення моделі в режимі реального часу.

4. Проводимо експерименти над трьома реальними наборами даних для перевірки ефективності запропонованого рішення з точки зору різних показників ранжирування та проводимо дослідження для аналізу внесків ключових компонентів у запропонованій моделі. Результати експериментів використовуємо для здійснення самонавчання нейронної мережі.



Рисунок 4.1 - Архітектура системи

ВИСНОВКИ

В результаті проведених досліджень та завдань з усунення недоліків відомих підходів було [95] розроблено метод розподілу обчислювальних ресурсів для розробленої системи згідно розробленого методу та проведено з нею експерименти.

Системи рекомендацій спрямовані на те, щоб допомогти користувачам у виявленні та отриманні певних елементів з великої колекції елементів, які мають відношення до їх уподобань. Переваги користувачів, як правило, вивчаються з їх попередніх транзакцій, тобто для їх ефективного застосування повинен накопичуватися попередній результат. Уподобання користувача важко зрозуміти, оскільки на них впливають різні незрозумілі ситуації або контексти, які існують, коли користувач приймає рішення. Контексти динамічні, і значення кожного контексту кожного користувача змінюється, що призводить до динамізму уподобань користувача. Було доведено, що система рекомендацій з контекстною обізнаністю (СРКО) підвищує якість рекомендацій шляхом включення контекстної інформації та персоналізації їх для кожного користувача. Персоналізація контекстів для кожного користувача та включення їх у процес рекомендацій є двома основними задачами СРКО. Було розглянуто ці два питання з метою підвищення продуктивності рекомендаційної системи з точки зору ефективності та продуктивності. Зокрема, залежно від характеристик наявних даних, застосовано три різні підходи для вирішення цих завдань. Перша характеристика даних пов'язана з ситуацією, коли налаштування користувачів доступні через відгуки користувачів про придбані товари. Як правило, ці відгуки описують рейтинг користувача товару. У цьому дослідженні запропоновано використовувати вики користувача згідно від'ємної матриці (ВКВМ), нову модель для визначення контекстів користувачів з їх заявлених переглядів. ВКВМ персоналізує та визначає найбільш релевантні контексти для кожного користувача. Потім включаємо персоналізовані контекстом у процес генерації рекомендацій. Введемо першу модель - декларативно-контекстну систему рекомендацій (ДКСР). ДКСР може підвищити якість рекомендацій для найбільш активних користувачів за допомогою комплексних досліджень. Також, введемо другу модель для декларативного користувача з елементом профілювання згідно контекстною обізнаністю рекомендацій системи (ДКПКОРС). Модель явних факторів ДКПКОРС, яка включає не тільки профілі користувачів, але і профілі товарів в модель рекомендацій має на меті працювати над усіма користувачами, незалежно від того, чи активні вони чи не активні. ДКПКОРС відповідає профілям користувачів і елементів, щоб рекомендувати користувачеві найбільш релевантні елементи.

У більшості випадків налаштування користувачів можуть бути взяті з історії активності. Через динамічний характер контекстів у прийнятті рішень, може з'явитися нова характеристика, яка ще не була захоплена, що є другою особливістю даних. З огляду на цю характеристику, розроблено новий метод захоплення користувачьких проміжних переваг, використовуючи взаємодію користувачів із системою. Кожен контекст має різноманітний пріоритет для кожного користувача. Спочатку введемо новий підхід за помилкою середнього значення (СЗ) за персоналізованою зважено-контекстною матрицею факторизації (ПЗКМФ), щоб дізнатися важливість контекстів для кожного користувача. Ми збираємо миттєві налаштування користувачів у верхній частині персоналізованої зважено-контекстною матриці факторизації, використовуючи їх для надання входу в систему, використовуючи один з трьох розглядуваних варіантів. Відповідь користувача повертається в модель за допомогою контекстного навчання, а потім найбільш релевантні елементи повертаються користувачеві. Третя характеристика даних виникає, коли порядок транзакцій впливає на переваги користувача. Рішення приймається в цьому випадку на основі попередніх або майбутніх дій користувача. Тому, пропонується контекстний графік контекстно згрупованої нормованої взаємної точкової інформації (КЗНВТІ) та контексту транзакції (КТ) для послідовного захоплення контекстів користувачів. Зокрема, КЗНВТІ моделює взаємозв'язок між контекстами, що дозволяє нам вибодувати всі відповідні контексти в даному контексті. Контекст транзакції моделює важливість контекстів у кожній транзакції користувача, фіксуючи ситуації, в яких користувач транзакцій. Включаємо вивчені переваги користувача в систему рекомендацій за допомогою двох направлених представлень кодерів трансформерів (ДПКТ). Це надає змогу розглядати декларативні контексти, в яких користувачі проводять транзакції, особливо з точки зору відгуків користувачів та / або попередньо визначених неявних або явних текстів. Тому, пропонується три нові рекомендаційні підходи, які базуються на характеристиках наявних даних. Результати експериментів показують, що запропоновані підходи можуть вирішити цілі запропонованих рішень і покращити існуючі аналоги конкурентів.

Доповідь закінчено
Дякую за увагу!

Завідувачу кафедри інженерії програмного
забезпечення проф. Бедратюку Л. П.
здобувача вищої освіти
Коваля А.І
факультет ІТ, 2 курс, група ПЗМ-20-1

ЗАЯВА

З правилами чинного Положення «Про дотримання академічної доброчесності в Хмельницькому національному університеті» від 26.09.2020 (зі змінами від 26.11.2020), згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений. Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

23.11.2021 р.
дата


підпис

Anti-Plagiarism v-15.257**Максимальне співпадіння з одним документом 2.0%**

Словники перевірки: en_US, ru_RU, ua_UA. Помилки в документах: 12%

ID: 99081 Назва: Метод створення рекомендаційних систем Додано в БД: 2021-12-13 Автора: А.І. Коваль Керівники: О.М. Яшина Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	157483	1219	4014 (3%)	45 (4%)

Джерело плагіату		Наявність плагіату в документі	
ID	Опис	Символи	Лексеми



Ім'я користувача:
Кафедра ІПЗ

ID перевірки:
1009667635

Дата перевірки:
13.12.2021 19:26:04 EET

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
13.12.2021 19:26:53 EET

ID користувача:
100005589

Назва документа: diplom_Koval_A_без_дод

Кількість сторінок: 86 Кількість слів: 20583 Кількість символів: 162533 Розмір файлу: 237.77 KB ID файлу: 1009667026

8.28%**Схожість**

Найбільша схожість: 2.43% з джерелом з Бібліотеки (ID файлу: 1009533639)

6.13% Джерела з Інтернету 371 Сторінка 88

2.76% Джерела з Бібліотеки 98 Сторінка 92

0% Цитат

Вилучення цитат вимкнено

Вилучення списку бібліографічних посилань вимкнено

0%**Вилучень**

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи 4

**РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ
КАФЕДРИ ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ**

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: «Метод створення рекомендаційних систем»

Автор: Коваль Андрій Ігорович

Спеціальність: 121 – Інженерія програмного забезпечення

Освітня програма: Освітньо-професійна програма «Інженерія програмного забезпечення»

Науковий керівник: Яшина Оксана Миколаївна, кандидат технічних наук, доцент

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
5	Інше:	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

1) у тексті дипломної роботи системами перевірки на плагіат виявлено схожість з деякими документами в частині загальноживаних обов'язкових словосполучень у стандартних бланках (титулка, бланк завдання), у структурі змісту, назвах розділів/підрозділів тощо та в назвах публікацій у переліку джерел посилання;

2) в якості запозичень системою було зафіксовано деякі послідовності вихідного коду і посилання на бібліотеки, які є стандартними мовними конструкціями програмування та не можуть розглядатися як об'єкт авторських прав і, відповідно, їх порушення;

3) усі запозичення є фрагментарними або мають належним чином оформленні посилання;

4) виявлені модифікації тексту не впливають на відсоток схожості.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів ідентичності/схожості, складає 8,28% і адресується до 469 джерел, що, з урахуванням наведених обґрунтувань, відповідає характеру теми і свідчить на користь дипломної роботи.

Керівник

О. М. Яшина

Гарант ОП

О. М. Яшина

Завідувач кафедри

Л. П. Бедратюк

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА ДИПЛОМНУ РОБОТУ

Дипломник: Коваль А.І.

Тема: Метод створення рекомендаційних систем

Спеціальність: 121 «Інженерія програмного забезпечення»

Обсяг дипломної роботи:

Кількість листів креслень —; кількість сторінок записки 88

1. Короткий зміст роботи та прийнятих рішень У роботі запропоновано метод створення рекомендаційних систем, який дає змогу досягти ефективності у взаємодії користувача щодо пропонованих товарів та послуг в мережі інтернет з використанням сучасних інформаційних технологій та методів розробки програмних продуктів. На відміну від відомих методів, враховує поєднання стратегій релевантного пошуку на запит та фільтрації даних в динамічному режимі.

2. Висновок про відповідність роботи дипломному завданню Дипломна робота відповідає виданому завданню

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У першому розділі наведено основні поняття та стратегії досягнення результату в рекомендаційних системах. У другому розділі представлено метод створення рекомендаційних систем з контекстною обізнаністю згідно журналу відгуків користувачів. У третьому розділі представлено вимоги до створення інтерактивної рекомендаційної системи та метод створення інтерактивних рекомендаційних систем. У четвертому розділі наведено реалізації прототипу рекомендаційної системи.

4. Позитивні сторони роботи: Запропоновано метод створення рекомендаційних систем, що реалізує взаємодію користувача щодо пропонованих товарів та послуг в мережі інтернет із залученням елементів штучного інтелекту.

5. Негативні сторони роботи: 3 роботи не зрозуміло на скільки підвищилась ефективність запропонованої системи (зокрема часова). В експериментальній частині відсутні приклади векторів наборів тестових даних, графіки навчання та похибки системи. Не обґрунтовано вибір K-means кластеризації.

6. Оцінка графічного оформлення та пояснювальної записки роботи: —

7. Відгук про роботу в цілому: В загальному робота виконана на достатньому рівні.

8. Інші зауваження: —

9. Оцінка дипломної роботи:

Розглянувши позитивні та негативні сторони представленої дипломної роботи вважаю, що робота заслуговує оцінки «добре» 3,75 (С)

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) Нічепорук Андрій Олександрович, к.т.н., доцент, доцент кафедри комп'ютерної інженерії та інформаційних систем

“ 14 ” грудня 2021р.