

Хмельницький національний університет
Факультет програмування
та комп'ютерних і телекомунікаційних систем
Кафедра інженерії програмного забезпечення

ДИПЛОМНА РОБОТА

Технологія розробки програмної системи рекомендацій для знайомств людей та
Назва теми

груп за інтересами у віртуальних спільнотах

Рівень вищої освіти Другий (магістерський)


Галузь знань 12 «Інформаційні технології»

Спеціальність 121 «Інженерія програмного забезпечення»

Освітня програма Освітньо-професійна програма «Інженерія програмного
забезпечення»


Шифр ДРПЗ. 150171.01.06.ПЗ

Виконав студент 2 курсу група ПЗм-19-1


Підпис


А. О. Протоковський
Ініціали, прізвище

Керівник канд. техн. наук, доцент
Науковий ступінь, звання


Підпис

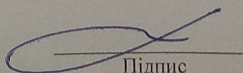
Ю. В. Форкун
Ініціали, прізвище

Нормоконтролер


Підпис

Г. І. Радельчук
Ініціали, прізвище

До захисту допускаю:
Завідувач кафедри інженерії
програмного забезпечення


Підпис

Л. П. Бедратюк
Ініціали, прізвище

7 грудня 2020 р.

Хмельницький 2020

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет Програмування та комп'ютерних і телекомунікаційних систем
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

02 09 2020 р.

ЗАВДАННЯ НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ)

Протоковському Андрію Олексійовичу

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Технологія розробки програмної системи рекомендацій для
знайомств людей та груп за інтересами у віртуальних спільнотах

Керівник проекту (роботи) Форкун Юрій Вікторович

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 01.09.2020 р. № 118

2. Строк подання студентом проекту (роботи) на кафедру 01.12.2020 р.

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

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

1 Аналіз сучасного стану рекомендаційних систем

2 Моделі та методи технології розрахунку рекомендацій людей та груп

3 Алгоритми та технологія для розрахунку рекомендацій людей та груп

4 Реалізація програмного забезпечення для розрахунку рекомендацій людей та груп

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

Презентаційні матеріали (слайди)

6. Консультанти розділів дипломного проєкту (роботи)

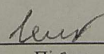
Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання «01» вересня 2020 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 - 07.09.2020	
2. Аналіз предметної області, актуальних технологій, моделей та методів	08.09 - 25.09.2020	
3 Розробка моделей та методів вирішення завдання	26.09 - 10.10.2020	
4. Робота над науковими публікаціями	11.10 - 20.10.2020	
5. Проектування архітектури системи для вирішення задачі, розробка вимог.	11.10 - 26.10.2020	
6 Детальний опис реалізації системи та тестування	27.10 - 15.11.2020	
7 Оформлення пояснювальної записки згідно вимог чинних стандартів	16.11 - 30.11.2020	
8 Попередній захист дипломної роботи	10.11.2020	
9 Перевірка роботи на наявність плагіату; норм контроль; брошурування пояснювальної записки; підготовка супровідних документів	01.12 - 04.12.2020	
10 Підготовка до захисту дипломної роботи	05.12 - 08.12.2020	

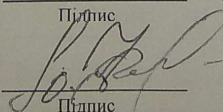
Студент


Підпис

А.О. Протоковський

Ініціали, прізвище

Керівник проєкту (роботи)


Підпис

Ю.В. Форкун

Ініціали, прізвище

РЕФЕРАТ

Тема дипломної роботи: Технологія розробки програмної системи рекомендацій для знайомств людей та груп за інтересами у віртуальних спільнотах.

Автор роботи: Протоковський Андрій Олексійович.

Керівник роботи: Форкун Юрій Вікторович.

Пояснювальна записка: 96 с., 15 рис., 4 табл., 3 дод., 26 джерел.

ГІБРИДНА ФІЛЬТРАЦІЯ, ПОДІБНІСТЬ ІНТЕРЕСІВ, РЕКОМЕНДАЦІЙНА СИСТЕМА, РЕКОМЕНДАЦІЯ ГРУП, РЕКОМЕНДАЦІЯ ЛЮДЕЙ.

Об'єктом дослідження є технологія розробки програмної системи рекомендацій людей та груп за інтересами у віртуальних спільнотах.

Предметом дослідження є моделі та підходи до формування списків рекомендацій людей та груп за інтересами у віртуальних спільнотах.

Метою дослідження є створення технології розробки програмної системи рекомендацій для знайомств людей та груп за інтересами у віртуальних спільнотах шляхом використання та розвитку актуальних методів розрахунку рекомендацій людей.

У роботі використані наступні методи дослідження та апаратура:

- аналіз, синтез, абстрагування, порівняння, опис, експеримент, вимірювання, тестування;
- програмні компоненти Visual studio 2017, Python 3.7;
- персональний комп'ютер.

Досліджено актуальні технології та методи розрахунку рекомендацій людей та груп з використанням інтересів. Запропоновано модифікацію методу розрахунку рекомендацій людей на основі єдності. Здійснено моделювання процесу розрахунку рекомендацій людей та груп з врахуванням інтересів. На основі результатів виконаних досліджень запропоновано алгоритм розрахунку рекомендацій людей, який застосовується для формування списку рекомендованих людей, що дозволяє з більшою точністю рекомендувати користувачам дружбу. Виконана програмна реалізація алгоритмів розрахунку рекомендацій людей та

розрахунку рекомендацій груп. Порівняно точність базового та модифікованого алгоритмів розрахунку рекомендацій людей.

Упровадження розробленого алгоритму та програмного засобу дає можливість рекомендувати людей у соціальних мережах на основі наявних соціальних зв'язків, активності користувача всередині системи, а також профілів інтересів користувачів.

Результати роботи можуть бути використані у будь-якій соціальній мережі, яка зберігає інформацію про соціальні зв'язки між користувачами, інформацію про активність користувачів у системі, а також інтереси користувачів або інформацію, з якої можна їх розрахувати.

Практична значимість отриманих результатів полягає у тому, що запропонована реалізація алгоритму розрахунку рекомендацій людей може бути застосована для формування списку рекомендованих людей, що дозволяє покращити досвід користувача соціальної мережі, тим самим задовольнивши потребу користувача у соціальних зв'язках.

У подальших дослідженнях розроблену технологію можна вдосконалити шляхом модифікації методів, які вона використовує, через введення нових факторів та модифікації методу розрахунку оптимальних ваг для кожного з цих факторів.

ABSTRACT

Master's thesis: Technology of developing a software system for recommending people and groups in virtual communities based on their interests.

Author: Protokovskyi Andrii Oleksiiovych.

Head of work: Forkun Yuriy Viktorovych.

Master's thesis consists of: 96 pages of general text, 15 graphics, 4 tables, 3 supplements, 26 sources.

HYBRID FILTERING, INTEREST SIMILARITY, RECOMMENDATION SYSTEM, RECOMMENDATION OF GROUPS, RECOMMENDATION OF PEOPLE.

The subject of the research is the technology of developing a software system for recommending people and groups in virtual communities based on their interests.

The study focuses on models and approaches for computing lists of recommended people and groups in virtual communities based on their interests.

The aim of the research is to create a technology for developing a software system for recommending people and groups in virtual communities based on their interests by using and developing relevant methods for calculating people's recommendations.

The following research methods and instruments have been used:

- analysis, synthesis, abstraction, comparison, description, experiment, measurement, testing;
- Visual Studio 2017 IDE, Python 3.7;
- personal computer.

Were explored relevant technologies and methods of calculation the list of recommended of people and groups based on interests. Modification of the method of calculation of recommendations of people based on cohesion was proposed. The process of calculating the recommendations of people and groups from the perspective of interests has been modelled. Based on the results of the research carried out, an algorithm for calculating the recommendations of people was proposed, which is used to form a list of recommended people, which makes it possible to recommend friends with greater precision. A software implementation of algorithms for calculating people's

recommendations and calculating group recommendations has been implemented. Compared to the accuracy of basic and modified algorithms for calculating the list recommended people.

The implementation of the developed algorithm and software makes it possible to recommend people in social networks on the basis of the existing social connections, the activity of the user within the system, as well as the profiles of the users' interests.

The results of the work can be used in any social network that preserves information on social connections between users, information on the activity of users in the system, as well as the interests of users or information from which they can be calculated.

The practical significance of this work lies in the fact that the proposed implementation of the algorithm for calculating the recommendations of people can be used to create a list of recommended people, allows to improve the experience of the user of the social network, thereby satisfying the user's need for social connections.

In further works, the developed technology can be improved by modifying the methods it uses, introducing new factors and modifying the method for calculating the optimal weights for each of these factors.

ЗМІСТ

Вступ.....	9
1 Аналіз сучасного стану рекомендаційних систем.....	12
1.1 Аналіз технологій рекомендаційних систем.....	12
1.2 Аналіз методів рекомендації людей.....	16
1.3 Аналіз методів рекомендації груп.....	23
1.4 Постановка задачі.....	27
1.5 Висновки.....	28
2 Моделі та методи технології розрахунку рекомендацій людей та груп.....	30
2.1 Функціональні моделі системи рекомендацій людей та груп.....	30
2.2 Методи для вирішення задач формування рекомендацій людей та груп..	32
2.2.1 Базовий метод рекомендації людей.....	32
2.2.2 Удосконалений метод рекомендації людей.....	35
2.2.3 Метод рекомендації груп.....	38
2.3 Висновки.....	39
3 Алгоритми та технологія для розрахунку рекомендацій людей та груп.....	42
3.1 Алгоритми розрахунку рекомендацій людей та груп.....	42
3.1.1 Алгоритм розрахунку рекомендацій людей.....	42
3.1.2 Алгоритм розрахунку рекомендацій груп.....	43
3.2 Розробка вимог до програмного забезпечення та даних, що використовуються для вирішення задачі.....	46
3.3 Проектування програмного забезпечення для вирішення задачі.....	48
3.4 Висновки.....	57
4 Реалізація програмного забезпечення для розрахунку рекомендацій людей та груп	60
4.1 Програмна реалізація.....	60
4.2 Обробка та аналіз тестових наборів даних.....	62
4.3 Постановка експерименту та отримані результати.....	69
4.4 Оцінка ефективності моделей та методів для вирішення задачі.....	73
4.5 Висновки.....	76

Висновки	79
Перелік джерел посилання	81
Додаток А Програмний код основних модулів.....	84
Додаток Б Копії наукових публікацій.....	90
Додаток В Презентаційні матеріали.....	102

ВСТУП

Системи рекомендацій людей використовуються як у відомих соціальних мережах, як Facebook, так і у менших Інтернет або мобільних додатках, наприклад Friendprobe. Рекомендаційні системи, що підбирають людей у соціальних мережах, забезпечують користувачеві досвід спілкування всередині мережі. Чим краще такого роду системи підбирають людей, тим кращий досвід переживає користувач і тим більше він задоволений.

Сьогодні активно відбувається пошук способів покращити якість рекомендацій таких систем. Перспективними є напрямки аналізу соціальних зв'язків (метод «друзі людей»), врахування інтересів (явних з профілю користувача, або не явних, отриманих з активності користувача) та врахування контексту (геолокація, мова спілкування тощо).

Метою дослідження є створення технології розробки програмної системи рекомендацій для знайомств людей та груп за інтересами у віртуальних спільнотах шляхом використання та розвитку актуальних методів розрахунку рекомендацій людей.

Завданнями дипломного проектування є:

- провести аналіз методів та інструментів для рекомендації людей;
- розробити метод розрахунку рекомендацій людей, опираючись на актуальні методи, шляхом перегляду списку факторів, які враховуються та перерахунком їх ваг у системі;
- створити технологію розробки програмної системи рекомендацій для знайомств людей та груп за інтересами у віртуальних спільнотах;
- провести практичну перевірку ефективності запропонованих нововведень у порівнянні з оригінальним методом;
- провести практичну перевірку ефективності запропонованого методу рекомендацій спільнот.

Об'єктом дослідження є технологія розробки програмної системи рекомендацій людей та груп за інтересами у віртуальних спільнотах.

Предметом дослідження є моделі та підходи до формування списків рекомендацій людей та груп за інтересами у віртуальних спільнотах.

В процесі дослідження літератури були використані наступні теоретичні методи:

- аналіз;
- синтез;
- абстрагування;
- порівняння.

Використання цих методів дозволило провести якісний аналіз стану справ за темами рекомендацій людей та спільнот, а також оцінити їх.

В процесі перевірки висунутих гіпотез були використані емпіричні методи:

- опис;
- експеримент;
- вимірювання;
- тестування.

Використання цих методів дозволило перевірити висунуті у ході дослідження гіпотези, оцінити вплив нових даних на якість сформованих рекомендацій та зробити висновок про ефективність запропонованих покращень у порівнянні із базовим алгоритмом.

З метою аналізу впливу різних факторів, за якими розраховуються рекомендації людей, були використані наступні емпіричні методи дослідження, як:

- експеримент;
- вимірювання.

Наукова новизна отриманих результатів:

- отримав подальший розвиток метод, на основі єдності, основна ідея якого полягає у поетапному вимірі міцності зв'язку у мережі і виявленні можливих зв'язків у цій мережі, які ще не створено; виявленні груп у мережі з використанням модулярності та рекомендації друзів, за рахунок включення фактору подібності інтересів та врахування двосторонньої сили зв'язку між користувачами, що дозволило покращити точність рекомендацій;

– набула подальшого розвитку формальна модель користувача сервісу розрахунку рекомендацій друзів, за рахунок її розширення профілями інтересів користувачів, що дало змогу покращити точність рекомендацій через точніше моделювання кожного окремого користувача;

– запропонована оригінальна технологія розробки програмної системи рекомендацій для знайомств людей та груп за інтересами у віртуальних спільнотах.

Практичне значення отриманих результатів:

– використання запропонованого підходу всередині соціальних мереж для формування точніших рекомендацій людям;

– поліпшення якості розрахованих рекомендацій вдосконаленого методу відносно оригінального.

Отримані результати можуть бути застосовані при розробці рекомендаційної системи людей та груп за інтересами у сервісах, які мають таку інформацію про користувача, як профіль його інтересів, список його друзів та список груп, у яких він перебуває.

За темою дипломної роботи зроблено дві публікації тез доповідей: одна видана на конференції «Інформація, комунікація, суспільство 2020», а друга – на конференції «Актуальні проблеми комп'ютерних наук».

1 АНАЛІЗ СУЧАСНОГО СТАНУ РЕКОМЕНДАЦІЙНИХ СИСТЕМ

1.1 Аналіз технологій рекомендаційних систем

Системи рекомендацій людей використовуються як у відомих соціальних мережах, як Facebook, так і у менших Інтернет або мобільних додатках, наприклад Friendprobe. Рекомендаційні системи, що підбирають людей у соціальних мережах, забезпечують користувачеві досвід спілкування всередині мережі. Чим краще такого плану системи підбирають людей, тим кращий досвід переживає користувач і тим більше він задоволений.

Сьогодні активно відбувається пошук способів покращити якість рекомендацій таких систем. Перспективними є напрямки аналізу соціальних зв'язків (метод «друзі людей»), врахування інтересів (явних з профілю користувача, або не явних, отриманих з активності користувача) та врахування контексту (геолокація, мова спілкування тощо).

Рекомендаційні системи такого плану можуть опиратися на різні загальні технології рекомендаційних систем. У наукових роботах зустрічаються різні класифікації технологій, за якими виконується розрахунок рекомендацій. У роботах [1-6] виділяють три загальних технології рекомендаційних систем:

- контентна фільтрація;
- колаборативна фільтрація;
- гібридна фільтрація.

Контентна фільтрація або фільтрація вмісту описується у роботах [1-11] як така, що формує рекомендації на основі схожості об'єктів. У класичному випадку така система опирається на профілі об'єктів, користувачів або на їх комбінацію. Ці профілі містять інформацію про якісні характеристики об'єктів та користувачів, за якими їх прямо ідентифікувати або принаймні класифікувати. Рекомендації формуються на основі історії попередньої взаємодії користувача з існуючими об'єктами за принципом підбору схожих об'єктів.

Для задач рекомендації людей контентна фільтрація може використовувати інформацію з профілю користувача, наприклад, вік, стать, інтереси тощо, для

рекомендації людей, базуючись на усередненому профілі всіх підтверджених людей користувача, або на подібності профілю одного користувача до профілів решти користувачів.

Система рекомендацій спільнот на основі контентної фільтрації у найпростішому варіанті може використовувати інформацію з профілів спільнот, у яких перебуває користувач, для підбору їм подібних.

Набір можливих характеристик, за якими буде оцінюватися те, чи схожий новий об'єкт на переглянуті раніше, може базуватися на ключових словах з опису об'єкта або категорій, на які розбиті об'єкти, якщо вони супроводжуються метаданими та профілями.

Сильні та слабкі сторони контентної фільтрації розглядаються у роботах [5,11]. Із сильних сторін цієї технології виділяють:

- можливість рекомендації нових об'єктів – вся необхідна інформація вноситься до системи при додаванні нового об'єкту;
- відносна простота реалізації.

До слабких сторін цієї технології відносять:

- робота з новими користувачі – виникає у системах, що керуються історією взаємодії користувача з об'єктами для формування рекомендацій;
- робота із старими непопулярними об'єктами – виникає у системах, що для оцінки об'єкта якісних характеристик керуються історією взаємодії користувачів з цим об'єктом;
- бульбашка фільтрів – замкнутість рекомендацій навколо одноманітних об'єктів із схожими характеристиками.

Колаборативна фільтрація у різних формах описується у роботах [1-12]. У класичному варіанті система, що реалізовує цю технологію розраховує рекомендації на основі схожості поведінки користувачів. Головною ідеєю колаборативної фільтрації є те, що користувачам із схожою поведінкою можна порекомендувати схожі об'єкти.

Для завдань рекомендації людей простим прикладом колаборативної рекомендаційної системи може бути взаємодія різних користувачів з одними і тими

ж об'єктами системи. На основі однакового або подібного списку переглянутих об'єктів може висуватися припущення про подібність поведінки користувачів, а отже, вірогідно і про подібність їх інтересів. Це означає, що їм можна рекомендувати однакові об'єкти.

Система підбору рекомендованих спільнот на основі колаборативної фільтрації у найпростішому вигляді може розраховувати подібність користувачів на основі списку спільнот, у яких вони перебувають, і формувати рекомендації із спільнот, членами яких є користувачі із подібною поведінкою (із подібним списком спільнот, у яких вони перебувають).

Сильні та слабкі сторони колаборативної фільтрації описані у роботах [5,11].
Із сильних сторін колаборативної фільтрації виділяють:

- відносно низька складність реалізації;
- низька прив'язаність до об'єктів, що рекомендуються – основним джерелом даних є історія взаємодії подібних користувачів із конкретними об'єктами у системі;
- адаптивність – нові рейтинги будуть автоматично враховані у подальших активаціях алгоритму.

Виділяють наступні слабкі сторони колаборативної фільтрації:

- погана робота з новими користувачами – для формування рекомендацій необхідна історія взаємодії користувача з об'єктами, на базі якої розраховується подібність між поведінкою користувачів;
- погана робота з новими та старими непопулярними об'єктами – для розрахунку цільової аудиторії об'єкта, необхідно мати історію його взаємодії з користувачами системи;
- розрідженість оцінок – велика кількість наявних об'єктів, за низької кількості фактичних отриманих оцінок від користувачів;
- необхідність тренувального набору даних – для старту системи необхідно мати початковий набір взаємодій користувачів з об'єктами системи;
- бульбашка фільтрів – замкнутість рекомендацій навколо одноманітних об'єктів, притаманних окремій групі користувачів.

Гібридна фільтрація описується у роботах [1-6,11,12] як така, що володіє характеристиками декількох технологій. У роботах [4,11,12] описуються різні види гібридизації. Так у роботі [12] гібридизацію поділяють на:

- weight – передбачає комбінування результатів роботи декількох рекомендаційних систем;
- switch – передбачає перемикання між рекомендаційними системами.

У роботі [11] ця класифікація доповнюється двома пунктами:

- mixed – за результатами окремої роботи декількох рекомендаційних систем відбувається розрахунок кінцевого значення ступеня рекомендованості конкретного об'єкта, наприклад, за принципом блискавки (найвищий бал отримує об'єкт з найбільшою оцінкою колаборативної рекомендаційної системи, другу за ним оцінку отримує об'єкт з найбільшою оцінкою рекомендаційної системи на основі фільтрації контенту тощо);
- cascade – передбачає, що результати роботи однієї рекомендаційної системи будуть вхідними даними для роботи іншої.

У роботах [2,4-6,11] підкреслюється покращення якості рекомендацій та усунення недоліків з отриманням переваг кожної окремої технології за рахунок гібридизації рекомендаційних систем. Проте, у всіх випадках доводиться розплачуватися більшою складністю у реалізації такої системи.

Для системи рекомендацій знайомств та груп за інтересами у віртуальних спільнотах найкраще підійде гібридна рекомендаційна система [18]. Шляхом об'єднання характеристик різних технологій вдасться зменшити або навіть повністю нівелювати недоліки окремих систем. Так у випадках, коли в профілі віртуальної спільноти бракує деякої інформації, система може спиратися на колаборативну фільтрацію. У випадку відсутності історії участі користувача у різних віртуальних спільнотах, система може формувати рекомендації спільнот на основі наявних профілів користувача та спільнот.

Якщо у системі відсутні дані про історію участі користувача у різних спільнотах, то система зможе рекомендувати знайомства на основі подібності профілів користувачів. У випадку наявності історії участі користувача у різних

віртуальних спільнотах, а також інформації в профілі користувача, система може комбінувати колаборативну та контентну фільтрації, що теоретично покращить якість рекомендацій.

Таким чином, найкращим варіантом гібридизації системи рекомендацій знайомств та груп за інтересами у віртуальних спільнотах буде *weight* – призначення вагового коефіцієнта для результатів роботи окремих рекомендаційних систем. Чим менше інформації для роботи конкретної системи, тим меншу вагу матиме сформований нею список рекомендацій відносно системи, для якої достатньо вхідних даних.

1.2 Аналіз методів рекомендації людей

Дослідження літератури напрямку рекомендаційних систем, як підкласу систем фільтрації інформації, показало, що цим питанням активно займаються науковці всього світу. У підрозділі проаналізовано роботи [19-24], як такі, що використовують актуальні методи розрахунку рекомендацій людей.

У роботі [19] система розраховує рекомендації, розраховуючи подібність за 5-ма базовими категоріями: музика, фільми, спорт, ігри та книги. У розрахунку беруть участь найбільший інтерес користувача, а також інші, які він може обрати окремо. Інтереси користувачів розраховуються шляхом аналізу інформацію з профілю користувача соціальної мережі Facebook. У процесі реєстрації користувач дає дозвіл на синхронізацію свого облікового запису із обліковим записом Facebook, з якого система дістає необхідні для роботи значення (вподобання, місце знаходження тощо) та зберігає їх у своїй базі даних для подальшого аналізу. Із зібраної інформації шляхом її аналізу, як тексту, алгоритмом LDA, чи йому подібним, система обчислює відношення користувача до кожної з основних категорій. Далі запропонована система обчислює подібність двох користувачів та формує матрицю подібності користувачів, на основі якої будуть формуватися

рекомендації людей за принципом «чим більш подібні два користувачі, тим більша ймовірність виникнення між ними дружби».

До переваг цієї системи можна віднести:

- відносно невелику обчислювальну складність, оскільки розрахунок подібності буде відбуватися всього за 5-ма категоріями;
- відносно невеликий об'єм даних, які потрібно зберігати;
- мала варіативність вподобань забезпечує високі шанси знайти співпадіння серед користувачів системи.

До недоліків цього методу можна віднести:

- малу кількість категорій, яка може недостатньо добре описувати користувача і як наслідок негативно впливати на точність рекомендацій;
- необхідність отримувати доступ до облікового запису користувача у соціальні мережі Facebook, що може відштовхнути окремих користувачів.

У роботі [20] пропонується підхід рекомендацій людей, базуючись на їх схожості до персонажів телепрограм. Розрахунок подібності людей до персонажів ТВ програм у запропонованій системі відбувається шляхом опитування їх людей з метою виявлення тих персонажів, на яких цільовий користувач схожий найбільше. Головна ідея цієї системи – якщо два персонажі ТВ програм дружать, то людям, які на них схожі, треба пропонувати дружбу.

Важливим недоліком цієї системи є обмеженість на телепрограмах, оскільки не завжди можна знайти якісний опис персонажів телепрограм.

Робота [21] рекомендації людей формуються шляхом аналізу стилю життя користувачів на основі даних із сенсорів їх телефонів (акселерометр та гіроскоп). Також ця система передбачає використання GPS для визначення віддаленості користувачів. Окрім стилю життя система аналізує граф зв'язків між користувачами для оцінки їх здатності до дружби з тими, хто має інший стиль життя. Ще одним важливим фактором цієї системи врахування зворотного зв'язку з користувачами. Так, користувач може оцінити запропонований йому варіант дружби, а саме іншого користувача. Отримане значення буде враховане системою

у матриці, що зберігає оцінки здатності користувача до дружби із іншими людьми, які мають відмінний стиль життя.

До переваг цієї системи можна віднести:

- врахування такої важливої характеристики, як стиль життя;
- врахування зворотного зв'язку у подальших ітераціях.

До недоліків цієї системи можна віднести:

- необхідність тривалого спостереження за користувачем;
- необхідність персональної інформації, яку не кожен користувач може або готовий надати.

У роботі [22] розглядається загальний спосіб розрахунку рекомендацій з врахуванням контексту та з виділенням вподобань користувачів та ваги думок інших користувачів(людей) через факторизацію або розклад матриць взаємодії користувачів, користувачів та об'єктів. Виділення явних властивостей об'єктів відбувається шляхом аналізу їх профілів алгоритмом LDA чи його аналогом. Виділення неявних властивостей об'єктів відбувається шляхом розкладу матриць оцінок об'єктів користувачами та матриці впливу або ваг думок користувачів один відносно одного.

У роботі [23] пропонується виділення інтересів із тегів, що використовує користувач, з використанням ієрархії тегів. У цій системі рекомендації формуються через розрахунок подібності користувачів з використанням виділених інтересів з урахуванням кількості спільних людей.

Робота [24] пропонує розраховувати рекомендації використовуючи фізичний (місця знаходження, часу, профілю тощо) та соціальний (мережа наявних людей) контексти. Фінальна оцінка вірогідності дружби розраховується як зважена сума цих двох контекстів. Основним недоліком цієї системи є відсутність точного опису фізичного та соціального контекстів.

У роботі [25] пропонується метод розрахунку рекомендацій людей з урахуванням кількості спільних людей, кількості спільних груп та кількості спільних програм, якими користувалися користувачі. Ці три фактори зважуються і з них розраховується одностороння оцінка дружби. Рекомендація щодо можливості

дружби формується, якщо середнє значення односторонніх оцінок дружби між користувачами більше за порогове значення, яке розраховується для користувача, для якого обчислюється рекомендація, як середнє арифметичне всіх оцінок дружб для користувачів, з якими він вже дружить.

Методи розрахунку подібності є невід’ємною частиною практично будь-якої рекомендаційної системи. Такого роду методи можуть бути як основними так і допоміжними. Роботи [1,7] пропонують цілу низку методів для розрахунку подібності: косинус подібності, коефіцієнт кореляції Пірсона, Евклідова відстань та ін. Робота [2] описує використання різних способів розрахунку подібності для різних ситуацій. Так у ній пропонується використовувати косинус подібності у випадках, коли дані розріджені (деякі координати вектору ознак рівні 0). Інакше рекомендується використовувати на вибір Евклідову відстань або Манхеттенську відстань. У роботі [19] значення косинуса подібності використовується для виявлення користувачів з подібним стилем життя. Евклідова відстань використовується у роботі [13] для розрахунку подібності користувачів за їх інтересами.

У загальному випадку косинус подібності це числове значення косинуса кута між двома ненульовими векторами. Чим більший кут між двома векторами, тим менше значення косинуса його кута і тим менш подібними будуть вважатися ці два вектори. Величина косинуса подібності може приймати значення у проміжку $[-1;1]$, де -1 відповідає двом діаметрально протилежним за напрямком векторам, 0 відповідає перпендикулярним векторам, а 1 відповідає двом векторам, напрямки яких повністю співпадають, незалежно від їх довжини.

Для двох користувачів або груп, що представлені двома векторами ознак $A = (a_1, a_2, \dots, a_n)$ та $B = (b_1, b_2, \dots, b_n)$, косинус подібності буде обчислюватися за формулою (1):

$$\text{cossim}_{A,B} = \frac{\sum_{k=1}^n a_k b_k}{\sqrt{\sum_{k=1}^n a_k^2} \sqrt{\sum_{k=1}^n b_k^2}}, \quad (1)$$

де $\text{cossim}_{A,B}$ – значення, що відображає подібність користувачів або груп;

A та B – вектори ознак користувачів або груп, що порівнюються;

n – кількість координат або ознак векторів;

a_k та b_k – k -ті координати векторів, які означають вагу відношення користувача або групи до конкретної ознаки A та B відповідно.

Формулу (1) можна використовувати як для розрахунку подібності користувачів (роботи [19,21,24]) та для розрахунку подібності об'єктів (робота [2]). Єдина вимога – щоб користувачі або групи, що порівнюються, були у одному просторі факторів, тобто якщо перший користувач характеризується атрибутами (вік, стать, мова тощо), то і другий користувач характеризується точно таким самим набором атрибутів. Аналогічно для двох груп, що порівнюються.

Як приклад буде доцільно привести один з можливих способів використання цієї величини у системі рекомендацій людей, що орієнтується на вподобання або інтереси користувачів, у якій користувачі представлені у вигляді векторів однієї розмірності, координатами яких є конкретні вподобання або інтереси, а числові значення означають міру важливості конкретного інтересу для обраного користувача. У такому випадку косинус подібності між двома векторами інтересів користувачів дозволить зрозуміти наскільки збігаються вподобання користувачів, що порівнюються, а отже, й оцінити чи варто рекомендувати їм знайомство. Косинус подібності двох користувачів із максимально схожими інтересами буде числовим значенням, близьким до 1, у той час як косинус подібності для двох користувачів з діаметрально протилежним набором інтересів, буде наближатися до -1. Крім цього косинус подібності може використовуватися як допоміжний інструмент, наприклад для задачі кластеризації для обчислення згрупованості всередині кластерів.

Евклідова відстань фігурує у роботах [2,7,16] як метод розрахунку подібності векторів, що представляють різних користувачів. Чим більша відстань між двома векторами, тим більше вони відрізняються. Міра подібності двох векторів ознак користувачів або груп $A = (a_1, a_2, \dots, a_n)$ та $B = (b_1, b_2, \dots, b_n)$ виражається через Евклідову відстань за формулою (2).

$$euclidsim_{A,B} = A - B = \sqrt{\sum_{k=1}^n (a_k - b_k)^2}, \quad (2)$$

де $euclidsim_{A,B}$ – значення, що відображає подібність векторів ознак користувачів або груп A та B ;

A та B – вектори ознак, що порівнюються;

n – кількість ознак векторів або їх розмірність;

a_k та b_k – ступінь, у якій користувач або об'єкт задовольняє k -тій ознаці векторів A та B відповідно.

На нашу думку, для кращої оцінки подібності результати розрахунків за формулами (1) та (2) варто скомбінувати шляхом їх множення за формулою:

$$sim_{A,B} = cossim_{A,B} \times euclidsim_{A,B} \quad (3)$$

де $cossim_{A,B}$ – косинус подібності;

$euclidsim_{A,B}$ – Евклідова відстань.

За результатами аналізу наукових публікацій, проведеному у розділі 1.2, можна виділити наступні методи розрахунку рекомендацій людей:

- розрахунок подібності на основі інтересів;
- розрахунок подібності на основі контексту;
- розрахунок подібності на основі стилю життя;
- розрахунок подібності на основі списку наявних людей.

Для роботи описаного списку методів необхідний специфічний набір даних. Так, для врахування інтересів користувачів необхідно мати профіль користувача, який містить інформацію про вподобання користувача. Такий профіль може заповнюватися самим користувачем, або ж розраховуватися з використанням інформації про поведінку користувача у системі, наприклад, на основі історії взаємодії користувача з об'єктами системи, на основі списку людей тощо.

Поняття «контексту» доволі розмите. У розглянутих роботах під контекстом розуміється інформація із сенсорів мобільного пристрою користувача (акселерометр, гіроскоп, GPS тощо), час доби у місці, де знаходиться користувач, те, чим користувач займається у момент запиту тощо.

Під стилем життя розуміється обчислений із історії трекінгу фізичної активності профіль користувача. Цей профіль може містити декілька стилів життя, яким відповідає користувач. Самі ж стилі життя можуть розраховуватися із історій фізичної активності всіх користувачів шляхом пошуку шаблонів, що повторюються серед профілів стилів життя всіх користувачів системи.

Розрахунок подібності з врахуванням наявних людей може відбуватися у різні способи. Типовими способами є розрахунок подібності через обчислення кількості спільних людей та розрахунок подібності через порівняння усередненого профілю інтересів людей користувачів. У випадку обчислення подібності через підрахунок кількості спільних людей, міра подібності між користувачами може визначатися як відношення кількості спільних людей до загальної кількості людей користувача. Цей підхід не вимагає великих обчислювальних затрат, але у такому випадку отримане значення відображає односторонню подібність. Це означає, що для обчислення подібності за кількістю спільних людей потрібно розрахувати односторонні значення подібності для обох користувачів та зважити їх для отримання кінцевого значення, на яке можна опиратися у процесі прогнозування.

Розробка некомерційних систем рекомендації людей, що використовують ці методи значно ускладнюється тим фактом, що для навчання, якщо ця система заснована на моделях машинного навчання, або для тестування така система потребує набору даних (англ. «dataset»), який міститиме детальну історію взаємодії користувача з об'єктами системи, зв'язки між користувачами, дані про пересування кожного користувача тощо. Набори даних, що задовольняють цим вимогам, надзвичайно складно знайти у відкритому доступі, оскільки вони можуть містити персональну інформацію про користувача або достатню кількість даних, які не вважаються персональними, але які можуть слугувати для розпізнавання реальної особи користувача навіть якщо справжні персональні дані були видалені або

замінені на умовні позначення, з яких не можливо вирахувати оригінальні дані. Тому, з метою забезпечення можливості протестувати розроблену систему, доцільно проектувати її так, щоб вона враховувала лише ту інформацію, яка міститься у наборах даних, які знаходяться у відкритому доступі, або передбачала опціональну можливість враховувати фактори, які відсутні у загальнодоступних наборах даних, але які можуть міститися у реальних наборах даних.

1.3 Аналіз методів рекомендації груп

З точки зору рекомендаційних систем, формування рекомендацій груп за інтересами є тривіальною задачею рекомендації об'єктів з відомими властивостями, або профілями об'єктів. Серед актуальних методів, які можна використати для рекомендацій груп за інтересами можна виділити:

- аналіз списку груп, членами яких є друзі користувача;
- аналіз профілю користувача та профілю спільнот;
- порівняння профілю користувача із усередненим профілем всіх користувачів, що входять до спільноти;
- використання інформації про активність користувача у спільнотах, у яких він вже перебуває.

У найпростішому випадку рекомендації груп шляхом аналізу списку груп, членами яких є друзі користувача, будуть розраховуватися наступним чином:

- розрахунок подібності між користувачем та його друзями;
- рекомендація груп, членами яких є друзі користувача, профіль яких найбільше схожий на профіль користувача.

Серйозним недоліком цього підходу є те, що рекомендації груп сильно обмежені колом спілкування користувача, а отже, для користувача, який має невелику кількість друзів, сформувані якісні рекомендації буде проблематичним або неможливим.

Рекомендація груп через порівняння профілів групи та користувача у найпростішому випадку матиме вигляд:

- профіль користувача містить вагові коефіцієнти для сталого набору властивостей (наприклад інтересів), якими володіють спільноти;
- профіль спільноти містить набір інтересів, яким вона відповідає;
- прогнозована оцінка спільноти користувачем розраховується шляхом множення вектора вподобань користувача та вектора інтересів, яким відповідає спільнота. Чим більше отримане значення, тим краще спільнота задовольняє інтереси користувача.

Найпростішим варіантом системи рекомендації груп, що базується на порівнянні профілю користувача із усередненим профілем члена спільноти може слугувати наступний:

- для кожної спільноти, яка має визначений список членів, розраховується усереднений профіль користувача спільноти;
- в процесі формування рекомендацій система порівнює різницю між профілем цільового користувача та профілем усередненого члена спільноти;
- якщо значення цих двох профілів достатньо співпадають, система рекомендує цю спільноту цільовому користувачеві.

Врахування рекомендаційною системою активності користувача всередині спільноти, може відбуватися у наступний спосіб:

- для користувача обчислюється його активність у всіх спільнотах, до яких він входить (наприклад кількість часу, який він провів на сторінці спільноти, кількість візитів на сторінку спільноти тощо);
- із профілів спільнот, для яких була обчислена активність користувача, виділяються ключові фактори (наприклад інтереси, яким вона відповідає);
- використовуючи активність як ваговий коефіцієнт обчислюються ваги всіх факторів, якими володіють спільноти;
- формується список спільнот, що найкраще задовольняють розраховані ваги факторів.

У підрозділі були розглянуті наступні методи розрахунку рекомендацій груп:

- аналіз груп, у яких перебувають друзі користувача;
- використання профілів з інтересами користувача та профілів з інтересами, яким відповідає конкретна група;
- порівняння профілю з інтересами користувача з усередненим профілем типового користувача спільноти;
- врахування інформації про активність користувача всередині конкретних спільнот, у яких він перебуває.

Використання аналізу груп, у яких перебувають друзі користувача, може мати два ефекти. З одного боку, шляхом такого аналізу можна формувати рекомендації об'єктів, які сам користувач не спробував би і, таким чином, розширити його коло інтересів. З іншого боку, якщо не правильно враховувати вплив цього фактору на список рекомендацій, то їх якість може впасти, у порівнянні з методом, який не використовує такий аналіз. Для правильного підбору ваги цього фактору необхідно натренувати модель на великому обсязі реальних даних. Для врахування особливостей поведінки жителів різних регіонів цей набір даних має містити інформацію про користувачів з різних куточків планети та включати такого роду параметри, як фактичне місце знаходження користувача, місце знаходження групи, мережа довіри між користувачами тощо. Більшість з необхідних параметрів буде відсутня у наборах даних, що знаходяться у відкритому доступі, тому розробка такого методу може мати лише теоретичний характер, а отже, оцінити реальну продуктивність такого методу буде складно.

Метод рекомендації груп, що базується на порівнянні профілю користувача із усередненим профілем типового користувача групи сильно залежить від кількості параметрів з профілів користувачів, які враховує система. До такої інформації належать наступні категорії: інтереси користувача, його вік, місце проживання та соціальна активність. Як правило у наборах даних, які знаходяться у відкритому доступі, відсутні або присутні частково такого роду категорії, як вік, місце проживання та соціальна активність. Це означає, що система, яка покладається на цей метод, як на основний, зможе бути протестована лише за наявності набору даних з усіма перерахованими категоріями. Такого роду набори

даних складно знайти у відкритому доступі, тому мова може йти лише про розробку такої системи у кооперації з організацією, яка має необхідні дані.

Для точної рекомендації груп на основі аналізу активності користувача у групах, у яких він перебуває, необхідно мати цілу низку даних, наприклад, як часто користувач заходить на сторінку групи, скільки контенту групи він переглядає, які ставить оцінки, скільки контенту всередині групи він генерує тощо. Від повноти вхідних даних залежить точність рекомендацій. У наборах даних, що знаходяться у відкритому доступі складно знайти настільки повний набір факторів. Зазвичай у таких наборах даних присутня інформація лише про участь користувачів у різних спільнотах. Для повноцінного тестування такого підходу цього єдиного фактора недостатньо, що не дозволяє винести такого роду системі справедливу оцінку.

Рекомендація груп на основі профілів користувачів та профілів груп, на нашу думку, є найкращим способом рекомендацій для проектованої системи. Використовуючи профілі користувачів та профілі груп можна спрогнозувати наскільки користувачеві може сподобатися група. Так, шляхом множення матриці профілів користувачів на матрицю профілів груп, можна спрогнозувати наскільки користувачеві може сподобатися група, при чому ця оцінка буде спиратися на актуальні інтереси користувача. Для покращення точності рекомендацій, що розраховуються цим методом, найкраще підійде не просто профіль з інтересами користувача, який він заповнює самотужки, а розрахований профіль його інтересів, наприклад, із списку груп, у яких він перебуває. Максимальної точності можна досягти шляхом зважено комбінації введеного користувачем профілю із розрахованим. На нашу думку, саме цей спосіб найкраще піддається тестуванню, враховуючи набір факторів, доступних у наборах даних, які знаходяться у відкритому доступі.

У випадку, якщо у системі визначено лише один профіль інтересів, який або введений користувачем, або розрахований з його активності, то прогнозування має повністю опиратися саме на наявний профіль, ігноруючи відсутній.

1.4 Постановка задачі

Системи рекомендацій людей використовуються у різних за розміром веб-спільнотах та соціальних мережах. Рекомендаційні системи, що підбирають людей, забезпечують користувачеві досвід спілкування всередині мережі. Чим краще такого роду системи підбирають людей, тим кращий досвід переживає користувач і тим більше він задоволений.

У дослідженнях по цій тематиці, які активно проходять, перспективними є напрямки аналізу соціальних зв'язків (метод «друзі людей»), врахування інтересів та врахування контексту (геолокація, мова спілкування тощо).

Система рекомендацій людей на основі всіх цих факторів може краще описувати кожного конкретного користувача і як наслідок така багата на фактори формальна модель користувача має забезпечити краще прогнозування нових людей всередині реальної віртуальної спільноти або соціальної мережі.

Об'єктом дослідження є технологія розробки програмної системи рекомендацій людей та груп за інтересами у віртуальних спільнотах. Через дослідження об'єкту дипломної роботи можна оцінити сильно та слабкі сторони існуючих рішень та запропонувати способи покращити їх.

Предметом дослідження є моделі та підходи до формування списків рекомендацій людей та груп за інтересами у віртуальних спільнотах. Дослідження моделей та підходів до формування списків рекомендацій людей та груп за інтересами у віртуальних спільнотах допоможе зрозуміти які актуальні методи використовуються у рекомендаційних системах такого типу та запропонувати покращення, які, на нашу думку, мають позитивно вплинути на точність рекомендацій.

Метою дослідження є створення технології розробки програмної системи рекомендацій для знайомств людей та груп за інтересами у віртуальних спільнотах шляхом використання та розвитку актуальних методів розрахунку рекомендацій людей.

Для досягнення цієї мети необхідно виконати наступні завдання:

- провести аналіз методів та інструментів для рекомендації людей;

- розробити метод розрахунку рекомендацій людей, опираючись на актуальні методи, шляхом перегляду списку факторів, які враховуються та перерахунком їх ваг у системі;
- створити технологію розробки програмної системи рекомендацій для знайомств людей та груп за інтересами у віртуальних спільнотах;
- провести практичну перевірку ефективності запропонованих нововведень у порівнянні з оригінальним методом;
- провести практичну перевірку ефективності запропонованого методу рекомендацій спільнот.

Створена технологія розробки програмної системи рекомендацій для знайомств людей та груп за інтересами у віртуальних спільнотах, на нашу думку, дозволить формувати точніші рекомендації людей, у порівнянні із тією, яку вирішено покращити через введення нових факторів та зміни їх ваг.

1.5 Висновки

За результатами проведеного дослідження визначено, що найкращий результат має дати гібридна рекомендаційна система.

Серед розглянутих актуальних методів рекомендації людей числяться: аналіз соціальних зв'язків, врахування інтересів користувачів та врахування контексту.

Були розглянуті наступні методи рекомендації груп: як аналіз груп, у яких перебувають друзі користувача, використання профілів з інтересами користувача та профілів з інтересами, яким відповідає конкретна група, порівняння профілю з інтересами користувача з усередненим профілем типового користувача спільноти та врахування інформації про активність користувача всередині конкретних спільнот, у яких він перебуває.

Із методів розрахунку подібності були розглянути метод розрахунку косинуса подібності та метод розрахунку Евклідової відстані. Для того, щоб усунути їх слабкі сторони вирішено скомбінувати ці два методи для розрахунку

фінального значення подібності у системі. Через добуток значень, отриманих від кожного окремого методу, очікується отримати точніше значення подібності, у порівнянні із значенням, отриманим від кожного окремого методу.

Після аналізу вирішено удосконалити алгоритм, запропонований у роботі [25], оскільки, на нашу думку, з усіх розглянутих у розділі 1.2 саме цей алгоритм, оскільки він враховує участь користувача у різних групах та соціальні зв'язки, що якнайкраще підходить для нашого завдання. У цьому алгоритмі вирішено змінити спосіб розрахунку односторонньої оцінки дружби між користувачами так, щоб враховувати не кількість програм, якими користуються користувачі, а подібність профілів з інтересами користувачів. Також вирішено змінити спосіб обчислення рекомендації дружби шляхом врахування порогового значення обох користувачів. Крім цього вирішено розширити запропонований спосіб розрахунку порогового значення шляхом врахування оцінок дружби людей, що найменше схожі з користувачем.

Як метод розрахунку рекомендацій груп вирішено використати прогнозування оцінок для кожної групи, які будуть розраховуватися на основі профілів вподобань користувачів та профілів вподобань, яким відповідають групи.

На основі цих методів вирішено створити власну технологію розробки програмної системи рекомендацій для людей та груп за інтересами у віртуальних спільнотах, яка б, у порівнянні із базовою, враховувала нові фактори.

Після реалізації система має бути порівняна з оригінальним алгоритмом у таких показниках, як точність передбачень, середньоквадратична помилка та інші параметри порівняння точності, які буде вирішено застосувати.

2 МОДЕЛІ ТА МЕТОДИ ТЕХНОЛОГІЇ РОЗРАХУНКУ РЕКОМЕНДАЦІЙ ЛЮДЕЙ ТА ГРУП

2.1 Функціональні моделі системи рекомендацій людей та груп

На нашу думку, для кращого моделювання системи рекомендацій людей та груп, доцільно буде побудувати окремі функціональні моделі для модуля рекомендації людей та для модуля рекомендації груп. Таким чином, вдасться краще описати особливостей процесів рекомендації людей та груп.

Функціональна модель процесу розрахунку рекомендацій людей, що містить основні процеси, наведена на рисунку 2.1.

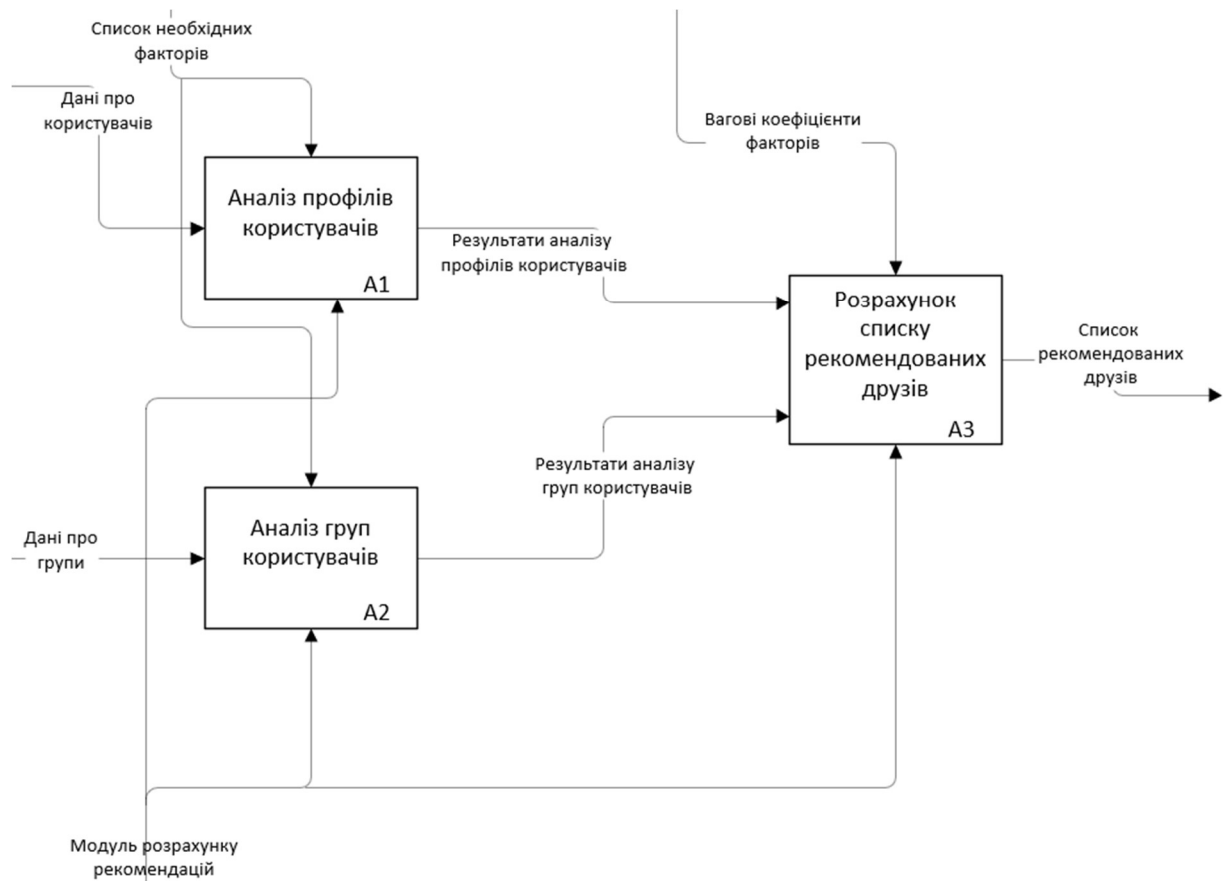


Рисунок 2.1 – Функціональна модель процесу рекомендацій людей

Запропонована модель складається із трьох основних пунктів: аналіз профілів користувачів, аналіз груп користувачів та розрахунок списку рекомендованих людей. Під аналізом профілів користувачів розуміється

зчитування, за потреби розрахунок, необхідних для розрахунку рекомендацій факторів. Набір необхідних факторів може бути як постійною величиною, так і змінною, залежною від наявних у системі даних або її налаштувань. У випадку системи, що проектується, передбачається, що система буде використовувати такого роду дані з профілю, як користувачі із списку людей та інтереси користувача.

Під аналізом груп користувачів розуміється підрахунок груп, у яких перебувають обидва користувача, а також підрахунок груп, у яких перебуває лише користувач, для якого розраховуються рекомендації.

Під розрахунком списку рекомендованих людей розуміється використання інформації, зібраної з профілів користувачів та списку груп, у яких вони перебувають, з метою визначення подібності між користувачами та оцінки можливості дружби між ними.

Таким чином, модель процесу розрахунку рекомендації людей буде враховувати спільних людей, спільні групи, у яких перебувають користувачі, котрі порівнюються, а також інтереси користувачів.

Функціональна модель процесу розрахунку рекомендацій груп, що містить основні процеси, наведена на рисунку 2.2.

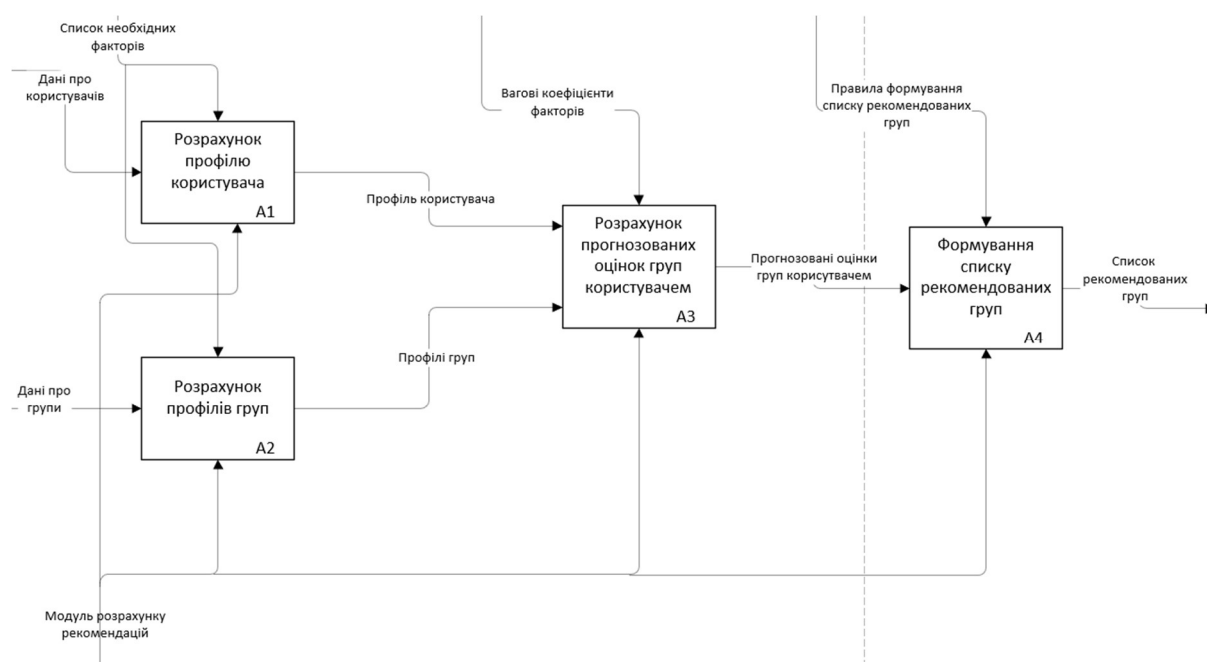


Рисунок 2.2 – Функціональна модель процесу рекомендацій груп

Запропонована модель складається із чотирьох основних пунктів: розрахунок профілю користувача, розрахунок профілів груп, розрахунок прогнозованих оцінок груп користувачів та формування списку рекомендованих груп. Під розрахунком профілю користувача розуміється розрахунок інтересів користувача із груп, у яких він перебуває. Набір інтересів користувача залежить від того списку груп, у яких він перебуває.

Під розрахунком профілів груп розуміється формування списку інтересів, яким відповідає група.

Під розрахунком прогнозованих оцінок розуміється обчислення добутку вектору, що містить ваги інтересів користувача та матрицю, що містить ваги інтересів, яким відповідає кожна група. У випадку, якщо необхідно розрахувати оцінку для окремої групи, на вектор ваг інтересів користувача множиться не вся матриця з вагами інтересів для груп, а лише деякий її рядок, який відповідає групі, для якої необхідно обчислити прогнозовану оцінку.

Під формуванням списку рекомендацій розуміється відбір із отриманого раніше вектору з прогнозованими оцінками користувача тих груп, для яких ці оцінки найбільші. Кількість груп, які необхідно відібрати, встановлюється вручну.

Таким чином, модель процесу розрахунку рекомендації груп буде враховувати інтереси користувачів та інтереси, яким відповідають групи. При цьому список можливих інтересів користувачів та груп ідентичний.

2.2 Методи для вирішення задач формування рекомендацій людей та груп

2.2.1 Базовий метод рекомендації людей

У цьому пункті вирішено розглянути запропонований у 2017 році у роботі [25] метод розрахунку рекомендації людей, модифікувати його та перевірити вплив запропонованих змін на результативність рекомендаційної системи, тому опишемо оригінальний метод та запропоновано покращення, які, на нашу думку, мають позитивно вплинути на якість та точність сформованих рекомендацій.

Метод, який вирішено взяти за основу, має наступні етапи:

- обчислюється сила зв'язків між користувачами, які є друзями;
- будується зважений граф, у якому вершинами є користувачі, які є друзями, а вагами ребер є обчислені сили зв'язків між ними;
- граф розбивається на групи з використанням методу Лювена;
- рекомендації нових людей формуються з користувачів однієї групи, якщо такого роду нерозглянуті варіанти існують;
- рекомендація людей формується з користувачів, інших груп, для яких значення зв'язку найбільше.

Сила зв'язків між користувачами обчислюється за формулою (4):

$$R(T, T_1) = \lambda_1 F_1(T, T_1) + \lambda_2 F_2(T, T_1) + \lambda_3 F_3(T, T_1), \quad (4)$$

де $R(T, T_1)$ – значення сили зв'язку між користувачами T, T_1 ;

F_1 – відношення кількості спільних людей до загальної кількості людей користувача T ;

F_2 – відношення груп, у яких перебувають обидва користувачі, до кількості груп, у яких перебуває лише користувач T ;

F_3 – відношення кількості програм, якими користуються обидва користувачі, до загальної кількості програм, якими користується користувач T ;

λ_1, λ_2 та λ_3 – це вагові коефіцієнти для кожного з параметрів.

Змінні F_1, F_2 та F_3 розраховуються за формулами (5), (6) та (7) відповідно.

$$F_1 = \frac{|Fr(T) \cap Fr(T_1)|}{|Fr(T)|}, \quad (5)$$

$$F_2 = \frac{|Gr(T) \cap Gr(T_1)|}{|Gr(T)|}, \quad (6)$$

$$F_3 = \frac{|Apps(T) \cap Apps(T_1)|}{|Apps(T)|}, \quad (7)$$

де $Fr(T)$ та $Fr(T_1)$ – множини людей обох користувачів;

$Gr(T)$ та $Gr(T_1)$ – множини груп, у яких перебувають користувачі;

$Apps(T)$ та $Apps(T_1)$ – множини програм, якими користувалися користувачі.

Для побудови зваженого графу, у якому вершинами є користувачі, а вагами ребер є обчислені сили зв'язків між користувачами, необхідно ці зв'язки встановити. Можливість встановлення зв'язку між двома користувачами T та T_1 визначається з використанням наступної нерівності:

$$\frac{R(T, T_1) + R(T_1, T)}{2} > threshold \quad (8)$$

де $R(T, T_1)$ – оцінка зв'язку між користувачами T та T_1 ;

$R(T_1, T)$ – оцінка зв'язку між користувачами T_1 та T ;

$threshold$ – порогове значення оцінки зв'язку.

Порогове значення, яке регулює можливу різницю між оцінками зв'язку, розраховується за формулою (9).

$$threshold = \frac{\sum_{i \in Fr(T)} R(T, T_i)}{|Fr(T)|} \quad (9)$$

де $threshold$ – порогове значення оцінки зв'язку;

$Fr(T)$ – множина людей користувача T .

Використовуючи формулу (4) система будує зважений граф, у якому ваги ребер визначаються як оцінка зв'язку між двома користувачами, що знаходяться у списку людей один одного. Після побудови графа зв'язків між користувачами, які є друзями, система використовує метод Лювена для розпізнавання груп користувачів. Після цього розбиття на групи система буде формувати рекомендації враховуючи користувачів, що знаходяться всередині однієї групи. Якщо всередині групи не залишилось прийнятних для користувача варіантів, то рекомендації

будуть формуватися із користувачів інших груп, для яких значення оцінки дружби найвище. Після розбиття графу користувачів на групи, система переходить до наступного етапу – прогнозування нових ребер, які будуть означати дружбу, використовуючи формули (8) та (9).

Для кожного із списку людей всіх людей користувача система розраховує необхідність створювати ребро між двома користувачами, використовуючи формулу (8). Якщо ця умова виконується, то система створює нове ребро між користувачами тим самим пропонуючи дружбу.

В пункті 2.2.1 був розглянутий метод, який вирішено взяти за основу та модифікувати з метою покращення результатів його роботи. У своїй роботі описаний метод покладається на розрахунки, що виконуються за формулами (4), (8) та (9) та на метод Лювена для виявлення груп всередині графа.

На нашу думку, цей метод можна покращити шляхом модифікації формул (4), (8) та (9), а саме введенням у них нових змінних або заміни деяких факторів, які враховує оригінальний метод на нові.

2.2.2 Удосконалений метод рекомендації людей

Взятий за основу метод має наступний список недоліків:

- він не враховує інтересів користувачів;
- інформація про програми, якими користуються користувачі відсутня у більшості соціальних мереж;
- при розрахунку можливості дружби за формулою (8) враховується одностороннє порогове значення;
- розрахунок порогового значення за формулою (9) не враховує різноманітність людей користувача.

На нашу думку, розширення списку факторів, що враховуються у процесі оцінки можливості дружби, подібністю інтересів користувачів має позитивно вплинути на якість рекомендацій. Подібність інтересів користувачів це важливий

фактор, ігнорування якого у процесі рекомендації людей може збільшувати кількість неякісних рекомендацій.

Інформація про програми, якими користуються користувачі, не знаходиться у відкритому доступі. Дізнатися таку інформацію можна лише опитавши користувача. Проте, у такому випадку точність рекомендацій буде залежати від того, наскільки точний список програм зможе назвати користувач. Крім цього список програм, які використовував користувач можуть сигналізувати про інтереси користувача, проте, для того, щоб розрахувати інтереси користувача за цим параметром необхідно мати детальні дані про всі можливі програми, які можуть використовувати користувачі. З цієї причини доцільніше замість цього параметру враховувати саме інтереси користувачів.

З урахуванням написаного, формулу (4) слід змінити у наступний спосіб: прибрати фактор програм, які використовував користувач та поставити на його місце фактор подібності інтересів користувачів. У результаті формула (4) прийме наступний вигляд:

$$R'(T, T_1) = \lambda_1 F_1(T, T_1) + \lambda_2 F_2(T, T_1) + \lambda'_3 F'_3(T, T_1) \quad (10)$$

де $R'(T, T_1)$ – оцінка зв'язку між користувачами T та T_1 ;

λ_1, λ_2 та λ'_3 – вагові коефіцієнти факторів F_1, F_2 та F'_3 ;

F_1 – відношення кількості спільних людей до загальної кількості людей користувача T , що розраховується за формулою (5);

F_2 – відношення груп, у яких перебувають обидва користувачі, до кількості груп, у яких перебуває лише користувач T , що розраховується за формулою (6);

F'_3 – значення, що відображає подібність інтересів користувачів, що розраховується за формулою (3).

Формула (8) використовує порогове значення лише одного користувача. Така одностороння оцінка може бути недостатньо точною та приводити до поганих результатів рекомендації людей. Тому вирішено змінити формулу (8) так, щоб вона

враховувала порогові значення обох користувачів. Змінений варіант формули (8), який буде враховувати порогові значення обох користувачів матиме вигляд:

$$\frac{R(T, T_1) + R(T_1, T)}{2} > \text{Min}(\text{threshold}_T, \text{threshold}_{T_1}) \quad (11)$$

де $R(T, T_1)$ – оцінка зв'язку між користувачами T та T_1 ;

$R(T_1, T)$ – оцінка зв'язку між користувачами T_1 та T ;

$\text{Min}()$ – функція визначення мінімального з двох значень;

threshold_T – порогове значення для користувача T ;

threshold_{T_1} – порогове значення для користувача T_1 .

З метою врахування того факту, що окремі користувачі можуть нормально спілкуватися із людьми, які не дуже схожі на них, формулу (9) можна розширити внесенням нового фактору для зважування порогового значення, для врахування поправки на терпимість користувача до людей, які від нього відрізняються. Формула (9) із запропонованими змінами матиме вигляд:

$$\text{threshold} = \gamma \frac{\sum_{i \in Fr(T)} R(T, T_i)}{|Fr(T)|} + (1 - \gamma) \frac{\sum_{j \in \text{Min}R_n(T)} R(T, T_j)}{|\text{Min}R_n(T)|} \quad (12)$$

де threshold – порогове значення, що розраховується;

γ – ваговий коефіцієнт для регулювання впливу обох факторів на фінальне порогове значення;

$Fr(T)$ – множина людей користувача T ;

$\text{Min}R_n(T)$ – множина людей користувача T потужності n , для яких значення оцінки зв'язку мінімальне.

У пункті були описані недоліки базового методу та запропоновані способи їх усунення. Запропоновані рішення знайшли своє відображення у формулах (10), (11)

та (12) та направлені на врахування нових факторів та більш повного врахування старих факторів.

2.2.3 Метод рекомендації груп

Як методу розрахунку рекомендацій груп вирішено використати метод прогнозування оцінок шляхом використання профілів, що містять інтереси користувачів та груп. На нашу думку, цей метод дозволить формулювати достатньо точні рекомендації, використовуючи не використовуючи величезну кількість факторів із складністю обчислень, яку можна порівняти із складністю множення двох матриць.

Прогнозування оцінок користувачів із їх профілів та профілів груп пропонується розраховувати шляхом множення матриці з інтересами користувачів на матрицю з інтересами груп.

Нехай визначено матрицю з інтересами користувачів $U(m \times n)$, у якій m – кількість користувачів системи, а n – кількість факторів або інтересів, які можуть мати користувачі та групи. Нехай визначено матрицю з інтересами груп $G(k \times n)$, у якій k – кількість груп у системі, а n – така ж кількість факторів або інтересів, яким може відповідати група, як і у матриці інтересів користувачів. Тоді матрицю прогнозованих оцінок $R(m \times k)$ для m користувачів та k груп можна розрахувати використовуючи наступну формулою:

$$R(m \times k) = U(m \times n) \cdot G^T(n \times k) \quad (13)$$

де $R(m \times k)$ – матриця прогнозованих оцінок розмірами у m рядків, що відповідають користувачам, та k стовпців, що відповідають групам;

$U(m \times n)$ – матриця з вагами інтересів користувачів розмірами у m рядків, що відповідають користувачам, та n стовпців, що відповідають інтересами;

\cdot – операція добуту двох матриць;

$G^T(n \times k)$ – транспонована матриця з вагами інтересів груп розмірами n рядків, що відповідають інтересами, та k стовпців, що відповідають групам.

В отримані матриці $R(m \times k)$ в i -тому рядку цієї матриці будуть знаходитися прогнозовані оцінки груп i -того користувача, а у k -тому стовпчику цієї матриці будуть знаходитися оцінки всіх користувачів k -тій групі. Рекомендувати користувачам варто ті групи, для яких значення прогнозованої оцінки найбільше.

У випадку, коли необхідно розрахувати прогнозовані оцінки не для всіх користувачів системи, а для деякого конкретного користувача, то формулу (13) можна спростити до виду:

$$R_i(k) = U_i(n) \cdot G^T(n \times k) \quad (14)$$

де $R_i(k)$ – вектор прогнозованих оцінок i -того користувача для k груп;

$U_i(n)$ – матриця з вагами n інтересів i -того користувача;

\cdot – операція добуту двох матриць;

$G^T(n \times k)$ – транспонована матриця з вагами інтересів груп розмірами n рядків, що відповідають інтересами, та k стовпців, що відповідають групам.

В пункті 2.2.3 описано метод розрахунку рекомендацій груп шляхом використання профілів інтересів користувачів та груп. Формула (13) дозволяє розрахувати прогнозовані оцінки для групи користувачів та деякого набору груп. Формула (14) дозволяє розрахувати прогнозовані оцінки груп для конкретного користувача системи.

2.3 Висновки

У цьому розділі була описана запропонована нами технологія для системи рекомендацій людей та груп за інтересами у віртуальних спільнотах, у процесі чого

були описані функціональні моделі систем рекомендацій людей та груп, описано алгоритм рекомендації людей, який вирішено взяти за основу та доопрацювати, описано конкретні недоліки базового алгоритму, які, на нашу думку, у ньому присутні, та способи їх усунення, описано метод рекомендації груп на основі профілів користувачів та груп.

В підрозділі 2.1 детально описано окремо функціональну модель системи рекомендацій людей та функціональну модель системи рекомендацій груп. Описані функціональні моделі були подані на діаграмах функціональної декомпозиції на рисунках 2.1 та 2.2.

В пункті 2.2.1 подано метод розрахунку рекомендацій людей, який вирішено доопрацювати. Описаний метод базується на використанні формул (4), (5), (6), (7), (8) та (9).

В пункті 2.2.2 визначено слабкі місця базового методу та запропоновано способи їх усунення, описано необхідні фактори та покращено формули за якими проводяться розрахунки у базовому методі. Фактор використання однакових програм запропоновано замінити на фактор подібності інтересів користувачів, при цьому мають бути переглянуті вагові коефіцієнти факторів, у порівнянні з оригінальним методом. Для врахування двосторонньої можливості дружби і уникнення ситуацій, коли розрахунок рекомендацій із сторін різних користувачів даватиме різні результати, запропоновано при визначенні можливості дружби використовувати не вагу одного користувача, а одразу двох, щоб врахувати інтереси обох сторін. Для того, щоб врахувати можливість того, що окремі користувачі можуть бути друзями з людьми, які мають погляди, відмінні від поглядів користувача, для якого формується список рекомендацій, запропоновано розширити формулу розрахунку порогового значення шляхом її перетворення на зважену суму середнього значення оцінки зв'язку між користувачем і його друзями, та середнього значення оцінки дружби для деякої кількості людей користувача, оцінка зв'язку з якими найменша. Змінені формули базового методу мають номери (10), (11) та (12). Запропонована система вагових коефіцієнтів є популярним рішенням у рекомендаційних системах, оскільки через зміну значень окремих

вагових коефіцієнтів можна кардинально змінити поведінку системи. Самі значення вагових коефіцієнтів у ідеальній ситуації підбираються індивідуальні під поставлене завдання, щоб просувати саме ті об'єкти, які на думку розробника найкраще задовольняють потреби користувачів системи.

В пункті 2.2.3 описано метод розрахунку рекомендацій груп з використанням профілів інтересів користувачів та профілів інтересів груп для прогнозування можливих оцінок. Розрахунок рекомендацій для групи користувачів здійснюється за формулою (13). Розрахунок рекомендацій для одного користувача здійснюється за формулою (14). Вирішено скористатися таким методом, оскільки через врахування активності користувача можна розрахувати його актуальні інтереси і як наслідок використати їх для того, щоб спрогнозувати наскільки користувачеві може сподобатися або не сподобатися конкретна група або деякий набір груп.

Вміст цього розділу висвітлює всі методи та містить функціональні моделі розроблюваної системи, які утворюють запропоновану нами технологію розробки програмної системи рекомендацій людей та груп за інтересами у віртуальних спільнотах. Розроблені моделі та методи необхідні для подальшої розробки запропонованої нами технології, і вони будуть враховані та використані у процесі розробки алгоритмів, вимог та проектування програмного забезпечення для вирішення поставленої задачі.

3 АЛГОРИТМИ ТА ТЕХНОЛОГІЯ ДЛЯ РОЗРАХУНКУ РЕКОМЕНДАЦІЙ ЛЮДЕЙ ТА ГРУП

3.1 Алгоритми розрахунку рекомендацій людей та груп

3.1.1 Алгоритм розрахунку рекомендацій людей

Алгоритм формування рекомендацій людей загалом складається із трьох сценаріїв: розбиття користувачів на групи, формування рекомендацій всередині групи та формування рекомендацій з користувачів інших груп. Всі сценарії наведено на рисунку 3.1.

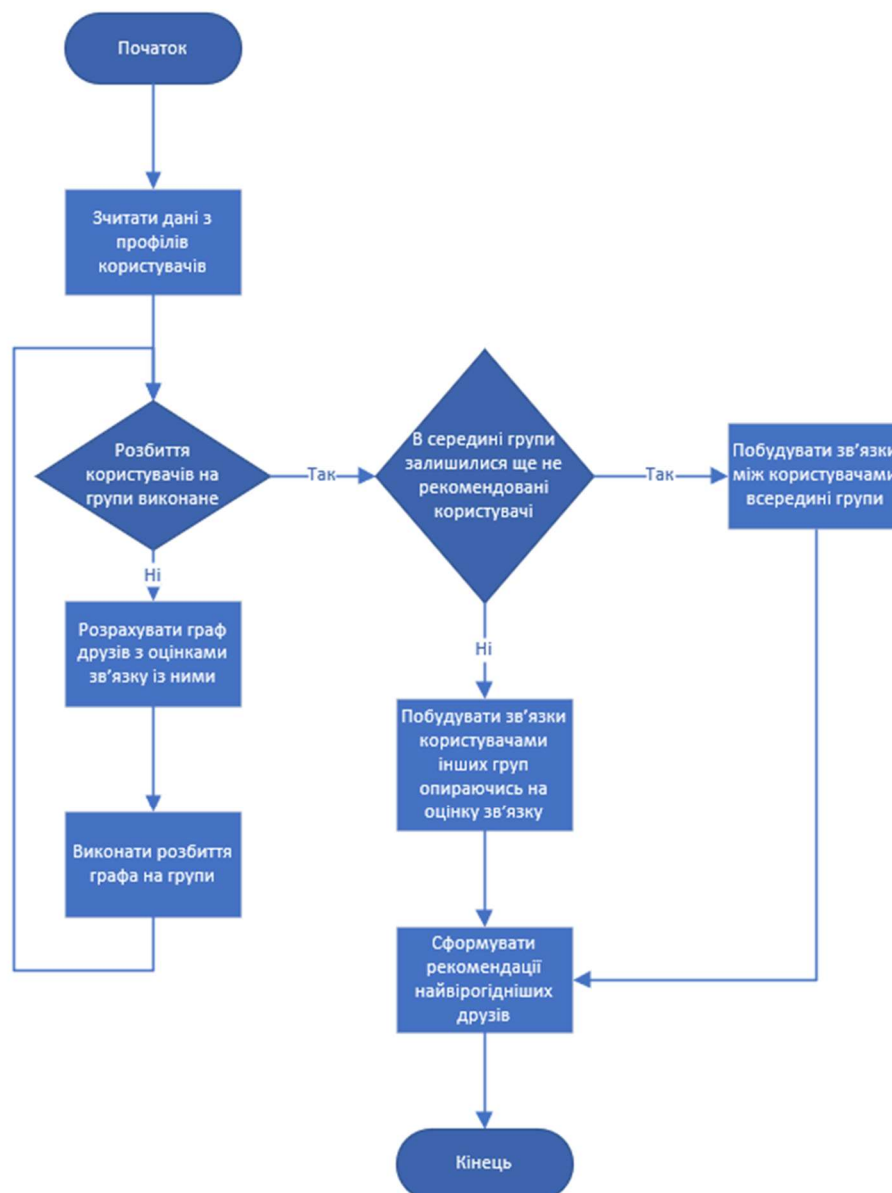


Рисунок 3.1 – Блок-схема алгоритму рекомендації людей

На кроці зчитування даних про користувачів системи з бази даних отримуються: інтереси користувачів, список груп, у яких перебувають користувачі та списки людей користувачів.

Після зчитування даних можливі два сценарії подальшої роботи алгоритму: якщо користувачі не розбиті на групи та якщо вони розбиті на групи.

Якщо користувачі не розбиті на групи, то запускається процес розрахунку цього розбиття, після чого система повертається на крок перевірки наявності розбиття. Сам процес розбивання користувачів на групи проходить у два етапи. На першому етапі будується зважений граф людей, ваги ребер якого розраховуються як оцінки сил зв'язків між користувачами. На наступному кроці алгоритму система виконує розбиття отриманого графа людей, використовуючи метод Лювена. У кінці цієї гілки алгоритму виконання переходить на етап перевірки наявності розбиття графа користувачів.

Якщо ж користувачі розбиті на групи, то відбувається перевірка, чи залишилися у групі користувача, для якого розраховуються рекомендації, ще не переглянуті користувачі. Якщо такого роду користувачі залишилися, то система розраховує силу зв'язку між ними, використовуючи формулу (10) та формує список рекомендованих людей із користувачів, для яких виконується нерівність (11) і з яким відповідають найбільші значення сили зв'язку. Інакше, якщо всі користувачі всередині групи вже були розглянуті, то система розраховує силу зв'язку за формулою (10) із користувачами інших груп і пропонує дружбу з тими, для кого виконується нерівність (11) і з ким у користувача найбільші значення сили зв'язку.

У пункті описано алгоритм роботи системи рекомендації людей, наведено блок-схему цього алгоритму та описано кожен його етап.

3.1.2 Алгоритм розрахунку рекомендацій груп

Алгоритм розрахунку рекомендацій груп, що базується на прогнозуванні оцінок користувачів з використанням їх профілів та профілів груп, може мати дві

варіації: прогнозування оцінок окремого користувача та прогнозування оцінок для множини користувачів. Сама ж система може використовуватися у двох режимах: прогнозування оцінок з подальшим їх збереженням та формування рекомендацій. Прогнозування оцінок без формування списку рекомендацій може бути корисним у випадку, коли у системи є час, у який вона найменше навантажена і може проводити розрахунки наперед. Блок-схему алгоритму розрахунку рекомендацій груп наведено на рисунку 3.2.

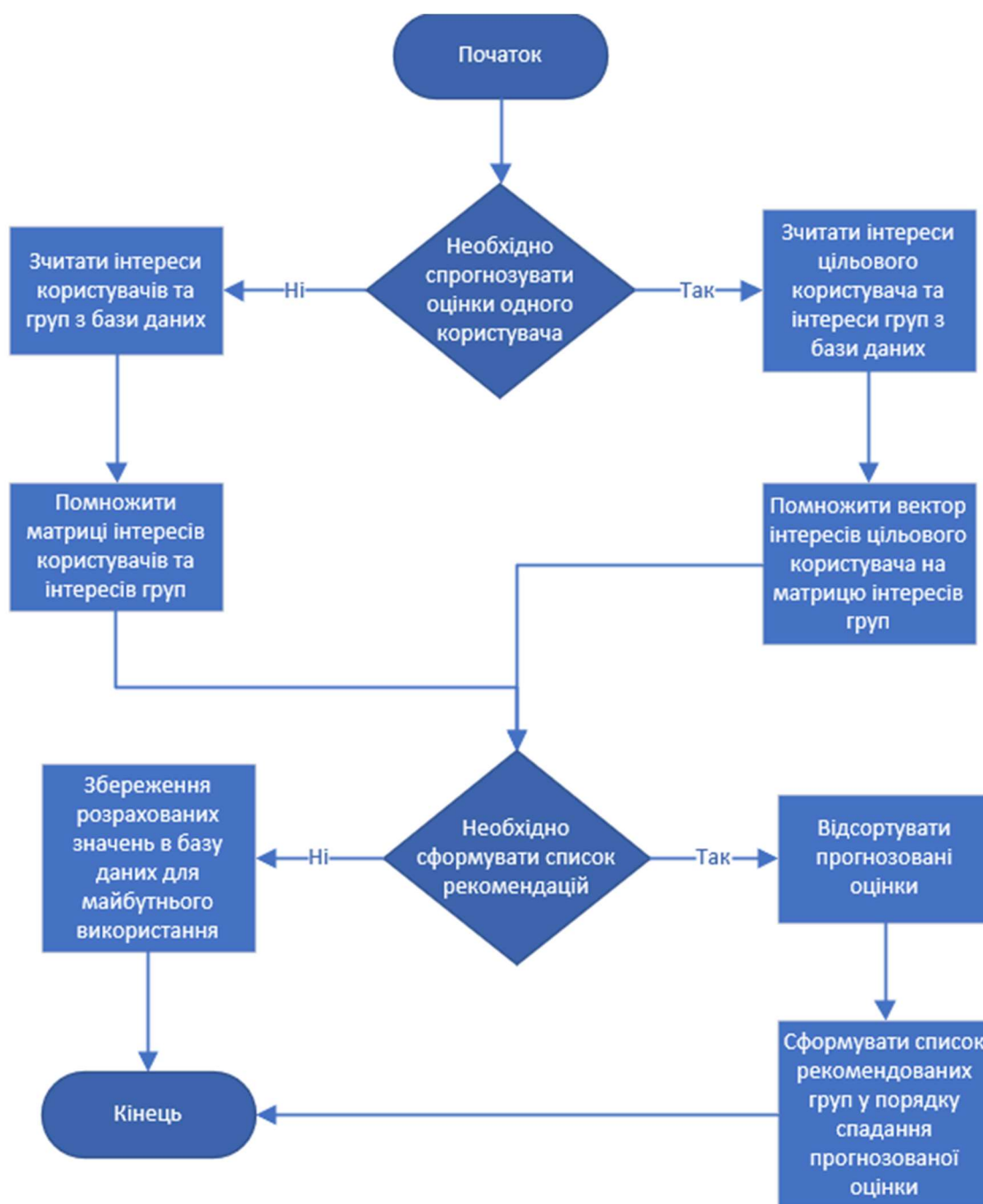


Рисунок 3.2 – Блок-схема алгоритму рекомендації груп

Цей алгоритм має декілька розгалужень на різні гілки. На першому розгалуженні система визначає для якої кількості необхідно спрогнозувати оцінки. Якщо потрібно спрогнозувати оцінки лише для одного користувача, то алгоритм переходить до гілки з наступних дій: зчитати інтереси користувача та профілі інтересів груп, виконати операцію множення над вектором інтересів користувача та матрицею з інтересами для кожної групи. Оскільки користувач та групи знаходяться у одному просторі інтересів, множення цих матриць дасть вектор з прогнозованими оцінками груп користувача, для якого вони розраховувалися. Після цього алгоритм переходить до іншого розгалуження, яке визначає потрібно розраховані оцінки зберегти або ж з них сформувати список рекомендованих груп.

Інша гілка алгоритму, яка слугує для розрахунку прогнозованих оцінок групи користувачів, містить наступні етапи: зчитування профілів з інтересами кожного користувача та профілів з інтересами кожної групи. Різниця між нею і гілкою, описаною у попередньому абзаці, полягає у тому, що розрахунки прогнозованих оцінок відбуваються множенням не вектору інтересів користувача, а цілої матриці, що містить інтереси всіх користувачів, на матрицю інтересів груп. У результаті таких розрахунків буде отримано матрицю, яка буде містити прогнозовані оцінки кожним користувачем кожної групи. Далі, як і у випадку протилежної гілки, алгоритм переходить до іншого розгалуження, яке визначає потрібно розраховані оцінки зберегти або ж з них сформувати список рекомендованих груп.

Якщо необхідно просто зберегти розраховані оцінки, то над ними не проводиться більше жодних операцій і вони зберігаються у базі даних. Якщо ж потрібно сформувати рекомендації, то значення прогнозованих оцінок для кожного користувача сортуються у порядку спадання і система відбирає тільки ті групи, для яких значення прогнозованої оцінки найбільше. Ці відібрані значення і є списками рекомендацій для кожного користувача системи.

У пункті описано алгоритм розрахунку рекомендацій груп, шляхом прогнозування оцінок користувачів з використанням профілів інтересів користувачів та профілів інтересів груп, наведено блок-схему алгоритму рекомендації груп та описано її етапи.

3.2 Розробка вимог до програмного забезпечення та даних, що використовуються для вирішення задачі

Вимоги до програмного забезпечення це набір факторів, які має задовольняти програмний продукт, що розробляється. Розробка вимог до програмного забезпечення є невід'ємним етапом процесу розробки програмного забезпечення. У розділі буде сформовано функціональні вимоги до систем рекомендації людей та груп, а також вимоги до даних, якими ці системи будуть оперувати у процесі розрахунків.

Для системи рекомендацій людей висунуто наступні функціональні вимоги:

- система повинна враховувати інтереси користувача, групи, у яких він перебуває та кількість спільних людей між користувачами;
- система повинна враховувати історію відмов користувачів;
- всі фактори, які враховує система, мають бути зважені відповідно до їх впливу на якість і точність рекомендацій;
- система має враховувати можливість дружби користувачів з інтересами, списком груп та людей, що відрізняються;
- система має на вимогу формувати списки рекомендованих людей необхідної довжини.

Всі ці функціональні вимоги визначають поведінку системи рекомендацій людей, що розробляється.

Для системи рекомендацій груп висунуто наступні функціональні вимоги:

- система повинна враховувати інтереси користувачів та груп;
- рекомендації повинні формуватися шляхом розрахунку прогнозованих оцінок з врахуванням інтересів;
- система повинна мати можливість розрахувати прогнозовані оцінки як для окремого користувача, так і для групи користувачів.

Для кращого розуміння факторів, які враховують описані вище рекомендаційні системи, доцільно буде розробити вимоги до даних. Для системи рекомендацій людей висунуто наступні вимоги до даних:

- для розрахунку фактору, що відображає подібність за списком людей, у системі мають бути доступними списки друзів кожного користувача;
- для розрахунку фактору, що відображає подібність за списком груп, у системі мають бути доступними списки груп, у яких перебувають користувачі;
- для розрахунку фактору, що відображає подібність за списком інтересів, у системі мають бути доступні дані з профілів інтересів користувачів або дані про активність користувачів, з яких можна розрахувати інтереси.

Таким чином, у системі розрахунку рекомендацій людей мають бути присутні такого роду дані, як списки людей, списки груп із списками користувачів, а також списки інтересів кожного користувача.

Вимоги до даних для системи рекомендацій груп виражаються так:

- для розрахунку прогнозованих оцінок груп користувачами у системі мають бути доступні профілі з інтересами користувачів або дані про активність користувачів, з яких можна розрахувати їх інтереси;
- для розрахунку прогнозованих оцінок груп користувачами у системі мають бути доступні профілі груп з інтересами, яким вони відповідають, або дані про вміст груп, з яких можна вирахувати яким інтересами вони відповідають.

У підрозділі сформульовано функціональні вимоги до систем розрахунку рекомендацій людей та груп, а також вимоги до даних, які використовують ці системи. У набір необхідних даних входять списки друзів користувачів, списки груп із списками користувачів, які у них входять, профілі інтересів користувачів або дані про активність користувачів у системі, які дозволять розрахувати інтереси, а також профілі груп з інтересами, яким вони відповідають, або інформацію про вміст груп, з якої можна розрахувати їх інтереси. За необхідності із введеного та розрахованого профілів користувачів та груп можна шляхом зваженої суми розрахувати єдиний профіль інтересів, проте, через необхідність врахування специфіки кожної конкретної системи при здійсненні такої операції, під час роботи профілі інтересів користувачів та груп будуть обмежені розрахованими із активності користувачів у системі.

3.3 Проектування програмного забезпечення для вирішення задачі

У розділі буде спроектовано загальну архітектуру програмного рішення, що розробляється, а також архітектуру його окремих компонентів. У системі планується реалізація трьох компонентів: модуля обробки даних, модуля розрахунку рекомендацій людей та модуля розрахунку рекомендацій груп. Компоненти системи та зв'язки між ними подано на діаграмі компонентів (рисунок 3.3).

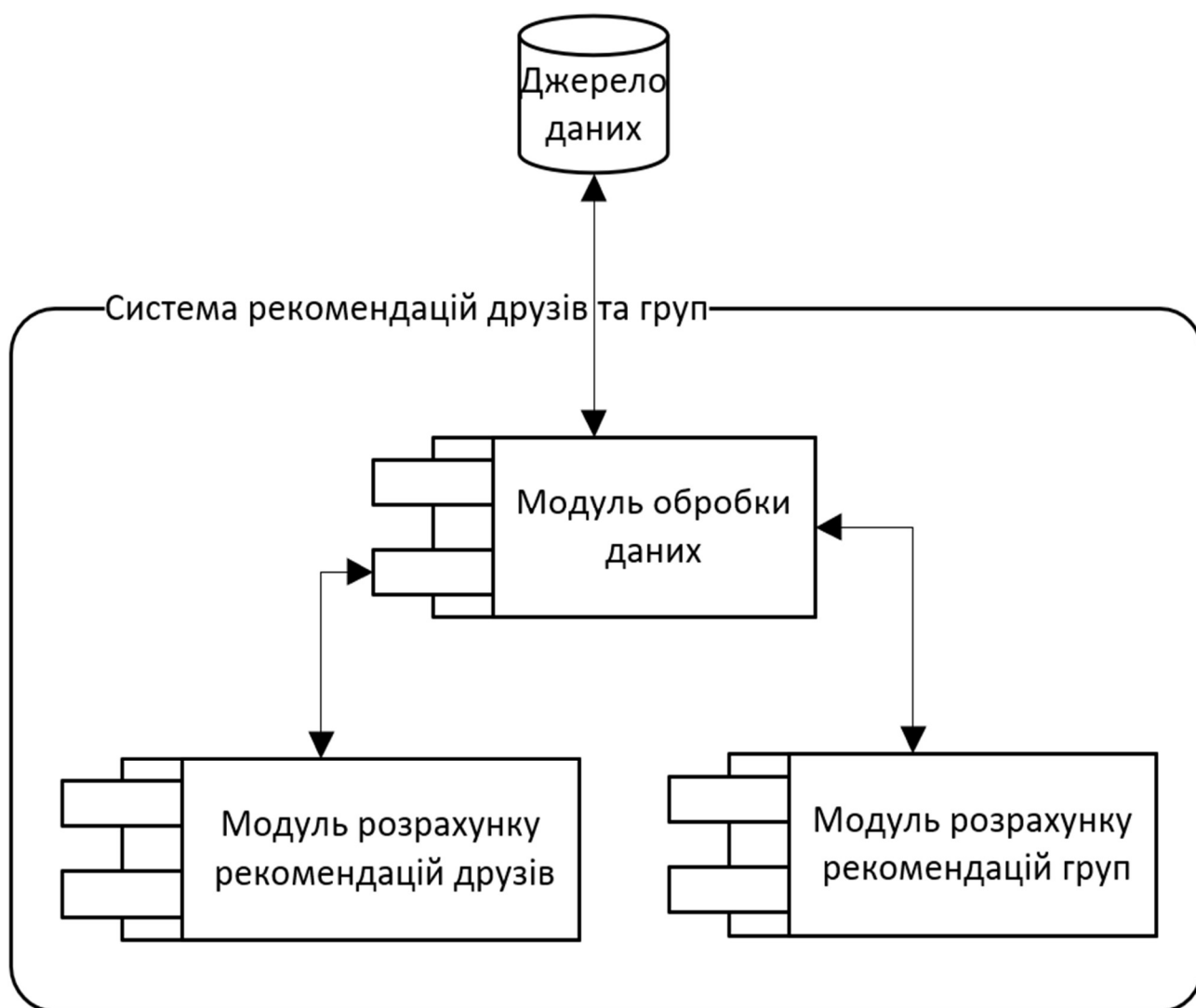


Рисунок 3.3 – Модель компонентів проектованої системи

Як показано на рисунку 3.3 передбачається, що система розрахунку рекомендацій людей та груп не буде великою. Джерело даних є зовнішньою

сутністю, яка знаходиться за межами системи. Зв'язок з цим джерелом даних здійснюється через модуль обробки даних. Оскільки нами система розробляється для її тестування на готовому наборі даних, модуль обробки даних буде виконувати наступні задачі:

- зчитування та запис даних у зовнішньому сховищі;
- приведення даних із зовнішнього сховища у формат, придатний для обробки модулями розрахунку рекомендацій;
- приведення даних, отриманих від систем розрахунку рекомендацій у вигляд, придатний для зберігання.

Модуль розрахунку рекомендацій людей буде виконувати наступні задачі:

- формування графу людей;
- розбиття графу людей;
- розрахунок можливості дружби між користувачами;
- формування списку рекомендованих користувачів.

Модуль розрахунку рекомендацій груп буде виконувати наступні задачі:

- множення матриць з вагами інтересів користувачів та груп;
- виділення із прогнозованих оцінок списку рекомендованих груп;
- розрахунок прогнозованих оцінок з метою їх збереження на майбутнє.

Для кращого розуміння того, як дані рухаються всередині системи, опишемо це використовуючи діаграми потоків даних. Загальна діаграма потоків даних наведена на рисунку 3.4.

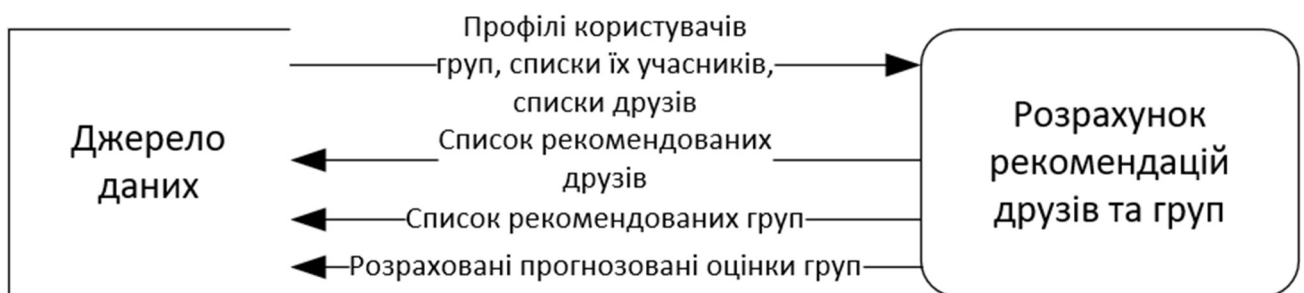


Рисунок 3.4 – Діаграма потоків даних між джерелом даних і системою

На рисунку 3.4 показано які дані система отримує на вхід від джерела даних, та які дані вона повертає для їх збереження. Всього на цій діаграмі визначено чотири потоки даних:

- профілі користувачів, списки людей користувачів, списки груп та їх учасників, профілі користувачів та груп;
- список рекомендованих людей;
- список рекомендованих груп;
- прогнозовані оцінки користувачів.

Діаграма потоків даних, наведена на рисунку 3.4, візуалізує лише потоки даних ззовні системи, не заглиблюючись у потоки даних між модулями. Діаграма потоків даних всередині системи між модулями обробки даних та розрахунку рекомендацій людей наведена на рисунку 3.5.

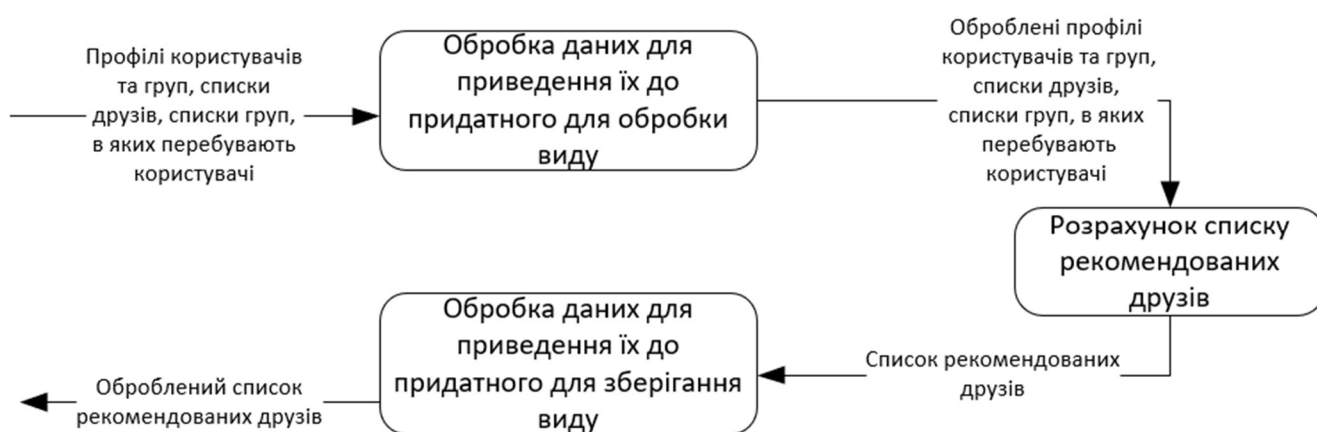


Рисунок 3.5 – Діаграма потоків даних між модулями обробки даних та формування списку рекомендованих людей

Всього на рисунку 3.5 визначено чотири потоки даних:

- профілі користувачів та груп, списки людей користувачів, списки груп, у яких перебувають користувачі;
- оброблені профілі користувачів та груп, списки людей користувачів, списки груп, у яких перебувають користувачі;
- список рекомендованих людей;
- оброблений список рекомендованих людей.

Ці потоки даних передаються між трьома процесами:

- обробка даних для приведення їх до придатного для обробки виду;
- розрахунок списку рекомендованих людей;
- обробка даних для приведення їх до придатного для зберігання виду.

Перший та останній процеси виконуються модулем обробки даних, а другий процес виконується модулем розрахунку рекомендацій людей.

Діаграма потоків даних у середині системи між модулями обробки даних та розрахунку рекомендацій груп наведено на рисунку 3.5.

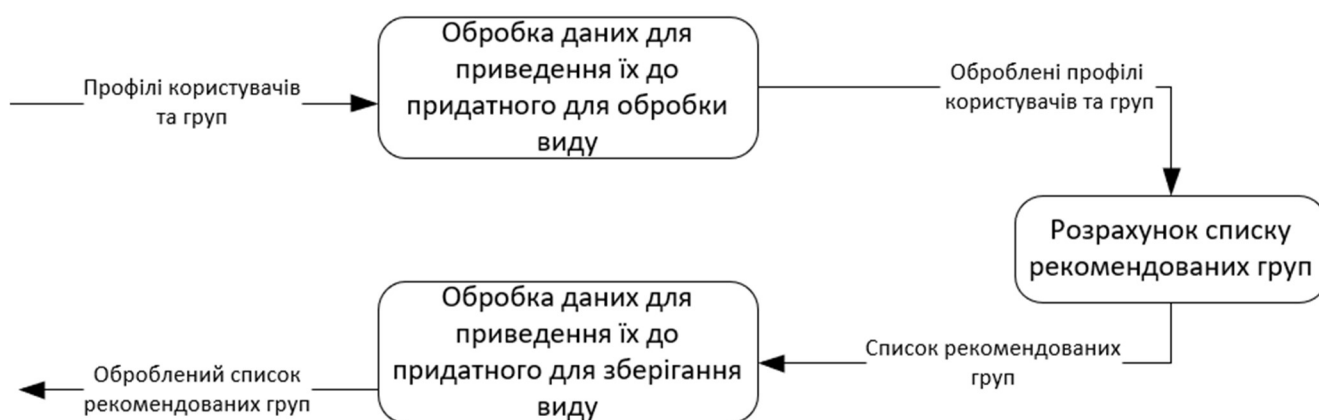


Рисунок 3.6 – Діаграма потоків даних між модулями обробки даних та формування списку рекомендованих груп

Всього на рисунку 3.6 визначено чотири потоки даних:

- профілі користувачів та груп;
- оброблені профілі користувачів та груп;
- список рекомендованих груп;
- оброблений список рекомендованих груп.

Ці потоки даних передаються між трьома процесами:

- обробка даних для приведення їх до придатного для обробки виду;
- розрахунок списку рекомендованих груп;
- обробка даних для приведення їх до придатного для зберігання виду.

Перший та останній процеси виконуються модулем обробки даних, а другий процес виконується модулем розрахунку рекомендацій груп.

Для виконання своїх функцій модуль обробки даних має містити такий набір функцій та методів:

- readUserFriends – метод для зчитування із файлу графа людей;
- readUserGroup – метод для зчитування списків груп, у яких перебуває кожен користувач;
- readItemInterMatr – метод для зчитування матриці з інтересами, яким відповідає кожна група;
- readUserInterMatr – метод для зчитування матриці інтересів користувачів;
- readUsers – метод для зчитування списку користувачів системи;
- read_sets – метод для зчитування множин людей користувачів та множин груп, у яких вони перебувають;
- writeToFile – метод для зберігання даних, які розрахувала система;
- saveSets – метод для збереження множин людей та множин груп, у яких перебувають користувачі;
- getFriends – метод для зчитування множини людей для конкретного користувача системи;
- calcFriendsRecList – метод для формування списку рекомендацій людей для конкретного користувача.

Модуль розрахунку рекомендацій людей має містити реалізацію наступного низки методів:

- threshold – метод для розрахунку порогового значення сили зв'язку;
- checkFriendPosib – метод для перевірки можливості дружби;
- linkStrength – метод для розрахунку сили зв'язку між користувачами;
- getMinK – метод для виділення K людей, сила зв'язку з якими найменша;
- getGraphPartition – метод для розбивання графа людей на групи.

Модуль розрахунку рекомендацій груп має реалізовувати наступні методи:

- calcSingleUser – метод для розрахунку прогнозованих оцінок для окремого користувача;

- calcForAll – метод для розрахунку прогнозованих оцінок для всіх користувачів системи;
- calcRecomSingle – метод для розрахунку списку рекомендованих груп для окремого користувача;
- calcRecomForAll – метод для розрахунку списку рекомендованих груп для всіх користувачів;

UML-діаграма класів для описаних модулів наведена на рисунку 3.7.

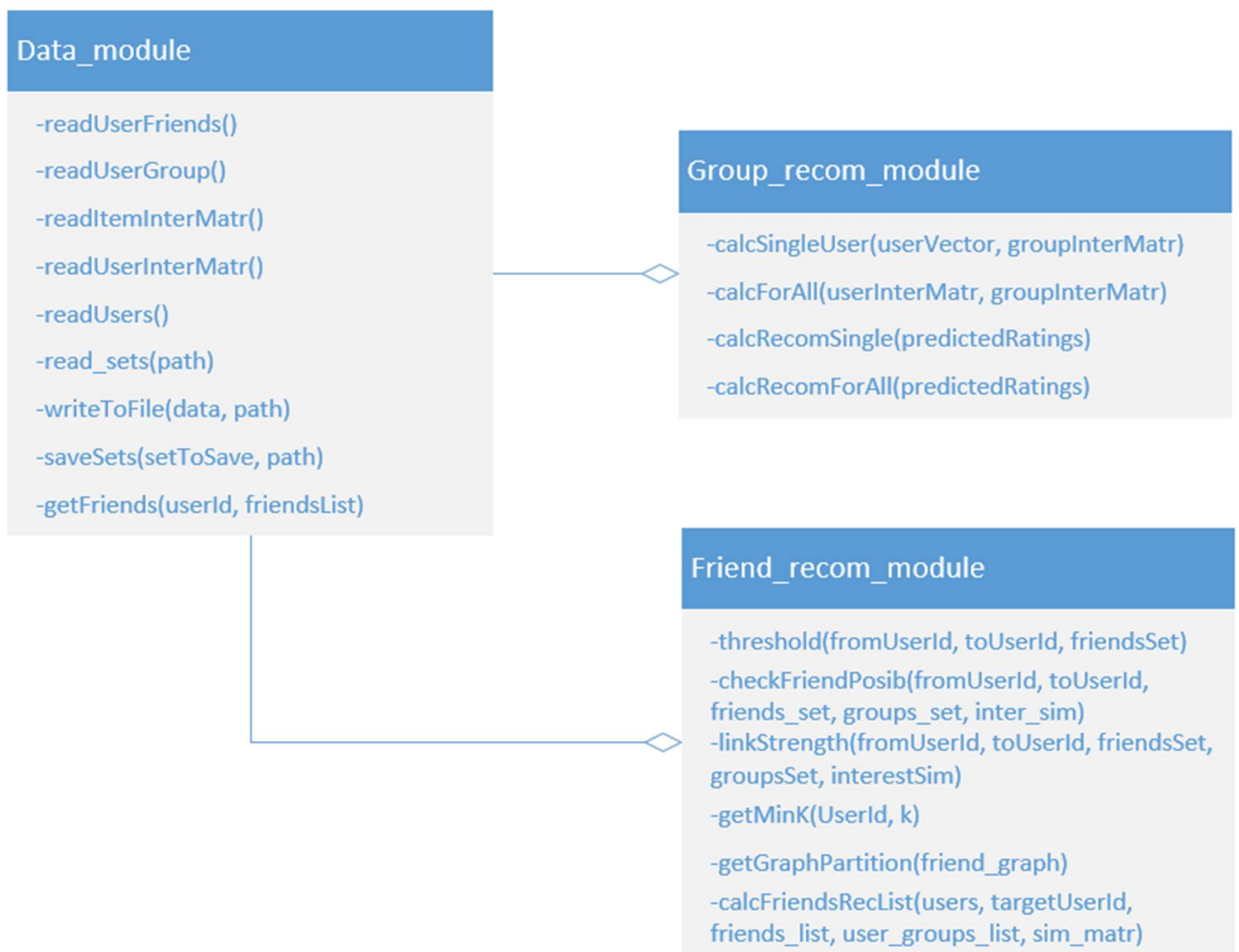


Рисунок 3.7 – UML-діаграма класів проектованої системи

Спроектована у цьому підрозділі система може бути реалізована та інтегрована практично у будь-який існуючий проект. Оскільки зв'язок між модулем розрахунку рекомендацій груп та модулем розрахунку рекомендацій

друзів відсутній, за необхідності використання функціоналу лише одного із них, інший модуль можна упустити. У той же час модуль для роботи із даними повинен бути присутнім у системі у будь-якому випадку. Він може бути як окремим елементом системи, так і частиною іншого модуля.

Модуль розрахунку рекомендацій груп може виконувати наступні дії:

- розрахунок прогнозованих оцінок груп для множини користувачів;
- розрахунок прогнозованих оцінок груп для конкретного користувача;
- розрахунок списку рекомендованих груп із прогнозованих оцінок для одного користувача;
- розрахунок списку рекомендованих груп із прогнозованих оцінок для множини користувачів.

Для розрахунку прогнозованих оцінок груп для множини користувачів у метод `calcForAll` необхідно передати два параметра:

- `userInterMatr` – матриця ваг інтересів для множини користувачів, у якій кожен рядок відповідає окремому користувачеві, а кожен стовпчик відповідає конкретному інтересу;
- `groupInterMatr` – матриця ваг інтересів для множини груп, яка має формат, аналогічний до матриці ваг інтересів користувачів.

У результаті цей метод поверне матрицю прогнозованих оцінок користувачів, у якій кожен рядок міститиме прогнозовані оцінки для конкретного користувача, а кожен стовпчик оцінки для конкретної групи.

Для розрахунку прогнозованих оцінок груп для конкретного користувача у метод `calcSingleUser` необхідно передати два параметра:

- `userVector` – вектор ваг інтересів конкретного користувача;
- `groupInterMatr` – матриця ваг інтересів груп, у якій кожен рядок відповідає окремій групі, а кожен стовпчик відповідає конкретному інтересу.

Як результат розрахунків, цей метод поверне вектор прогнозованих оцінок, у якому кожне значення міститиме оцінку для конкретної групи.

Для розрахунку списку рекомендованих груп із прогнозованих оцінок для одного користувача у метод `calcRecomSingle` необхідно передати один параметр

`predictedRatings`, який є вектором, у якому кожне значення відповідає прогнозованій оцінці для конкретної групи.

Для розрахунку списку рекомендованих груп із прогнозованих оцінок для множини користувачів у метод `calcRecomForAll` необхідно передати один параметр `predictedRatings`, який є матрицею, рядки якої містять прогнозовані оцінки груп для конкретного користувача, а кожен стовпчик містить прогнозовані оцінки для конкретної групи.

Модуль розрахунку рекомендацій людей може виконувати наступні дії:

- розрахунок порогового значення сили зв'язку між двома користувачами;
- розрахунок можливості дружби між двома користувачами;
- розрахунок сили зв'язку між двома користувачами;
- знайти підмножину k друзів з найменшим показником сили зв'язку;
- розрахунок розбиття графа методом Лювена;
- розрахунок списку рекомендованих людей для конкретного користувача.

Для розрахунку порогового значення сили зв'язку між двома користувачами у метод `threshold` необхідно передати наступні параметри:

- `fromUserId` – ідентифікатор першого користувача;
- `toUserId` ідентифікатор другого користувача;
- `friendsSet` – множини друзів користувачів виду (ідентифікатор користувача, множина друзів користувача).

Як результат цей метод поверне числове значення у діапазоні $[0;1]$, яке необхідне для розробленого методу для визначення того, формувати рекомендацію конкретної людини чи ні.

Для розрахунку можливості дружби між двома користувачами у метод `checkFriendPosib` необхідно передати наступні параметри:

- `fromUserId` – ідентифікатор першого користувача;
- `toUserId` ідентифікатор другого користувача;
- `friendsSet` – множини друзів користувачів виду (ідентифікатор користувача, множина друзів користувача);

- `groups_set` – множини груп, у яких перебувають користувачі, виду (ідентифікатор користувача, множина груп користувача);
- `inter_sim` – значення подібності інтересів двох користувачів.

Цей метод поверне значення типу `bool`, при чому значення `True` означатиме, що двом користувачам можна запропонувати один одного, а значення `False` означатиме, що користувачам не потрібно пропонувати один одного.

Для розрахунку сили зв'язку між двома користувачами у метод `linkStrength` необхідно передати такого ж роду параметри, як і для методу `checkFriendPosib`. У результаті своєї роботи цей метод поверне значення у діапазоні $[0;1]$, при чому чим воно більше, тим сильніший зв'язок між користувачами.

Для знаходження підмножини із k друзів з найменшим показником сили зв'язку у метод `getMinK` необхідно передати ідентичні параметри:

- `UserId` – ідентифікатор користувача;
- k – число друзів з найменшою силою зв'язку.

Цей метод поверне вектор, у якому кожне значення відповідатиме силі зв'язку із деяким користувачем. Кількість значень регулюється параметром k .

Для знаходження розбиття графа методом Лювена у метод `getGraphPartition` необхідно передати параметр `friend_graph`, який є об'єктом звичайного графа бібліотеки `network`. У результаті виконання цього методу на місце його виклику повертається словник, який містить значення вузлів та номер кластеру, до якого вони відносяться.

Для розрахунку списку рекомендованих людей для конкретного користувача у метод `calcFriendsRecList` необхідно передати наступні параметри:

- `users` – множина ідентифікаторів користувачів системи;
- `targetUserId` – ідентифікатор користувача, для якого розраховується список рекомендацій;
- `friends_list` – множина друзів виду (ідентифікатор користувача, множина ідентифікаторів друзів);
- `user_groups_list` – множина груп, у яких перебувають користувачі, виду (ідентифікатор користувача, множина ідентифікаторів груп, у яких він перебуває);

– `sim_matr` – матриця подібності інтересів користувачів, у якій всі значення нижче головної діагоналі рівні нулю.

Цей метод поверне на місце виклику відсортований список рекомендованих людей із елементами виду (ідентифікатор користувача, сила зв'язку), на початку якого будуть люди з найбільшим значенням сили зв'язку.

Таким чином, використання описаних модулів зводиться до виклику їх окремих методів та передачі у них необхідної інформації. Це означає, що їх інтеграція до існуючої системи зводиться до написання модуля обробки даних, який слугуватиме мостом між цими методами та джерелом даних системи.

У підрозділі описано структуру розроблюваної системи та подано її на рисунку 3.3, описано модулі, з яких має складатися система, діаграми потоків даних між системою та зовнішнім джерелом даних на рисунку 3.4 та між модулями системи на рисунках 3.5, 3.6. Діаграма класів для проектованої системи наведена на рисунку 3.7. Також для цієї діаграми класів, виконано опис методів для модулів розрахунку рекомендацій груп та розрахунку рекомендацій людей, а також описано можливості їх інтеграції до існуючої системи.

3.4 Висновки

У розділі виконано наступні дії:

- описано алгоритм розрахунку рекомендацій людей;
- описано алгоритм розрахунку рекомендацій груп;
- розроблено функціональні вимоги до програмного забезпечення та до даних, які використовує система;
- спроектовано структуру модулів та потоки даних у системі.

У підрозділі 3.1 розроблено блок-схеми алгоритмів розрахунку рекомендацій людей та груп та наведено їх на рисунках 3.1 та 3.2. Для кожної блок-схеми описано всі зображені на ній гілки алгоритмів та всі присутні на блок-схемах умови та розгалуження.

У підрозділі 3.2 розроблено функціональні вимоги до систем розрахунку рекомендацій людей та груп, а також вимоги до даних, які має використовувати система. Зокрема визначено, що для функціонування системи потрібно у джерелі даних мати наступну інформацію:

- список користувачів системи;
- список груп, які визначено у системі;
- списки людей кожного з користувачів;
- профілі інтересів користувачів або інформація про активність користувачів у системі, з якої можна розрахувати їх інтереси;
- профілі інтересів груп або інформація про контент групи, яка дозволить розрахувати інтереси груп;
- списки користувачів, які перебувають у кожній групі

У підрозділі 3.3 спроектовано структуру модулів системи та наведено її на рисунку 3.3. Для кращого розуміння того, як рухаються дані між системою та джерелом даних та між модулями системи були створені діаграми потоків даних, наведені на рисунках 3.4, 3.5 та 3.6. На рисунку 3.4 наведено діаграму потоків даних між зовнішнім джерелом даних та системою, на рисунку 3.5 наведено діаграму потоків даних між модулем обробки даних та модулем розрахунку рекомендацій людей, а на рисунку 3.6 наведено діаграму потоків даних між модулем обробки даних та модулем розрахунку рекомендацій груп. Були описані всі потоки даних, які присутні на рисунках 3.4, 3.5 та 3.6. Діаграма класів для модулів системи наведена на рисунку 3.7. До цієї діаграми класів наведено детальний опис методів для кожного окремого модуля із деталізацією вхідних та вихідних параметрів. Були описані можливості інтеграції запропонованих модулів до будь-якої існуючої системи.

Блок-схеми з підрозділу 3.1 призначені для кращого розуміння того, як у загальному працюють алгоритми розрахунку рекомендацій людей та груп. Доданий до них опис чітко визначає послідовність кроків алгоритму та дає пояснення кожному процесу, який у ньому представлений.

Функціональні вимоги до систем рекомендації людей та груп призначені для чіткого визначення задач, які вони мають вирішувати, а вимоги до даних

призначені для чіткого визначення факторів, які необхідні для здійснення розрахунків за алгоритмами, описаними у підрозділі 3.1.

Діаграма компонентів з підрозділу 3.3 призначена для кращого розуміння структури системи та визначення компонентів, з яких вона складається. Діаграми потоків даних слугують для чіткого визначення того, як дані поступають до системи та як вони рухаються всередині системи. Діаграма класів необхідна для точного моделювання модулів системи, а її детальний опис слугує для детального розкриття кожного окремого методу для кожного модуля.

У розділі 3 розроблено та описано алгоритми для вирішення задачі дипломної роботи, розроблено функціональні вимоги до програмного забезпечення для вирішення задачі та вимоги до даних, які необхідні для функціонування системи, а також спроектовано структуру модулів системи та потоки даних всередині та ззовні системи. За матеріалами розділу 3 можна реалізувати розроблену систему розрахунку рекомендацій людей та груп та інтегрувати реалізоване рішення у готовий програмний продукт. Завдяки тому, що розроблені моделі не прив'язані до конкретної мови програмування, вибір інструментів для реалізації системи не обмежений практично жодними програмними засобами чи платформами.

4 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ РОЗРАХУНКУ РЕКОМЕНДАЦІЙ ЛЮДЕЙ ТА ГРУП

4.1 Програмна реалізація

Важливим кроком розробки програмного рішення є вибір мови програмування, на якій воно буде розроблятися. У процесі вибору мови програмування для програмної реалізації розроблених моделей та методів проведено аналіз популярних мов програмування, а саме: C++, C#, Java, Python.

C++ – мова програмування високого рівня, розроблена у 1985 році. Незважаючи на вік, вона є дуже популярною через свою потужність та високу швидкість. C++ чудово підходить як для об'єктно-орієнтованого, так і для функціонального програмування. Основною сферою її використання є розробка програм, у яких необхідна низькорівнева робота з обладнанням та висока продуктивність, проте, потребує часу та зусиль для освоєння;

C# – мова програмування високого рівня, розроблена у 2000 році. Основним напрямком використання цієї мови є розробка об'єктно-орієнтованих кросс-платформних додатків. Через здатність роботи із задачами асинхронного програмування вона добре підходить для розробки API, роботи з графікою та розробки бізнес-додатків;

Java – мова програмування високого рівня, розроблена у 1990 році. Вона розроблялася для об'єктно-орієнтованого програмування та створення кросс-платформних застосунків, що виражається у тому, що розроблений цією мовою програмування застосунок можна запустити на практично будь-яких пристроях, на яких встановлена віртуальна машина Java. Основними напрямками застосування цієї мови програмування є корпоративні системи, кросс-платформні застосунки, веб-сервіси та ігри.

Python – мова програмування, що орієнтована на функціональну парадигму, але у якій реалізовано функції об'єктно-орієнтованого програмування. Ця мова програмування широко використовується у системах, що обробляють дані, у науці, при розробці штучного інтелекту та у багатьох інших сферах. Цією мовою

програмування реалізовано велику кількість математичних бібліотек, бібліотек для обробки даних та машинного навчання.

У результаті проведеного аналізу мов програмування вирішено, що для реалізації системи рекомендацій найкраще підходить мова програмування Python, оскільки вона чудово підходить для роботи з даними та має велику кількість корисних бібліотек у відкритому доступі. Серед бібліотек, які можуть знадобитися у розробці рекомендаційної системи, можна виділити: `numpy`, `math`, `community`, `network`.

`Numpy` – бібліотека для числових обчислень із реалізованими у ній алгоритмами різного роду математичних обчислень, які реалізовані у функціях та операторах. Реалізований у ній набір методів та алгоритмів дозволяє створювати складні структури даних, які можна ефективно використовувати у різних алгоритмах аналізу даних;

`Math` – модуль, що містить широкий функціонал для роботи з числами і реалізовує різні математичні функції;

`Community` – модуль, що містить реалізовані функції для виявлення груп у різного роду графах;

`Networkx` – модуль для створення, обробки та вивчення структури, динаміки та функцій складних мереж.

Як середовище розробки, вирішено обрати Microsoft Visual Studio, оскільки воно містить обширний функціонал для розробки додатків різними мовами, містить засоби для відлагоджування програм, зручний текстовий редактор для коду та дозволяє зручно керувати проектами.

Для запропонованої системи необхідно реалізувати три модулі: модуль обробки даних, модуль розрахунку рекомендацій груп та модуль розрахунку рекомендацій людей. Діаграма класів із вказанням методів для цих трьох модулів наведена на рисунку 3.7.

Основними методами модуля рекомендації людей є:

- розрахунок сили зв'язку між двома користувачами;
- розрахунок порогового значення для користувача;
- визначення можливості дружби між користувачами;

- розбиття користувачів на групи;
- розрахунок списку рекомендованих людей для конкретного користувача.

Основними методами модуля рекомендацій груп є:

- розрахунок рекомендацій груп для одного користувача;
- розрахунок рекомендацій груп для множини користувачів;
- розрахунок прогнозованих оцінок для конкретного користувача;
- розрахунок прогнозованих оцінок для множини користувачів.

Всі основні методи розробленої системи, реалізовані на мові Python, наведено у додатку А.

4.2 Обробка та аналіз тестових наборів даних

Як тестовий набір даних вирішено обрати опублікований у відкритому доступі набір даних з сайту Lastfm, оскільки він містить необхідну для тестування інформацію про соціальні зв'язки та переглянуті користувачами об'єкти системи. Також вагомим фактором є те, що цей набір даних за умовами ліцензії можна використовувати для некомерційної діяльності. Він має наступні характеристики:

- містить 1892 користувачі;
- 17632 об'єкта;
- 12717 двосторонніх зв'язків, що означають дружбу;
- 92834 взаємодій користувачів з об'єктами;
- 11946 тегів;
- 186479 записів про позначення деяким тегом деякого об'єкта.

Набір даних Lastfm відображає знімок мережі, присвяченій музиці, тобто об'єктами всередині системи є виконавці, а тегами – жанри. Коли користувач позначає об'єкт тегом, він позначає виконавця жанром.

Цей набір даних складається з наступних файлів:

- artists.dat – містить інформацію про виконавців;
- tags.dat – містить інформацію про теги;

- user_artists.dat – містить інформацію про виконавців, яких слухав користувач, а також кількість прослуховувань для цього виконавця;
- user_taggedartists.dat – містить інформацію про теги, якими користувач позначив виконавця;
- user_friends.dat – містить пари формату (користувач, користувач), які позначають дружбу між ними.

У нашому експерименті виконавці будуть виступати у ролі груп, а теги у ролі інтересів. Це можливо тому, що для рекомендаційної системи група є точно таким же об'єктом, як і виконавець. Завдяки цьому можна стверджувати, що сформований список рекомендацій виконавців моделює список рекомендацій груп. Оскільки у цьому наборі даних не міститься готової інформації про належність кожного виконавця деякому набору жанрів, але міститься інформація про позначення виконавців користувачами та інформацію про кількість прослуховувань конкретного виконавця кожним користувачем, можна розрахувати профілі інтересів користувачів та об'єктів. Розрахунок кожного конкретного інтересу групи всередині її профілю здійснювався як відношення кількості позначень користувачами деякої властивості до кількості користувачів, які позначили об'єкт хоч одною властивістю. Таким чином, якщо одна властивість була відмічена всіма користувачами, то вона отримає вагу, рівну 1.

У першу чергу були розраховані профілі об'єктів, які у нашій системі відображають групи. Розрахунок профілів об'єктів відбувався наступним чином:

- для кожного об'єкта системи розраховуються кількість згадувань конкретного тегу та кількість користувачів, які позначили об'єкт хоч одним тегом.
- вага кожного тегу, які у нашій системі визначають інтереси, розраховується як відношення згадувань конкретного тегу до загальної кількості користувачів, які позначили об'єкт хоча б одним тегом.

У результаті кожен об'єкт системи представляється вектором, у якому значення кожної координати відповідає значенню, з яким об'єкт відноситься до деякого інтересу.

Профілі інтересів користувачів були розраховані на основі розрахованих на попередньому кроці профілів інтересів об'єктів системи та кількості взаємодій кожного користувача з об'єктами системи. Як і у випадку профілів інтересів об'єктів, профілі інтересів користувачів розраховувалися у два етапи:

- для кожного об'єкта, з яким взаємодіяв користувач, розраховується його вага, як відношення кількості взаємодій з цим об'єктом до кількості взаємодій цього користувача з усіма об'єктами системи;

- профіль інтересів користувача розраховується як зважена сума профілів інтересів об'єктів, де вагами ве значення, розраховані на попередньому кроці алгоритму.

У результаті були отримані матриця з профілями інтересів об'єктів та матриця з профілями інтересів користувачів. Розрахувавши ці дві матриці ми отримали всі необхідні для нашої системи дані, а саме: граф людей, зв'язки між користувачами та об'єктами (у нашій системі вони означають список груп, у яких перебуває користувач), матрицю інтересів користувачів та матрицю інтересів об'єктів системи.

Для більшої зручності тестування та через підхід до тестування базового методу, описаний його автором, вирішено із обраного набору даних відібрати два менших підграфа друзів із загального графа друзів, а також групи, у яких перебувають користувачі з відібраних підграфів. Для отримання ширшої картини вирішено перший підграф обмежити 15-ма користувачами, між якими встановлено 38 двосторонніх зв'язків, а другий 117-ма користувачами, між якими встановлено 332 двосторонні зв'язки. Структуру меншого підграфа наведено на рисунку 4.1.

Гістограма кількостей друзів для кожного користувача відібраного підграфа наведена на рисунку 4.2.

Як показано на рисунку 4.2 більшість користувачів у відібраному підграфі, а саме восьмеро, мають всього одного друга, один користувач має двох друзів, один має трьох друзів, ще один має чотирьох друзів, троє мають по п'ять друзів і один користувач має аж шість друзів. Це дає у середньому 2,53 друга на кожного користувача, що є доволі невеликим показником і може вказувати на розрідженість соціальних зв'язків всередині відібраної множини користувачів.

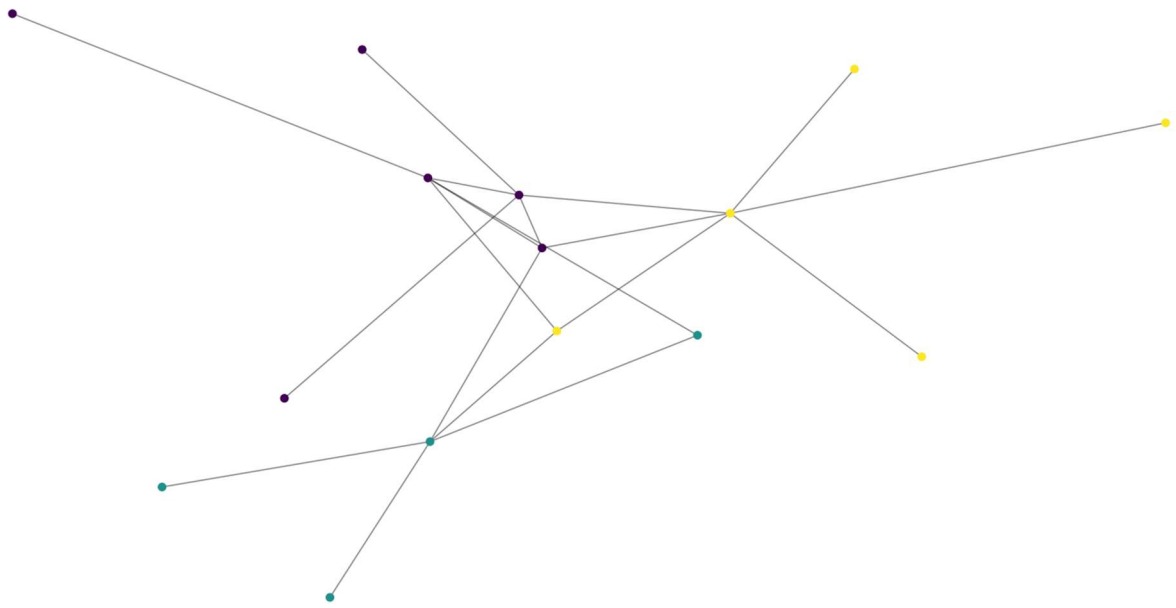


Рисунок 4.1 – Перший тестовий підграф

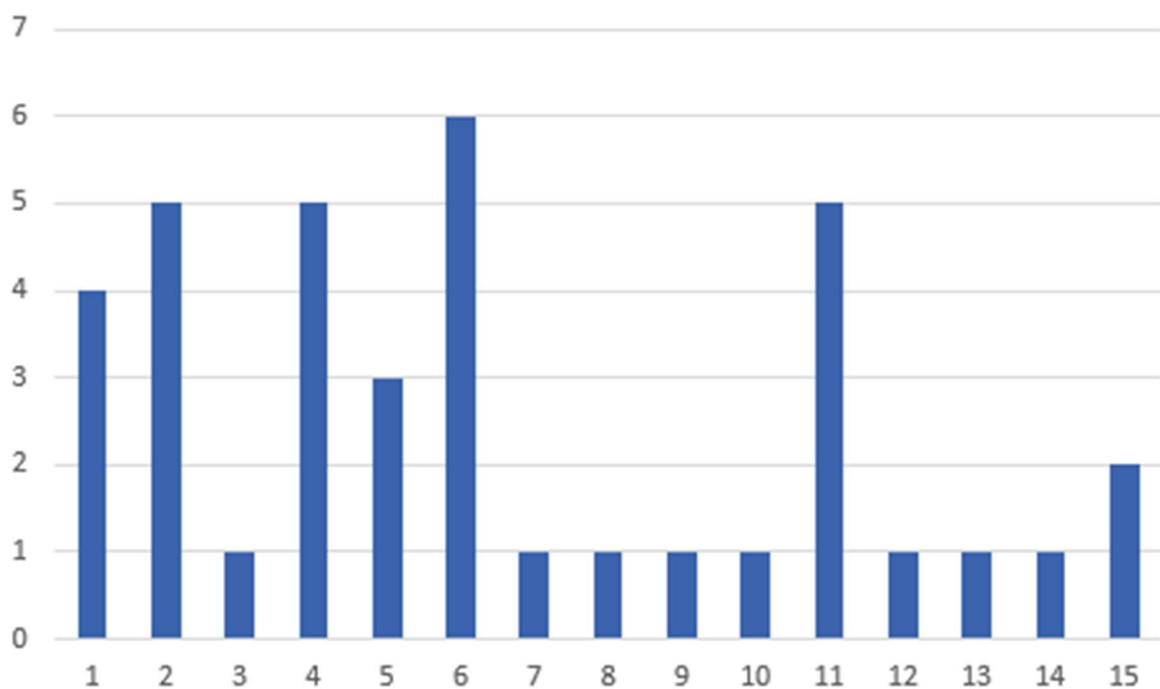


Рисунок 4.2 – Гістограма кількостей друзів для кожного користувача першого підграфа

Для цих користувачів були сформовані списки груп, учасниками яких вони є. Загалом відібрано 504 групи. Гістограма кількостей груп, у яких перебувають користувачі, наведена на рисунку 4.3.

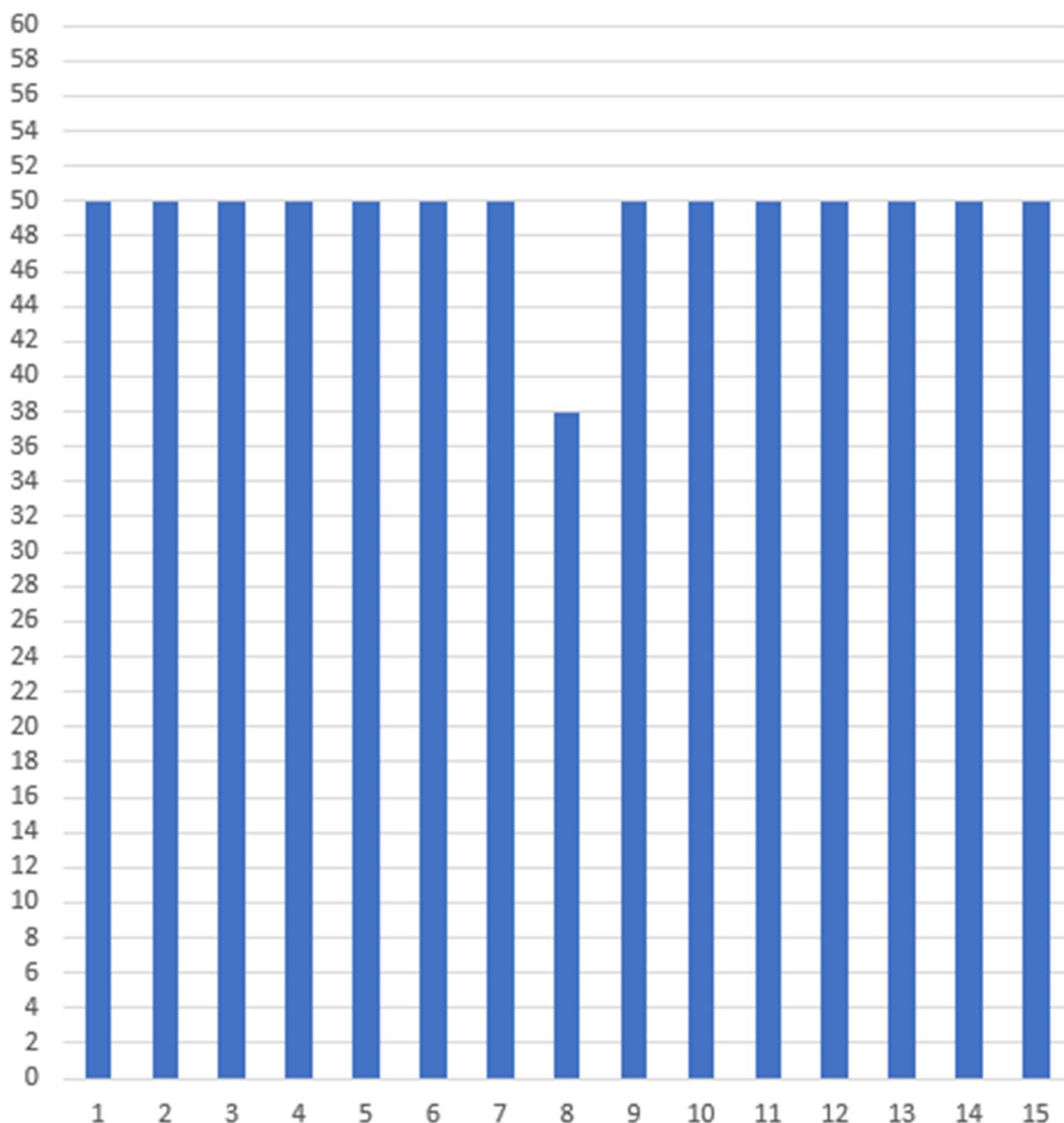


Рисунок 4.3 – Гістограма кількості груп, у яких перебувають користувачі першого підграфа

У відібраному підграфі користувачів, кожен користувач у середньому знаходиться у 49,2 групах, що практично у 10 разів менше за загальну кількість відібраних груп.

Як другий підграф вирішено відібрати 117 користувачів та всі присутні зв'язки між ними. Структура другого підграфа наведена на рисунку 4.4.

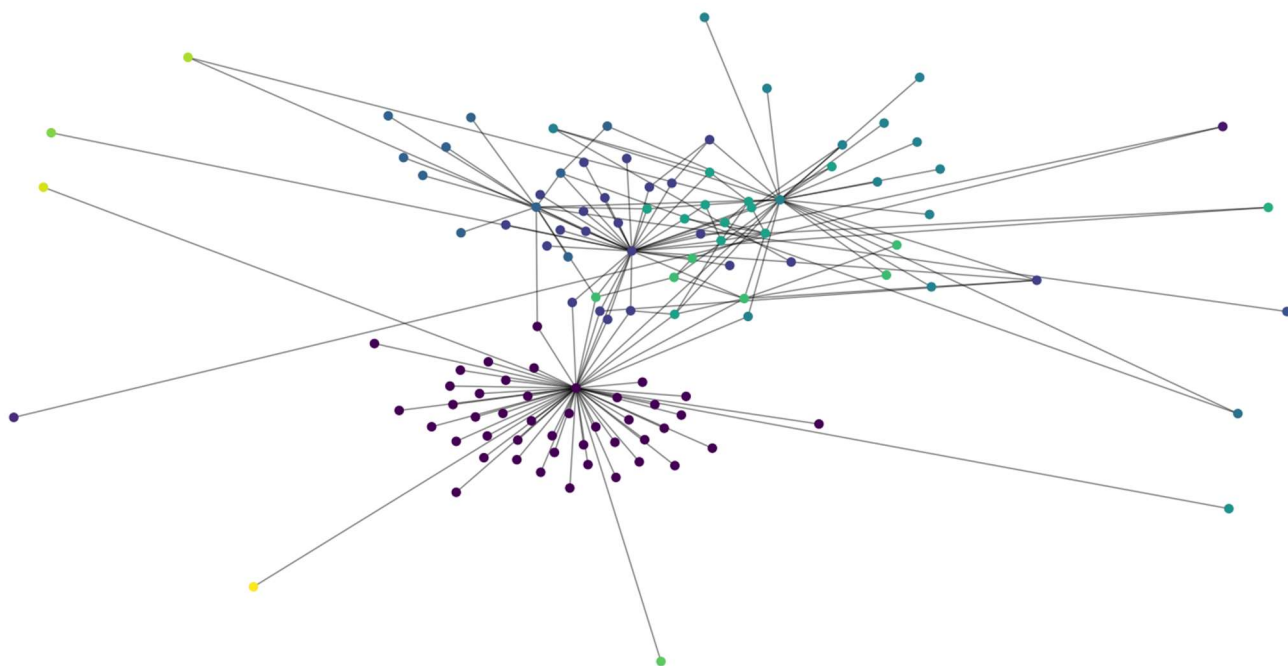


Рисунок 4.4 – Другий тестовий підграф

Гістограма кількостей друзів для кожного користувача другого підграфа наведена на рисунку 4.5.

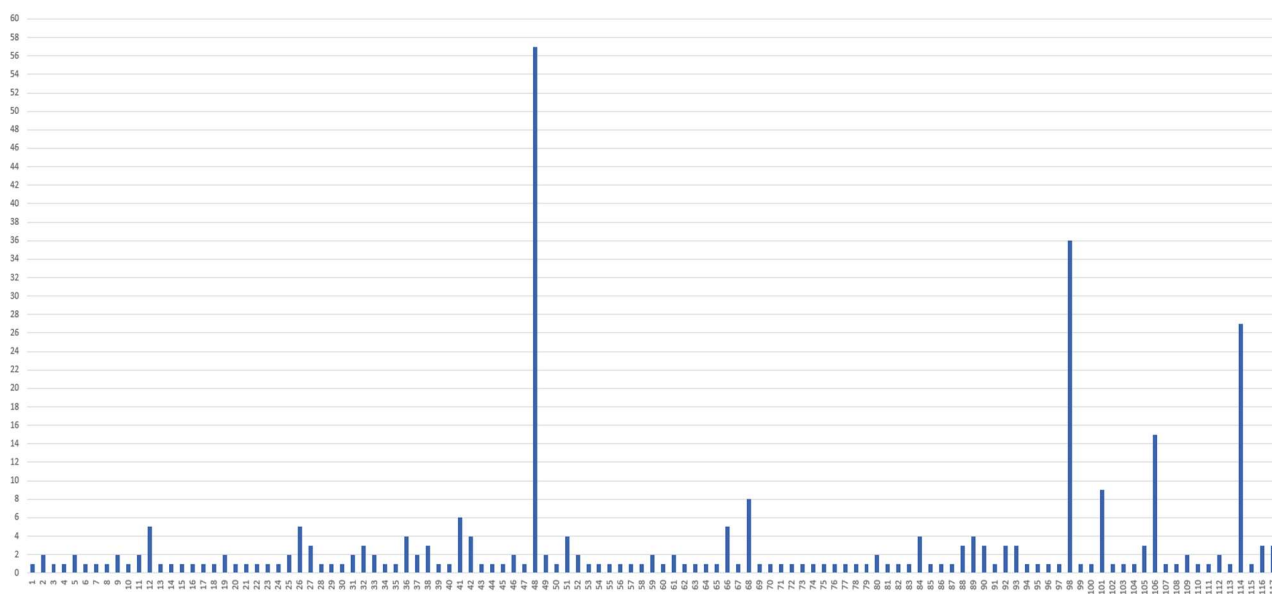


Рисунок 4.5 – Гістограма кількостей друзів для кожного користувача другого підграфа

У другому підграфі кількість користувачів, у яких визначений лише один друг, рівна 75, кількість користувачів, у яких є два друга, рівна 17, кількість

користувачів, у яких є три друга, рівна 10, чотири друга мають 5 користувачів, п'ять друзів є у 6 користувачів, шість друзів у одного користувача, вісім друзів є лише у одного користувача, дев'ять друзів має лише один користувач, п'ятнадцять друзів визначено лише для одного користувача, двадцять сім друзів є лише у одного користувача, тридцять шість друзів має лише один користувач і п'ятдесят сім друзів має лише один користувач. У результаті для цього підграфа середня кількість друзів більша і рівна 2,83.

Аналогічно до першого відібраного підграфа, для всіх відібраних користувачів були відібрані групи, до яких вони входять. Загалом відібрано 1755 груп. Гістограма кількостей груп, у яких перебувають користувачі, наведена на рисунку 4.6.

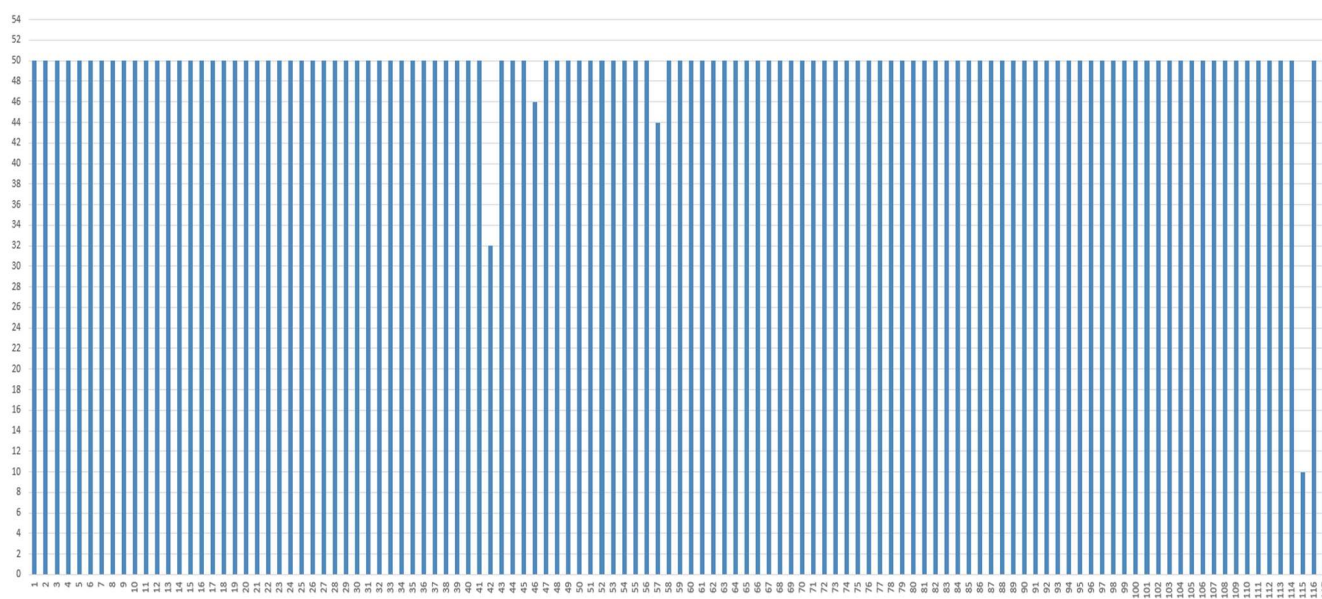


Рисунок 4.6 – Гістограма кількості груп, у яких перебувають користувачі другого підграфа

У відібраноу підграфі користувачів, кожен користувач у середньому знаходиться у 49,41 групах, що у 35,5 разів менше за загальну кількість відібраних для всіх користувачів груп.

У цьому підрозділі був описаний базовий набір даних, способи обробки наявних у ньому даних для отримання необхідних для нашої системи факторів,

були відібрані два підграфу користувачів різного розміру для кращої оцінки результативності запропонованого методу, а також описані відповідні цим підграфам множини груп, до яких входять користувачі.

4.3 Постановка експерименту та отримані результати

Постановка експерименту є важливою частиною перевірки розробленого програмного продукту, оскільки вона визначає схему, за якою проводиться тестування запропонованого рішення. Експеримент із системою розрахунку рекомендацій людей вирішено поставити наступним чином:

- вибрати із тестової множини підграфу людей, у якому якого будуть розраховуватися рекомендації;
- у тестовому підграфі випадковим чином видаляються ребра, що означають дружбу між користувачами;
- для пари вершин графу людей (назвемо їх F_1 та F_2), для яких видалено ребро (дружбу), розраховується список рекомендованих людей;
- розраховується середнє арифметичне позицій цих користувачів у списках рекомендованих людей;
- ребро між F_1 та F_2 повертається до графу.

У результаті із отриманих значень середніх арифметичних позицій розраховується середнє значення. Чим меншим воно буде, тим більшою є точність рекомендацій системи. У експериментах нами вирішено виконати 10 ітерацій по 100 експериментів кожна, розрахунок середніх арифметичних позицій розраховувати у кінці кожної ітерації, а у кінці розрахувати мінімальне, максимальне, середнє значення та дельту. Сумарно для кожного набору ваг факторів, які враховує система, буде проведено 1000 випадкових проходжень про графу користувачів, що дозволить більш-менш точно оцінити можливості запропонованого рішення.

Експеримент із системою розрахунку рекомендацій груп вирішено поставити схожим чином:

- вибрати випадковим чином із тестової множини для кожного користувача набір груп, у яких він перебуває;
- у для випадкової пари (користувач, об'єкт) (назвемо їх U_1 та O_1) видаляється зв'язок між ними;
- для цього користувача U_1 , для якого була видалена група, розраховується список рекомендованих груп;
- система запам'ятовує номер групи O_1 у списку рекомендованих груп;
- зв'язок між користувачем U_1 та групою O_1 відновлюється.

У результаті із отриманих значень позицій груп у списку рекомендацій розраховується середнє значення. Чим меншим воно буде, тим більшою є точність рекомендацій системи. Експериментів вирішено виконати 10 ітерацій по 100 експериментів кожна, розрахунок середніх арифметичних позицій розраховувати у кінці кожної ітерації, а у кінці розрахувати мінімальне, максимальне, середнє значення та дельту. Проходження по обраних випадковим чином парах (користувач, група) 1000 разів, на нашу думку, дозволить краще оцінити систему на даних різної якості.

Експерименти для запропонованого методу розрахунку рекомендацій людей вирішено проводити у два етапи:

- провести експеримент на першому підграфі;
- провести експеримент на другому підграфі.

У ході обох експериментів вирішено змінювати вагові коефіцієнти наступним чином:

- сформувати послідовність з усіх варіацій вагових коефіцієнтів λ_1 , λ_2 та λ'_3 із рівняння (10);
- для кожної варіації значень вагових коефіцієнтів λ_1 , λ_2 та λ'_3 провести оцінку точності розробленої системи, змінюючи значення коефіцієнта γ із формули (12) у діапазоні $[0,5;1]$ з кроком 0,05 на 1000 ітерацій розрахунку рекомендацій.

Значення вагових коефіцієнтів λ_1 , λ_2 та λ_3 представляють собою набір чисел, кожне у діапазоні $[0,1;0,8]$, а їх сума завжди рівна 1. Крок зміни цих вагових коефіцієнтів рівний 0,05. У результаті отримано 118 трійок значень вагових коефіцієнтів, і для кожної такої трійки значень розраховано точність рекомендацій з різними значеннями параметра γ та параметра k , що відповідає за кількість друзів з найменшим зв'язком для формули (12) і приймає значення з діапазону. У такому випадку для охоплення всіх можливих значень ваг сумарно необхідно провести 594 експерименти.

За результатами проведеного на першому підграфі експерименту для оцінки точності рекомендацій запропонованої системи були виявлені 10 наборів значень вагових коефіцієнтів та параметрів системи, для яких отримана точність рекомендацій найкраща. Ці значення наведено у таблиці 4.1.

Таблиця 4.1 – Десять наборів параметрів за значенням точності рекомендацій

λ_1	λ_2	λ_3	γ	Кількість найменш подібних друзів	Мінімальне значення	Максимальне значення	Середнє значення	Δ
0,1	0,8	0,1	0,95	3	5,03	6,27	5,578	1,24
0,1	0,8	0,1	0,85	3	4,97	6,39	5,635	1,42
0,1	0,8	0,1	0,5	2	5,16	6,2	5,644	1,04
0,1	0,8	0,1	1	3	5,13	6,16	5,651	1,03
0,1	0,75	0,15	0,75	3	5,32	6,04	5,652	0,72
0,1	0,8	0,1	0,55	3	5,16	5,96	5,655	0,8
0,1	0,8	0,1	0,6	3	4,72	6,06	5,666	1,34
0,1	0,8	0,1	0,6	2	5,38	6,06	5,667	0,68
0,1	0,8	0,1	0,5	3	5,39	6,22	5,686	0,83
0,1	0,8	0,1	0,55	2	5,08	6,2	5,692	1,12

За результатами проведеного експерименту на другому підграфі для оцінки точності рекомендацій запропонованої системи були виявлені 10 наборів значень вагових коефіцієнтів та параметрів системи, для яких отримана точність рекомендацій найкраща. Ці набори параметрів наведено у таблиці 4.2.

Таблиця 4.2 – Десять наборів параметрів за значенням точності рекомендацій

λ_1	λ_2	λ_3	γ	Кількість найменш подібних людей	Мінімальне значення	Максимальне значення	Середнє значення	Δ
0,1	0,75	0,15	0,5	3	36,44	41,86	39,31	5,42
0,1	0,75	0,15	0,7	3	34,73	44,43	40,064	9,7
0,1	0,75	0,15	0,9	3	37,5	44,7	40,087	7,2
0,1	0,7	0,2	0,7	2	37,35	43,57	40,447	6,22
0,1	0,7	0,2	1	3	34,79	45,76	40,589	10,97
0,1	0,8	0,1	0,5	2	37,64	43,79	40,704	6,15
0,1	0,8	0,1	0,8	3	34,99	45,65	40,747	10,66
0,1	0,8	0,1	0,85	2	34,32	45,03	40,777	10,71
0,1	0,75	0,15	0,6	2	35,86	45,26	40,96	9,4
0,1	0,8	0,1	0,75	3	35,77	46,78	41,094	11,01

Для порівняння із результатами роботи розробленого методу, на цих же тестових даних був протестований базовий метод. У базовий метод була внесена єдина зміна – замість кількості програм, якими користуються обидва користувачі, система враховує кількість тегів, якими користувалися обидва користувачі. Ця зміна вимушена через відсутність у тестовому наборі даних необхідної для роботи повністю незміненого базового методу. Проте, на нашу думку, запропонована альтернатива не повинна погіршити точність базового алгоритму так, щоб це помітно при проведенні декількох експериментів. Результати роботи базового методу на обох підграфах наведено у таблиці 4.3.

Таблиця 4.3 – Результати роботи на експериментальних даних базового алгоритму рекомендації людей

λ_1	λ_2	λ_3	Мінімальне значення	Максимальне значення	Середнє значення	Δ	Номер підграфа
0,5	0,3	0,2	6,29	7,15	6,792	0,86	1
0,5	0,3	0,2	39,55	49,38	45,245	9,83	2

Запропонований метод рекомендації груп показав себе гірше у порівнянні з методом рекомендації людей. Дані точності з експерименту розрахунку рекомендацій груп на обох наборах даних наведено у таблиці 4.4.

Таблиця 4.4 – Точність рекомендацій методу розрахунку рекомендацій груп

Мінімальне значення	Максимальне значення	Середнє значення	Δ	Номер набору даних
99,18	143,95	118,345	44,77	1
235,07	338,76	291,958	103,69	2

У підрозділі описано спосіб постановки експерименту та оцінки точності рекомендацій розробленої системи, а також механізми зміни ваг різних факторів для виявлення набору ваг, який забезпечить найкращу точність.

4.4 Оцінка ефективності моделей та методів для вирішення задачі

Із значень, наведених у таблицях 4.1 та 4.2, можна зробити висновок, що запропонована система найкраще прогнозує наявні соціальні зв'язки при значеннях ваг λ_1 , λ_2 та λ_3 , рівних 0,1, 0,7 та 0,15, параметру γ , рівному 0,5 та значенню k , рівному 3, оскільки саме за таких значень розбіжність між мінімальною та максимальною точністю приймає найменші значення. При цьому, у порівнянні із базовим

алгоритмом, точність прогнозування видалених соціальних зв'язків на першому підграфі, покращена на 1,14, а точність прогнозування на другому підграфі покращена на 5,935, що вказує на позитивний вплив запропонованих змін. Крім цього невелика розбіжність між першим та останнім значеннями у таблицях 4.1 та 4.2 означають, що всі набори вагових коефіцієнтів та параметрів, які у них наведені, дозволяють прогнозувати видалені соціальні зв'язки з приблизно однаковою точністю.

На нашу думку, отриманим результатам можна довіряти з точки зору оцінки ефективності та точності розробленої системи, проте, для отримання максимально достовірних результатів необхідно провести тестування розробленої системи на реальних даних а ще краще за участі реальних користувачів, за рахунок чого можна буде зробити висновок щодо запропонованих рекомендацій людей, які знаходилися вище у списку за людей, зв'язок з якими був видалений під час експерименту. За відсутності наборів даних, які, на нашу думку, підходять для тестування краще, за обраний, запропонований нами варіант, на нашу думку, найкраще підходить для оцінки точності системи рекомендацій людей.

На першому відібраному наборі даних було отримане середнє значення точності розрахунку рекомендацій груп, рівне 118,345, а на другому наборі даних це значення складає 291,958. Отримані середні значення точності розрахунку рекомендацій груп можуть вказувати не лише на погану роботу методу розрахунку рекомендацій. Серйозну роль міг зіграти той факт, що у тестових вибірках кожному користувачеві відповідала невелика кількість груп, у яких він перебуває відносно кількості груп, які містяться у відібраних наборах даних. Якщо у першому наборі даних для тестування міститься загалом 504 групи, то серед всіх користувачів цієї першої вибірки середня кількість груп, у яких перебуває один користувач, рівна 49,2, що практично у 10 менше від всіх відібраних груп. Для другого набору даних відібрано 1755 груп при чому у середньому кожен користувач із другого тестового набору даних перебуває у 49,41 групах, що у 35,5 разів менше за загальну кількість груп. За відсутності відміток «не подобається» для кожного об'єкта, не можна точно стверджувати, що сформовані системою зв'язки абсолютно хибні. У ході тестування оцінка точності рекомендацій, отриманих від системи, базувалася на

тому, як вона зможе відновити існуючі зв'язки. Тому за аналогією з отриманими результатами тестування запропонованого методу розрахунку рекомендацій людей, для отримання максимально достовірного результату необхідно проводити тестування на реальних користувачах.

За результатами проведених експериментів з використанням виділених підграфів встановлено, що за сукупністю параметрів середньої точності та різниці між максимальною та мінімальною точністю на різних кроках, розроблений метод розрахунку рекомендацій людей при значеннях вагових коефіцієнтів λ_1 , λ_2 та λ_3 рівняння (10), рівних 0,1, 0,75 та 0,15 відповідно, досягає найкращої точності при встановленні значення вагового коефіцієнта γ формули (12) рівного 0,5 а кількості найменш подібних на користувача людей рівній 3. При таких значеннях вагових коефіцієнтів та параметрів запропонований метод має більшу точність рекомендацій – 5,652 на першому наборі даних у запропонованого методу проти 6,792 у базового та 39,31 на другому наборі даних проти 45,245 для базового, менше значення різниці між найбільшим та найменшим значеннями середньої точності на першій множині – 0,72 для запропонованого методу проти 0,86 для базового, а для другої тестової множини 9,4 для запропонованого методу проти 9,83 для базового. Середній приріст точності на першому наборі даних складає 1,14, а на другій тестовій множині 4,285. Менше розходження між найбільшим та найменшим значенням точності може сигналізувати про більшу стабільність точності рекомендацій, а той факт, що при цьому запропонована система має краще середнє значення точності, вказує на те, що вона може стабільно формувати якісні, з точки зору задоволення потреб користувача, рекомендації. За рахунок цього можна покращити досвід користувача із використання системи типу соціальної мережі через задоволення його потреб у нових знайомствах та, потенційно, у спілкуванні із новими, запропонованими рекомендаційною системою, користувачами.

4.5 Висновки

У розділі описано програмну реалізацію спроектованої системи, обрано мову програмування для її реалізації, основні бібліотеки, необхідні для її роботи, середовище розробки, описано набір даних для тестування розроблених систем, описано способи обробки даних для їх приведення до придатного для експериментів стану, визначено спосіб постановки експериментів з обома розробленими методами, оцінено та проаналізовано отримані у ході тестування результати.

За мову реалізації системи вирішено обрати мову Python, оскільки вона якнайкраще підходить для наших задач та має велику кількість готових бібліотек, які містять більшість необхідних нам алгоритмів та функцій.

Як середовище розробки вирішено обрати Microsoft Visual Studio, оскільки воно забезпечує зручний процес розробки програмного забезпечення, має вбудований широкий функціонал для тестування та відлагоджування як окремих програмних компонентів так і усієї системи у цілому.

Як тестовий набір даних, вирішено обрати набір даних, що моделює сайт Lastfm, оскільки він містить більшість необхідних для роботи системи параметрів, а також параметри, які необхідні для обчислення відсутніх у наборі факторів, а саме профілів інтересів груп та користувачів.

Для тестування системи вирішено провести окремі експерименти із модулем розрахунку рекомендацій людей та модулем розрахунку рекомендацій груп за подібними схемами.

Для проведення експериментів та перевірки гіпотез із обраного набору даних були відібрані дві підмножини зв'язків дружби між користувачами та набори груп, у яких вони перебувають.

У проведеному експерименті із розробленою моделлю системи розрахунку рекомендації людей були протестовані різні набори значень ключових параметрів системи.

Також описані способи, за якими були розраховані профілі інтересів користувачів та профілі інтересів об'єктів системи.

Шляхом проведення багатократних експериментів над розробленою моделлю рекомендації людей визначено низку оптимальних значень для ключових ваг, які регулюють вплив окремих параметрів на сформовані рекомендації. Кращими значеннями ваг факторів подібності набору людей, що розраховується за формулою (5), подібності набору груп, у яких перебувають обидва користувачі і яка розраховується за формулою (6), а також фактору подібності інтересів користувачів, що розраховується за формулою (3), виявилися 0,1, 0,75 та 0,15 відповідно. Шляхом вимірювань точності за різних значень параметру кількості найменш подібних на користувача людей, виявлено, що оптимальним її значенням є 3, при цьому вага параметру середньої сили зв'язку з користувачем та його друзями у такому випадку має бути рівна 0,5. Менш вдалим, але таким, що має право на життя, є варіант встановлення ваги середньої сили зв'язку між користувачем та усіма його друзями до значення 0,75.

У результаті експерименту із розробленою системою розрахунку рекомендацій людей на першій тестовій множині отримано середнє значення точності, рівне 5,65 та 39,31 на другій тестовій множині. Для базового алгоритму середні значення точності, розраховані на цих же даних за цим же принципом складають 6,792 та 45,245 для першого та другого тестового набору даних відповідно. Це означає, що запропоновані зміни у систему позитивно вплинули на точність розрахованих рекомендацій.

У результаті експерименту із розробленою системою розрахунку рекомендацій груп на першій тестовій множині отримано середнє значення точності рекомендацій, рівне 118,34, а на другій – 291,95. Оскільки значення не є близьким до еталонного, точність прогнозування наявних зв'язків розробленої системи не можна вважати високою.

Проте, таке значення точності може також сигналізувати про надто низьку зв'язність даних всередині тестової множини. Для отримання найточнішої оцінки запропонованого методу необхідно провести його тестування або на реальних

користувачах, або на наборі даних, у якому явно розмічено реакцію користувачів на кожен об'єкт системи.

Отримані у ході експериментів значення точності для розробленої системи на різних наборах даних дозволяють стверджувати, що наукова новизна полягає у тому, що:

- отримав подальший розвиток метод, на основі єдності, основна ідея якого полягає у поетапному вимірі міцності зв'язку у мережі і виявленні можливих зв'язків у цій мережі, які ще не створено; виявленні груп у мережі з використанням модулярності та рекомендації друзів, за рахунок включення фактору подібності інтересів та врахування двосторонньої сили зв'язку між користувачами, що дозволило покращити точність рекомендацій;

- набула подальшого розвитку формальна модель користувача сервісу розрахунку рекомендацій друзів, за рахунок її розширення профілями інтересів користувачів, що дало змогу покращити точність рекомендацій через точніше моделювання кожного окремого користувача;

- запропонована оригінальна технологія розробки програмної системи рекомендацій для знайомств людей та груп за інтересами у віртуальних спільнотах.

Практичне значення отриманих результатів можна виразити як:

- використання запропонованого підходу для соціальних мереж для формування точніших рекомендацій людям;

- поліпшення якості розрахованих рекомендацій вдосконаленого методу відносно оригінального.

У подальшому запропонований метод, на нашу думку, можна розвивати через розширення кількості факторів, які він враховує, або перегляду списку факторів, які були запропоновані. Крім цього, для факторів, які враховує система, можна запропонувати альтернативні способи їх зважування, а також способи знаходження їх ваг.

ВИСНОВКИ

За результатами виконаних теоретичних та практичних досліджень розроблено технологію розробки програмної системи рекомендацій для знайомств та груп за інтересами у віртуальних спільнотах.

У першому розділі виконано аналіз літератури та існуючих рішень та обрано алгоритм рекомендації людей, який якнайкраще підходить для рекомендації людей для нашого завдання та який проаналізовано на предмет наявних у ньому недоліків та описано способи їх усунення.

У другому розділі виконано функціональне моделювання систем розрахунку рекомендацій людей та груп, описано базовий метод розрахунку рекомендацій людей, описано конкретні зміни цього методу.

У третьому розділі були описані алгоритми, за якими працюють обидві системи, а також описано процес проектування їх програмної реалізації від розробки вимог до проектування модулів.

У четвертому розділі описано програмну реалізацію обох систем, використаний у експериментах набір даних, його розбиття на менші тестові множини, способи розрахунку даних, які відсутні у готовому наборі, але необхідні для функціонування системи, способи постановки експериментів для обох систем, а також зроблено аналіз та висновки із отриманих результатів експериментів.

У роботі поставлена мета усунути виявлені у методі розрахунку рекомендацій людей недоліки, досягнута шляхом включення у процес розрахунків нової інформації про інтереси користувачів та інакшому використанню старої інформації, а саме розрахунку двостороннього зв'язку замість одностороннього та врахування сили зв'язку із найменш подібними друзями у процесі розрахунку порогового значення.

Отримав подальший розвиток метод, на основі єдності, основна ідея якого полягає у поетапному вимірі міцності зв'язку у мережі і виявленні можливих зв'язків у цій мережі, які ще не створено; виявленні груп у мережі з використанням модулярності та рекомендації людей, за рахунок включення фактору подібності

інтересів та врахування двосторонньої сили зв'язку між користувачами, що дозволило покращити точність рекомендацій.

Набула подальшого розвитку формальна модель користувача сервісу розрахунку рекомендацій людей, з введенням профілів інтересів користувачів, що дало змогу покращити точність рекомендацій через точніше моделювання кожного окремого користувача.

Проведена порівняльна оцінка точності рекомендацій людей, сформованих базовим алгоритмом, та відповідна точність для модифікованого алгоритму, яка показала, що модифікований алгоритм розрахунку рекомендацій людей має більшу точність у порівнянні з базовим.

У подальшому розроблений метод розрахунку рекомендацій людей можна вдосконалити за рахунок включення у процес розрахунку рекомендацій нових факторів, які дозволять краще змоделювати користувачів системи.

Запропонований метод може бути рекомендований для впровадження у соціальних мережах для розрахунку рекомендацій людей з урахуванням соціальних зв'язків, активності всередині системи та інтересів користувачів.

За результатами виконаних досліджень було опубліковано тези доповіді на Міжнародній науковій конференції «Інформація, комунікація та суспільство» [18] та 1 тези на всеукраїнській науково-практичній конференції «Актуальні проблеми комп'ютерних наук» [26].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. J. Luo, “Three Essays On Friend Recommendation Systems For Online Social Networks”, The University of Wisconsin-Milwaukee, Milwaukee, Wisconsin, USA, 2016.
2. S. K. Gorripati, V. K. Vatsavayi, “Community-Based Collaborative Filtering to Alleviate the Cold-Start and Sparsity Problems”, International Journal of Applied Engineering Research, Vol. 12, No. 15, pp. 5022–5030, 2017.
3. Пономарев А. В. Обзор методов учета контекста в системах коллаборативной фильтрации / А. В. Пономарев. // Труды СПИИРАН. – 2013. – №30. – С. 169–188.
4. G. Adomavicius, A. Tuzhilin, “Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions”, IEEE Transactions on Knowledge and Data Engineering, Vol. 17, No. 6, pp.734-749, 2005.
5. Мелешко Є. В. Проблеми сучасних рекомендаційних систем та методи їх рішення / Є. В. Мелешко. // Системи управління, навігації та зв'язку. Збірник наукових праць. – 2018. – №4(50) – С. 120–124.
6. Меньшикова Н. Обзор рекомендательных систем и возможностей учета контекста при формировании индивидуальных рекомендаций / Н. .. Меньшикова, И. В. Портнов, И. Е. Николаев. // Научный журнал Academy. – 2016. – №6. – С. 20–22.
7. S. Lokesh, K. A. Ashok, “Study of Recommendation System for Web Portals”, International Journal of Computer Applications, Vol. 84, No. 9, pp. 1-6, 2013.
8. G. Ibtissem, H. Seridi-Bouchelaghem, H. Labar, A. K. Baareh, “Collaborative filtering recommendation based on dynamic changes of user interest”, Intelligent Decision Technologies, Vol. 9, pp. 271-281, 2015.
9. P. Bedi, P. Vashisth, “Interest Based Recommendations with Argumentation” Journal of Artificial Intelligence, Vol 4, pp. 119-142, 2011.
10. Y. Koren, R. Bell, and C. Volinsky, “Matrix Factorization Techniques for Recommendation Systems”, IEEE computer, Vol. 42, Issue 8, pp. 30-37, 2009.
11. A. Felfernig, J. Michael, N. Gerald, R. Floria, R. Stefan, S. Martin, “Basic Approaches in Recommendation Systems”, Springer Verlag, pp. 15-37, 2014.

12. B.Mehmood Khan, A. Mansha, F. Hassan Khan, S. Bashir, “Collaborative Filtering based Online Recommendation Systems: A Survey”, International Conference on Information & Communication Technologies, Vol. 7, pp. 22-28, 2017.

13. Y. Yunfei, Z. Yinghua, “Research on recommendation system based on interest clustering”, AIP Conference Proceedings, Vol. 1820, pp. 1-7, 2017.

14. X. Su and, T. Khoshgoftaar, “A Survey of Collaborative Filtering Techniques”, Advances in Artificial Intelligence, Vol. 12, pp.1-19, 2009.

15. Q. Yuan, S. Zhao, L. Chen, Y. Liu, S. Ding, X. Zhang, W. Zheng, “Augmenting collaborative recommender by fusing explicit social relationships”, Proceedings of the ACM RecSys'09 Workshop on Recommender Systems & the Social Web, Vol. 2009, pp. 49-56, 2009.

16. Z. Su, J. Yan, H. Ling, H. Chen, “Research on personalized recommendation algorithm based on ontological user interest model”, Journal of Computational Information Systems, Vol. 8, No. 1, pp. 169-181, 2012.

17. D. Li, Q. Lv, X. Xie, Shang li, H. Xia, T. Lu, N. Gu, “Interest-based real-time content recommendation in online social communities”, Knowledge Based Systems, pp 120-132, 2011.

18. Протоковський А.О. Аналіз стратегій рекомендаційних систем знайомств людей та груп за інтересами у віртуальних спільнотах/ А.О. Протоковський, Ю.В. Форкун // Матеріали 9-ї міжнародної наукової конференції «Інформація, комунікація, суспільство 2020». – Львів: Видавництво Львівської політехніки.– 2020. – № 9. – С. 154-155.

19. Y. Shruthy, K. R. Sreenimol, “FriendProbe - A New Friend Recommender System for Social Networks”, International Journal of Advanced Research in Computer and Communication Engineering, Vol. 4, pp. 511-515, 2015.

20. L. Bian, H. Holtzman, “Online Friend Recommendation through Personality Matching and Collaborative Filtering”, The Fifth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies, pp. 230-235, 2011.

21. Z.Wang, J. Liao, Q. Cao, “Friendbook: A Semantic-based Friend Recommendation System for Social Networks”, IEEE transactions on mobile computing, Vol. 14, pp. 538-551, 2015.

22. M. Jiang, P. Cui, R. Liu, “Social Contextual Recommendation”, In Proceedings of the 21st ACM international conference on Information and knowledge management (CIKM '12), pp. 45–54, 2012.

23. L. Gou, F. You, J. Guo, “SFViz: Interest-based Friends Exploration and Recommendation in Social Networks”, In Proceedings of the 2011 Visual Information Communication - International Symposium (VINCI '11), pp. 1–10, 2011.

24. J. Kwon, S. Kim, “Friend Recommendation Method using Physical and Social Context”, International Journal of Computer Science and Network Security, Vol.10, pp. 116–120, 2010.

25. S. Shamsu, “A cohesion based friend recommendstion system”, 3rd International Conference on Artificial Intelligence and Soft Computing, Vol. 3, pp. 109–124, 2017.

26. Протоковський А. О. Методологія розрахунку рекомендацій в рекомендаційних системах / А. О. Протоковський, Ю. В. Форкун // Збірник наукових праць за матеріалами XII Всеукраїнської науково-практичної конференції «Актуальні проблеми комп’ютерних наук АПКН-2020». – 2020. – № 7. – С. 237–241.

ДОДАТОК А

(обов'язковий)

ПРОГРАМНИЙ КОД ОСНОВНИХ МОДУЛІВ

А.1 Основні методи модуля розрахунку рекомендацій людей за модифікованим підходом

```

def linkStr(f1, f2, f3):
    return alpha * f1 + betha * f2 + gama * f3

def threshold(fromUserId, toUserId, friends_set):
    global lsMatr

    if len(friends_set[indexInSet(friends_set, fromUserId)][1]) == 0:
        tmp = 0
    else:
        tmp = (np.sum([linkStrength[fromUserId - 1] for linkStrength in lsMatr]) +
np.sum(lsMatr[fromUserId - 1]) - lsMatr[fromUserId - 1][fromUserId -
1])/len(friends_set[indexInSet(friends_set, fromUserId)][1])
        minK = getMinK(fromUserId, k)

    threshold = phi * tmp + (1-phi)* np.sum(minK)/ len(minK)
    return threshold

def checkFriendPosib(fromUserId, toUserId, friends_set, groups_set, inter_sim):
    val = linkStrength(fromUserId, toUserId, friends_set, groups_set, inter_sim)
    threshold1 = threshold(fromUserId, toUserId, friends_set)
    threshold2 = threshold(toUserId, fromUserId, friends_set)
    if val > min(threshold1, threshold2): return True
    return False

def linkStrength(fromUserId, toUserId, friendsSet, groupsSet, interestSim):
    if len(friendsSet[indexInSet(friendsSet, fromUserId)][1]) == 0: f1 = 0
    else:
        f1 = len(set.intersection(friendsSet[indexInSet(friendsSet, fromUserId)][1],
friendsSet[indexInSet(friendsSet, toUserId)][1])) /
len(friendsSet[indexInSet(friendsSet, fromUserId)][1])

        if len(groupsSet[indexInSet(groupsSet, fromUserId)][1]) == 0: f2 = 0
        else:
            f2 = len(set.intersection(groupsSet[indexInSet(groupsSet, fromUserId)][1],
groupsSet[indexInSet(groupsSet, toUserId)][1])) /
len(groupsSet[indexInSet(groupsSet, fromUserId)][1])
            f3 = interestSim
            ls1 = linkStr(f1, f2, f3)
            if len(friendsSet[indexInSet(friendsSet, toUserId)][1]) == 0: f1 = 0
            else:
                f1 = len(set.intersection(friendsSet[indexInSet(friendsSet, toUserId)][1],
friendsSet[indexInSet(friendsSet, fromUserId)][1])) /
len(friendsSet[indexInSet(friendsSet, toUserId)][1])

        if len(groupsSet[indexInSet(groupsSet, toUserId)][1]) == 0: f2 = 0

```

```

    else:
        f2 = len(set.intersection(groupsSet[indexInSet(groupsSet,toUserId)][1],
groupsSet[indexInSet(groupsSet,fromUserId)][1])) /
len(groupsSet[indexInSet(groupsSet,toUserId)][1])
        f3 = interestSim
        ls2 = linkStr(f1, f2, f3)
        res = (ls1 + ls2)/2
        return res

def getMinK(UserId, k):
    global lsMatr
    t1 = lsMatr[:,UserId - 1:UserId][:UserId]
    t1 = t1.T
    t2 = lsMatr[UserId - 1:UserId,UserId:]
    tmp = np.zeros((len(t1) + len(t2)))
    for i in range(0, len(t1)):
        tmp[i] = t1[0][i]
    for i in range(len(t1), len(t2)):
        tmp[i] = t2[0][i]

    arr = np.sort(tmp)[:k]
    return arr

def getGraphPartition(friend_graph):
    partition = community_louvain.best_partition(friend_graph)
    return partition

def calcFriendsReclList(users, targetUserId, friends_list, user_groups_list,
user_inter_matr, sim_matr, removedUserId = None):

    if removedUserId == None:
        return
    global lsMatr

    friends_sets = [[int(val), dm.getFriends(int(val), friends_list, int(val) - 1)] for
val in users]

    group_sets = [[ int(val), dm.getFriends(int(val), user_groups_list, int(val) - 1)] for
val in users]

    friends_sets[indexInSet(friends_sets,targetUserId)][1].remove(removedUserId)
    friends_sets[indexInSet(friends_sets,removedUserId)][1].remove(targetUserId)

    calcLsMatr(users,friends_list,friends_sets, group_sets, sim_matr)

    linkStrengthArr = []

    for u in users:
        if u in friends_sets[indexInSet(friends_sets,targetUserId)][1] or u ==
targetUserId: continue
        ls = [u, round(linkStrength(targetUserId, u, friends_sets, group_sets,
sim_matr[targetUserId - 1][u - 1]), 6)]
        if checkFriendPosib(targetUserId, u, friends_sets, group_sets,
sim_matr[targetUserId - 1][u - 1]):
            linkStrengthArr.append(ls)

    linkStrengthArr = sorted(linkStrengthArr,key= lambda x: x[1], reverse = True)
    return linkStrengthArr

```

A.2 Основні методи модуля розрахунку рекомендацій груп

```

def calcSingleUser(userVector, group_inter_matr):
    userVector = np.array(userVector)
    group_inter_matr = np.array(group_inter_matr)

    predict = group_inter_matr.dot(userVector)

    return predict

def calcForAll(users_inter_matr, group_inter_matr):
    users_inter_matr = np.array(users_inter_matr)
    group_inter_matr = np.array(group_inter_matr)

    predict = users_inter_matr.dot(group_inter_matr.T)

    return predict

def calcRecomSingle(predictedRatings):
    res = [[0,-1]]
    for i, rating in enumerate(predictedRatings):
        for j in range(0, len(res)):
            if rating >= res[j][1]:
                res.insert(j,[i + 1,rating] )
                break
        if res[j][1] != rating:
            res.append([i + 1,0])
    res.remove([0,-1])
    return res

def calcRecomForAll(predictedRatings):
    res = []
    for user in predictedRatings:
        res.append(calcRecomSingle(user))
    return res

def calcSingleReclis(userVector, group_inter_matr):
    pr = calcSingleUser(userVector, group_inter_matr)

    return calcRecomSingle(pr)

```

A.3 Основні методи модуля розрахунку рекомендацій людей за базовим підходом

```

def linkStrength(f1, f2, f3):
    return alpha * f1 + betha * f2 + gama * f3

def threshold(fromUserId, toUserId, friends_set):
    global lsMatr
    if len(friends_set[indexInSet(friends_set, fromUserId)][1]) == 0:
        tmp = 0
    else:
        tmp = (np.sum([linkStrength[fromUserId - 1] for linkStrength in lsMatr]) +
np.sum(lsMatr[fromUserId - 1]) - lsMatr[fromUserId - 1][fromUserId -
1])/len(friends_set[indexInSet(friends_set, fromUserId)][1])
    threshold = tmp

```

```

return threshold

def checkFriendPosib_usingTags(fromUserId, toUserId, friends_set, groups_set, tagSet):
    val = linkStrength_usingTags(fromUserId, toUserId, friends_set, groups_set, tagSet)
    if val > threshold(fromUserId, toUserId, friends_set): return True
    return False

def checkFriendPosib_usingInterestSim(fromUserId, toUserId, friends_set, groups_set,
inter_sim):
    val = linkStrength_usingInterestSim(fromUserId, toUserId, friends_set, groups_set,
inter_sim)
    if val > threshold(fromUserId, toUserId, friends_set): return True
    return False

def linkStrength_usingTags(fromUserId, toUserId, friendsSet, groupsSet, tagSet):
    if len(friendsSet[indexInSet(friendsSet, fromUserId)][1]) == 0: f1 = 0
    else:
        f1 = len(set.intersection(friendsSet[indexInSet(friendsSet, fromUserId)][1],
friendsSet[indexInSet(friendsSet, toUserId)][1])) /
len(friendsSet[indexInSet(friendsSet, fromUserId)][1])

        if len(groupsSet[indexInSet(groupsSet, fromUserId)][1]) == 0: f2 = 0
        else:
            f2 = len(set.intersection(groupsSet[indexInSet(groupsSet, fromUserId)][1],
groupsSet[indexInSet(groupsSet, toUserId)][1])) /
len(groupsSet[indexInSet(groupsSet, fromUserId)][1])
            if len(groupsSet[indexInSet(groupsSet, fromUserId)][1]) == 0: f3 = 0
            else:
                f3 = len(set.intersection(tagSet[indexInSet(tagSet, fromUserId)][1],
tagSet[indexInSet(tagSet, toUserId)][1])) / len(tagSet[indexInSet(tagSet, fromUserId)][1])
                ls1 = linkStrength(f1, f2, f3)
                res = ls1
                return res

def linkStrength_usingInterestSim(fromUserId, toUserId, friendsSet, groupsSet, interSim):
    if len(friendsSet[indexInSet(friendsSet, fromUserId)][1]) == 0: f1 = 0
    else:
        f1 = len(set.intersection(friendsSet[indexInSet(friendsSet, fromUserId)][1],
friendsSet[indexInSet(friendsSet, toUserId)][1])) /
len(friendsSet[indexInSet(friendsSet, fromUserId)][1])

        if len(groupsSet[indexInSet(groupsSet, fromUserId)][1]) == 0: f2 = 0
        else:
            f2 = len(set.intersection(groupsSet[indexInSet(groupsSet, fromUserId)][1],
groupsSet[indexInSet(groupsSet, toUserId)][1])) /
len(groupsSet[indexInSet(groupsSet, fromUserId)][1])
            f3 = interSim
            ls1 = linkStrength(f1, f2, f3)
            res = ls1
            return res

def getGraphPartition(friend_graph):
    partition = community_louvain.best_partition(friend_graph)
    return partition

def indexInSet(target_set, targetId):
    for i in range(0, len(target_set)):
        if target_set[i][0] == targetId:
            return i
    raise ValueError("Target set does not contain targetId")

```

A.4 Основні методи модуля обробки даних

```

def readUserFriends():
    file_name = "user_friends.dat"

    reader = open(data_source_path + file_name , "r")

    friends_file = reader.readlines()[1:]
    reader.close()
    buffer = None
    friends_list = [["User ID", "Friend ID"]]
    for row in friends_file:
        buffer = row.split('\t')
        friends_list.append([int(buffer[0]), int(buffer[1][:-1])])
    return friends_list

def readUserGroup():
    file_name = "user_group.dat"

    reader = open(data_source_path + file_name , "r")

    user_groups_file = reader.readlines()[1:]
    reader.close()
    buffer = None
    user_groups_list = [["User ID", "Artist ID", "Weight"]]
    for row in user_groups_file:
        buffer = row.split('\t')
        user_groups_list.append([int(buffer[0]), int(buffer[1]), buffer[2][:-1]])
    return user_groups_list

def readItemInterMatr():
    group_inter_matr = []
    file_name = "ItemInterestMatrix.dat"
    group_inter_file = None

    with open(data_source_path + file_name , "r") as reader:
        group_inter_file = reader.readlines()

    buffer = None
    for row in group_inter_file:
        buffer = row.split('\t')
        group_inter_matr.append([round(float(val),6) for val in buffer])

    return group_inter_matr

def readUserInterMatr():
    user_inter_matr = []
    file_name = "UserInterestMatrix.dat"
    user_inter_file = None

    with open(data_source_path + file_name , "r") as reader:
        user_inter_file = reader.readlines()

    buffer = None
    for row in user_inter_file:
        buffer = row.split('\t')
        user_inter_matr.append([round(float(val),6) for val in buffer])

```

```

    return user_inter_matr

def readUsers():
    file_name = "oldNewUserID.dat"

    reader = open(data_source_path + file_name , "r")

    users_file = reader.readlines()[1:]
    reader.close()
    users_list = [["Old ID", "New ID"]]
    buffer = None
    for row in users_file:
        buffer = row.split('\t')
        users_list.append([int(buffer[0]), int(buffer[1][:-1])])
    return np.array(users_list)

def read_sets(path):
    fullPath = data_source_path + path

    with open(fullPath, "r") as reader:
        set_file = reader.readlines()[1:]
        reader.close()

    set_to_return = []
    for line in set_file:
        buffer = line.split(':')
        set_to_return.append([int(buffer[0]), set(int(val) for val in
buffer[1].split('\t'))])

    return set_to_return

def writeToFile(data, fullPath):
    if os.path.exists(fullPath) and os.path.getsize(fullPath) is not 0:
        os.remove(fullPath)
    with open(fullPath, "w+", encoding="utf-8") as writer:
        for line in data:
            line = np.array(line)
            tmp = "\t".join([str(value) for value in line]) + "\n"
            writer.write(tmp)

def saveSets(setToSave, path):
    fullPath = data_source_path + path
    if os.path.exists(fullPath) and os.path.getsize(fullPath) is not 0:
        os.remove(fullPath)
    with open(fullPath, "w+", encoding="utf-8") as writer:
        writer.write("Id\tSet\n")
        for line in setToSave:
            writer.write(str(line[0]) + ":" + "\t".join([str(value) for value in line[1]]
+ "\n"))

    return

def getFriends(userId, friends_list, index):
    sub = set()
    for id in range(index, len(friends_list)):
        if int(friends_list[id][0]) < userId: continue
        if int(friends_list[id][0]) > userId: break
        sub.add(int(friends_list[id][1]))

    return sub

```

ДОДАТОК Б
(обов'язковий)

КОПІЇ НАУКОВИХ ПУБЛІКАЦІЙ



Матеріали 9-ої Міжнародної
наукової конференції

"Інформація, комунікація, суспільство 2020"

2020

Україна, Львів

<i>Зоряна Куньч, Ірина Марчук. Явище детермінологізації в романі Ліни Костенко “Записки українського самашедшого”</i>	130
<i>Ігор Кульчицький, Олена Фляк. Розподіл довжини слова в повісті Романа Іваничука “Як тиха ніч пов’є долину”</i>	132
<i>Зоряна Куньч, Юрій Берендій. Тематична класифікація детермінологізмів у романі Ю. Андруховича “Московіада”</i>	134
<i>Зоряна Куньч, Анна Шаблій. Медична термінологія в суспільно-політичному дискурсі (на прикладі статті Ліни Костенко “Україна як жертва і чинник глобалізації катастроф”)</i>	136
Психологічна та педагогічна компоненти документно-інформаційної діяльності	138
<i>Людмила Чередник. Кроскультурна комунікація як важливий чинник підготовки сучасного фахівця</i>	139
<i>Ніна Сорока, Наталія Химіця. Етико-психологічні аспекти комунікації в умовах дистанційного навчання</i>	141
Соціальні комунікації	143
<i>Anna Granat. Power of the Media as Part of the Communication Effects of Media Reception</i>	144
<i>Ivanna Sabadash. SMM as a Channel for Attracting Applicants to the Educational Institution</i>	146
<i>Ольга Антоник. Книга як комунікаційний інструмент: потенціал читацьких можливостей</i>	148
<i>Оксана Ількова. Перелік відомостей, що становлять службову інформацію: труднощі складання та доступу</i>	150
<i>Надія Зубко. Українська воєнна книга у системі сучасних засобів соціальної комунікації</i>	152
<i>Андрій Протоковський, Юрій Форкун. Аналіз стратегій рекомендаційних систем знайомств людей та груп за інтересами у віртуальних спільнотах</i>	154
<i>Olha Khytutsia, Galyna Hnat, Volodymyr Babyak. The analysis of factors of social interaction in the design of housing for highly skilled migrant workers</i>	156
<i>Ольга Стадник, Ольга Антоник, Надія Зубко. Товарознавчі аспекти соціокультурного буття книги</i>	158
<i>Олександр Морушко, Наталія Химіця. Ефективний поділ колективу на малі групи за допомогою соціонічного аналізу</i>	160
<i>Марія Комова, Аліна Петрушка. Комунікаційна взаємодія медіа та установи</i>	162
<i>Уляна Дихтярук, Наталія Вовк. Веб-сайт як засіб комунікативної діяльності органів студентського самоврядування</i>	164
<i>Юлія Якимович, Ольга Трач. Інформаційно-рекламне забезпечення меблевого виробництва</i>	166

Аналіз стратегій рекомендаційних систем знайомств людей та груп за інтересами у віртуальних спільнотах

Андрій Протоковський

Кафедра ПЗ
Хмельницький національний університет
Хмельницький, Україна
doc.emmett@ukr.net

Abstract. In this article were considered current problems of recommender systems. Were discussed an existing ways to solve some of them. Were chosen the most suitable strategy for recommender system of meeting and community recommender system.

Ключові слова: рекомендаційна система, фільтрація вмісту, колаборативна фільтрація, гібридна фільтрація.

Все більше інтернет ресурсів, що з'являються сьогодні, намагаються якнайкраще обробляти запити користувача. Для того, щоб користувач отримував саме те, що йому підходить найбільше, розробники реалізують системи рекомендацій контенту. Від моменту виникнення таких систем були розроблені різні підходи та комбінації підходів для побудови найкращих алгоритмів рекомендації контенту. Задачею дослідження є аналіз актуальних стратегій рекомендаційних систем, визначення їх основних переваг та недоліків з метою визначення оптимальної стратегії рекомендаційної системи знайомств людей та спільнот.

АНАЛІЗ СТРАТЕГІЙ РЕКОМЕНДАЦІЙНИХ СИСТЕМ

Задачі формування рекомендацій базуючись на персональних особливостях користувача з'явилися ще в 90-ті роки ХХ ст. Необхідність розробки рекомендаційних систем виникла через зростання обсягу інформації в мережі і як наслідок неможливість її повного вивчення окремим користувачем, а також мала слугувати способом максимального задоволення потреб

Юрій Форкун

Кафедра інженерії програмного забезпечення
Хмельницький національний університет
Хмельницький, Україна
forkun@ridne.net

користувача шляхом підбору контенту, який найкраще для цього підходить.

Існує три основних стратегії рекомендаційних систем [1]:

- Фільтрація вмісту або контентна фільтрація;
- колаборативна фільтрація;
- гібридна фільтрація.

Фільтрація вмісту або контентна фільтрація в загальному випадку використовує для прогнозування профілі об'єктів та профілі користувачів, які складаються із важливих для предметної області характеристик та історії активності відповідно. Її недоліком можна назвати те, що така система буде формувати рекомендації, підбираючи для користувача такі об'єкти, які схожі до об'єктів, що обиралися раніше.

Колаборативна фільтрація в загальному випадку здійснює прогнозування оцінки контенту конкретного користувача на підставі наявних оцінок групи користувачів. В її основі лежить припущення, що група користувачів, які однаково оцінювали деякі предмети в минулому, однаково оцінять предмети у майбутньому [2]. На цю стратегію повністю або частково опирається велика кількість вдалих існуючих рішень, проте вона має ряд недоліків. Недоліками колаборативної фільтрації зазвичай виступають:

- масштабованість – за наявності великої кількості користувачів та об'єктів розрахунок рекомендацій потребує великих обчислювальних потужностей;
- холодний старт – відсутність достатнього об'єму інформації по нових користувачах

та об'єктах для розрахунку списку рекомендацій;

- розрідженість – недостатня кількість оцінок конкретних об'єктів.

Гібридна фільтрація зазвичай представляється як комбінація колаборативної та контентної фільтрації [1]. Через об'єднання різних стратегій досягається перекривання недоліків кожної з систем і як результат краща точність підбору рекомендацій. В роботі [3] виділяється чотири способи поєднання колаборативної та контентної фільтрації:

- реалізація окремо колаборативної та контентної фільтрації та комбінування отриманих ними рекомендацій;
- включення деяких характеристик контентної фільтрації до колаборативної;
- включення деяких характеристик колаборативної фільтрації до контентної;
- розробка моделі, що об'єднує особливості обох стратегій.

Як недолік можна виділити більшу складність створення такої системи в порівнянні з кожною окремою стратегією.

В роботі [4] виділяються додаткові проблеми рекомендаційних систем, а саме стійкість до змін вподобань користувачів та властивостей об'єктів, а також проблема бульбашки фільтрів та розглядаються можливі шляхи їх вирішення.

Бульбашкою фільтрів називається проблема, коли через визначну роль історії активності та попередніх вподобань, користувач обмежується рекомендаціями однотипних об'єктів. Вирішення цієї проблеми запропоноване у роботі [4] через забезпечення рекомендацій таким списком властивостей:

- різноманітність – список рекомендацій має складатися з об'єктів різного типу;
- неочікуваність – несхожість на переглянуті раніше об'єкти;
- новизна – об'єкт має бути новий для користувача.

Важливо зазначити, що вирішивши цю проблему в такий спосіб, втрачається точність рекомендацій.

Для підвищення точності рекомендацій в роботі [5] пропонується враховувати контекст, в якому даються рекомендації.

Для збільшення адаптивності системи до змін вподобань користувачів або властивостей

об'єктів в роботі [4] розглядається використання методів машинного навчання.

ВИСНОВКИ

Як можна побачити існують різні підходи до вирішення проблем фільтрації в системах рекомендацій вмісту. Для досягнення результату їх потрібно реалізовувати з оглядкою на контекст предметної області.

Максимальної ефективності при розробці рекомендаційних систем можна досягти об'єднавши різні існуючі рішення за прикладом гібридних систем. При цьому, необхідно враховувати модель структурування даних віртуальних спільнот.

ЛІТЕРАТУРА

- [1] А. В. Пономарев, "Обзор методов учета контекста в системах коллаборативной фильтрации", Труды СПИИРАН, т. 7, вып. 30, сс. 169-188, март, 2013.
- [2] Su, X. and Khoshgoftaar, T., "A Survey of Collaborative Filtering Techniques", *Advances in Artificial Intelligence*, vol. 12, pp.1-19, Oct., 2009.
- [3] Adomavicius, G. and Tuzhilin, A., "Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions", *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp.734-749, June, 2005.
- [4] Є.В. Мелешко, Проблеми сучасних рекомендаційних систем та методи їх рішення", Системи управління, навігації та зв'язку. Збірник наукових праць, том 4, вип. 50, ст.120-124, вер., 2018.
- [5] Н. В. Меньшикова, И.В. Портнов, И.Е. Николаев, "Обзор рекомендательных систем и возможностей учета контекста при формировании индивидуальных рекомендаций", *ACADEMY*, т. 6, вып. 9, с. 20-22, 201
- [6] А.Пелешини, А.Вус, О.Марковець, «Побудова формальної моделі віртуальних спільнот як середовища соціокомунікативного протидорства», *Вчені записки таврійського національного університету імені В.І. Вернадського, Серія: технічні науки*, 2018, том 29 (68) № 4, с. 201-208.

Міністерство освіти і науки України
Хмельницький національний університет



ЗБІРНИК НАУКОВИХ ПРАЦЬ
за матеріалами XII всеукраїнської науково-практичної конференції
«Актуальні проблеми комп'ютерних наук АПКН-2020»

9-10 листопада 2020

Хмельницький 2020

Пирогов П. А., Чумаченко Д. І. Визначення ймовірності захворювання хворобами серця на основі методів Data Mining.....	225
Плацідим В. В., Міхалевський В. Ц. Рекомендаційна система пошуку житла та співмешканців в бюджетному сегменті.....	227
Придачук Ю. Р., Залуцька О. О., Кравчук Я. О. Параметри моделі тестового завдання при автоматизованому формуванні тестів	229
Прокопов Р. І., Манзюк Е. А., Скрипник Т. К. Інформаційна система для визначення подібності документів	232
Протоковський А. О., Форкун Ю. В. Методологія розрахунку рекомендацій в рекомендаційних системах.....	237
Пупченко О. О., Цололо С. О. Пересування колісного транспорту із використанням сплайнів в ігрових додатках на Unreal Engine	242
Рибчинський Б. О., Доброловський В. В., Медведчук В. Ю. Прогнозування завантаженості ресторану з використанням штучного інтелекту..	247
Римар П. В., Волошанов О. В. Розробка мобільного додатку «MyMoney».....	249
Римар П. В., Наскальний Д. С. Веб-додаток для прослуховування радіостанцій	253
Савенко Б. О., Каштальян А. С. Модель антивірусних інтелектуальних приманок в комп'ютерній мережі.....	257
Савінський В. В. Social Platform for Making Labeled Audio Datasets for Speech Synthesis of Human Voice.....	261
Сафоник А. П., Міщанчук М. М. Оптимізація маршруту MESH мережі засобами штучної нейронної моделі	265
Слободзян В. О., Мазурець О. В. Аналіз результатів автоматизованого пошуку ключових термінів у навчальних матеріалах	269
Смірнов О. П., Омельчук Р. В., Кисіль Т. М. Моніторинг у реальному часі за допомогою інтелектуальних агентів	275

УДК 004.043

Протоковський А. О., Форкун Ю. В.

*Хмельницький національний університет***МЕТОДОЛОГІЯ РОЗРАХУНКУ РЕКОМЕНДАЦІЙ В РЕКОМЕНДАЦІЙНИХ СИСТЕМАХ**

Проаналізовано публікації на теми рекомендаційні системи та факторизація матриць, описано основні методологічні підходи до формування рекомендацій, наведено найпоширеніші методи розрахунку подібності в рекомендаційних системах, перераховано популярні та ефективні методи факторизації матриць.

Were analysed publications on recommendatory systems and matrix factorization, were named the main methodological approaches to the formulation of recommendations, and were listed the most common methods for calculating similarities in recommendatory systems and popular and effective matrix factorization methods.

Рекомендаційні системи виконують важливу функцію фільтрації вмісту мережі інтернет, з метою підбору та ранжування лише того вмісту, який найкраще задовольнить конкретного користувача.

Різні наукові роботи по-різному класифікують методології рекомендаційних систем. У роботах [1, 2, 3] виділяють три базових методології рекомендаційних систем: фільтрація вмісту (вона ж контентна), колаборативна фільтрація та гібридна фільтрація. В загальному випадку рекомендаційні системи формують рекомендації використовуючи окремо, або комбінуючи методи розрахунку подібності, кластеризації та розкладання (факторизації) матриці.

Метою роботи є аналіз публікацій на тему «рекомендаційні системи» з метою виявлення та опису основних методологій, популярних методів розрахунку рекомендацій, що використовуються в рекомендаційних системах.

Розрахунок подібності в рекомендаційних системах може здійснюватися багатьма методами: Евклідова відстань, коефіцієнт кореляції Пірсона, коефіцієнт Жаккара, косинус подібності тощо. Робота [2] описує використання різних способів розрахунку подібності для різних ситуацій: косинус подібності у випадках, коли дані розріджені (деякі координати вектора рівні 0), в інших випадках рекомендується використовувати Евклідову відстань або Манхеттенську відстань. Саме тому в рамках цієї роботи вирішено висвітлити два способи – косинус подібності та Евклідову відстань. В загальному випадку ці методи дозволяють, за наявності матриці об'єктів (користувачів) X виду (об'єкт \times ознака) розмірністю n об'єктів та m ознак (наприклад інтересів), знайти матрицю подібності між

об'єктами S виду (об'єкт \times об'єкт) та розмірності $(n \times n)$. Кожен елемент s_{ij} цієї матриці буде означати міру подібності об'єкта x_i до об'єкта x_j .

Косинус подібності може виступати в якості основного або допоміжного методу для розрахунку подібності користувачів. Він виводиться із геометричної формули скалярного добутку двох векторів і розраховується за формулою:

$$sim_{ij} = \frac{\sum_{k=1}^m x_{ik} x_{jk}}{\sqrt{\sum_{k=1}^m x_{ik}^2} \sqrt{\sum_{k=1}^m x_{jk}^2}} \quad (1)$$

де sim_{ij} – значення, що відображає подібність двох користувачів;
 m – кількість ознак (інтересів);
 x_i – перший користувач;
 x_j – другий користувач.

Чим більше значення косинусу подібності, тим більше схожі два вектори, або в нашому випадку два користувачі або два об'єкти.

Евклідова відстань походить від геометричної формули відстані між двома векторами. Чим більша відстань між двома векторами, тим більше вони відрізняються. Міра подібності двох користувачів виражається через Евклідову відстань за формулою:

$$sim_{ij} = x_i - x_j = \sqrt{\sum_{k=1}^m (x_{ik} - x_{jk})^2} \quad (2)$$

де sim_{ij} – різниця між користувачами;
 x_i – перший користувач;
 x_j – другий користувач;
 m – кількість ознак (інтересів);
 x_{ik} – вага k -того фактора користувача x_i ;
 x_{jk} – вага k -того фактора користувача x_j .

Розкладання або факторизація матриці оцінок відображають користувачів та об'єкти в єдиний простір прихованих (латентних) факторів розмірності f в такий спосіб, що матриця взаємодії користувачів та об'єктів представляється як скалярний добуток в цьому просторі. Величина f визначається в процесі розкладання матриці і означає кількість факторів, за якими відбувається класифікація користувачів та об'єктів. Ідея полягає в тому, щоб з матриці оцінок $R(n \times m)$, де n – кількість користувачів, а m – кількість об'єктів, вирахувати матрицю (фактори \times користувачі) $P(f \times n)$ та матрицю (фактори \times об'єкти) $Q(f \times m)$ добуток яких буде давати матрицю $R(n \times m)$, значення якої будуть максимально близькими до значення оригінальної матриці R [4].

Кожен об'єкт представлений в цьому просторі як вектор $q_i \in \mathbb{R}^f$, а кожен користувач як вектор $p_u \in \mathbb{R}^f$. Для кожного i -того об'єкта координати вектора q_i виражають ступінь в якому об'єкт задовольняє тим чи іншим факторам. Для кожного u -того користувача координати вектора p_u виражають ступінь зацікавленості користувача в об'єктах, що мають яскраво виражений відповідний фактор. Скалярний добуток $q_i^T p_u$ виражає ступінь зацікавленості u -того користувача в i -тому об'єкті а також може використовуватися як прогнозована оцінка u -тим користувачем i -того об'єкта. Задачею факторизації матриці є пошук таких векторів q_i та p_u , скалярний добуток яких співпадає із наявними оцінками [4].

Сингулярний розклад матриці (SVD) це доволі популярний метод розкладу матриці, який реалізовано в багатьох рекомендаційних системах. Недоліком цього методу є те, що його традиційній формі він невизначений на розріджених даних, а його використання на таких даних може привести до перенавчання всієї моделі і як наслідок зменшення точності рекомендацій на реальних даних. Тому класичний сингулярний метод розкладу матриці практично не використовується у нових системах. Проте останні модифікації цього методу дозволяють проводити навчання на лише наявних оцінках навіть за умови їх розрідженості, зменшуючи перенавчання шляхом мінімізації регульованої квадратичної помилки [4]. Обчислення векторів факторів q_i та p_u відбувається шляхом вирішення наступної задачі оптимізації:

$$\min_{q, p} \sum_{(u,i) \in K} (r_{ui} - q_i^T p_u)^2 + \lambda (q_i^2 + p_u^2) \quad (3)$$

де u – індекс користувача;

i – індекс об'єкта;

K – множина пар (u,i) , для яких відомий r_{ui} , яку також називають навчальною множиною;

r_{ui} – реальна оцінка u -тим користувачем i -того об'єкта;

λ – константа, що контролює ступінь регуляризації і зазвичай визначається шляхом перехресної перевірки.

Метод стохастичного градієнтного спуску (SGD) є популярним підходом мінімізації рівняння (3) і поєднує простоту реалізації з відносно швидким часом роботи. Існує декілька варіацій цього методу, але далі буде наведено той, що описується в роботі [4]. До сильних сторін цього методу можна віднести гнучкість, можливість паралельних обчислень. До його слабких сторін можна віднести необхідність проходження великої кількості ітерацій для досягнення хороших результатів навчання.

Для кожного тренувального випадку система передбачає r_{ui} та обчислює відповідну помилку прогнозування e_{ui} за наступною формулою:

$$e_{ui} = r_{ui} - q_i^T p_u \quad (4)$$

Потім він змінює параметри на величину, пропорційну γ в протилежному напрямку градієнта, даючи результат:

$$q_i \leftarrow q_i + \gamma(e_{ui} p_u - \lambda q_i) \quad (5)$$

$$p_u \leftarrow p_u + \gamma(e_{ui} q_i - \lambda p_u) \quad (6)$$

де γ – коефіцієнт швидкості навчання;

λ – параметр регуляризації.

У вигляді псевдокоду алгоритм SGD можна описати як:

```
Initialize Q, V
repeat
  for random  $r_{ui} \in R$  do
     $e_{ui} = r_{ui} - q_i^T p_u$ 
     $q_i \leftarrow q_i + \gamma(e_{ui} p_u - \lambda q_i)$ 
     $p_u \leftarrow p_u + \gamma(e_{ui} q_i - \lambda p_u)$ 
  end for
until convergence
```

Варто звернути увагу на метод найменших квадратів ALS – метод мінімізації рівняння (3). До його сильних сторін можна віднести можливість розпаралелення обчислень, краща робота на розріджених даних. До його слабких сторін відносять більшу складність, в порівнянні з SGD.

Оскільки і q_i , і p_u невідомі, рівняння (3) не є опуклим. Однак, якщо зафіксувати одне з невідомих, завдання оптимізації стає квадратичним і може бути оптимально вирішено. Таким чином, в методі ALS чергуються фіксація q_i та фіксація p_u . Коли всі p_u зафіксовано, система перераховує q_i , вирішуючи задачу найменших квадратів, і навпаки. Це гарантує, що кожен крок зменшує рівняння (3) до співпадіння заданої міри подібності [4].

Помилка класичного методу найменших квадратів розраховується за формулою:

$$\sum_{(u,i) \in K} (r_{ui} - q_i^T p_u)^2 \quad (7)$$

Алгоритм ALS мовою псевдокоду матиме вигляд:

```
Initialize Q, V
repeat
  for u=1 to n do
```

$$p_u = \left(\sum_{i=1}^m q_i q_i^T + \lambda I_f \right)^{-1} \sum_{i=1}^m r_{ui} q_i$$

end for
for i=1 to m do

$$q_i = \left(\sum_{u=1}^n p_u p_u^T + \lambda I_f \right)^{-1} \sum_{u=1}^n r_{ui} p_u$$

end for
until convergence

Модифікацією методу найменших квадратів, що чергуються є метод зважених найменших квадратів, що чергуються (WALS). Цей метод відрізняється від класичного ALS тим, що він враховує не лише наявні оцінки. За рахунок системи вагових коефіцієнтів можливе тонке налаштування системи. Його помилка розраховується за формулою:

$$\sum_{(u,i) \in K} w_{i,j} (r_{ui} - q_i^T p_u)^2 + w_0 \sum_{(u,i) \notin K} (0 - q_i^T p_u)^2 \quad (8)$$

де $w_{i,j}$ – гіперпараметр, що регулює вагу визначених оцінок;
 w_0 – гіперпараметр, який регулює вагу відсутніх оцінок.

До популярних та простих методів розрахунку подібності належать косинус подібності та Евклідова відстань.

Методи розкладу матриці є зручним способом для передбачення оцінок від користувачів, але потребують навчального набору даних. Гнучкість методів розкладу матриці дозволяє виконувати тонкі налаштування системи шляхом введення в процес навчання вагових коефіцієнтів.

Отримані від алгоритму розкладу матриці вектори латентних факторів користувачів та об'єктів в подальшому можна використовувати не лише для прогнозування оцінок, але й для пошуку подібних користувачів та об'єктів або навіть для їх кластеризації. Приховані, або латентні фактори можуть бути корисним допоміжним інструментом в інтересоорієнтованій системі.

Перелік посилань

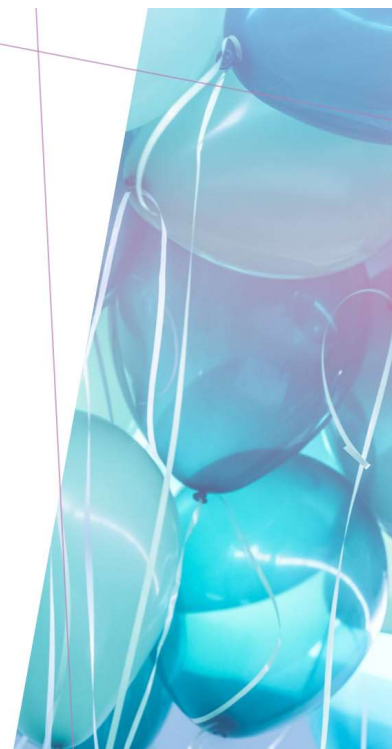
1. Luo J. THREE ESSAYS ON FRIEND RECOMMENDATION SYSTEMS FOR ONLINE SOCIAL NETWORKS / Jiaxi Luo. – Wisconsin-Milwaukee: The University of Wisconsin-Milwaukee, 2016. – 153 с.
2. Gorripati S. Community-Based Collaborative Filtering to Alleviate the Cold-Start and Sparsity Problems / S. Gorripati, V. Vatsavayi. // International Journal of Applied Engineering Research. – 2017. – Т. 12, вип. 15 – С. 5022–5030.
3. Adomavicius, G. and Tuzhilin, A., "Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions", IEEE Transactions on Knowledge and Data Engineering, vol. 17, no. 6, pp.734-749, June, 2005.
4. Y. Koren, R. Bell and C. Volinsky. "Matrix Factorization Techniques for Recommendation Systems", IEEE computer, 2009.

ДОДАТОК В (обов'язковий)

ПРЕЗЕНТАЦІЙНІ МАТЕРІАЛИ

ВИКОНАВ: ПРОТЯКОВСЬКИЙ АНДРІЙ ОЛЕКСІЙОВИЧ
КЕРІВНИК: ФОРКУН ЮРІЙ ВІКТОРОВИЧ
КАФЕДРА: ІНЖЕНЕРІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

ТЕХНОЛОГІЯ РОЗРОБКИ ПРОГРАМНОЇ СИСТЕМИ РЕКОМЕНДАЦІЙ ДЛЯ ЗНАЙОМСТВ ЛЮДЕЙ ТА ГРУП ЗА ІНТЕРЕСАМИ У ВІРТУАЛЬНИХ СПІЛЬНОТАХ



АКТУАЛЬНІСТЬ РОБОТИ

Системи рекомендацій друзів використовуються як у відомих соціальних мережах, як Facebook, так і у менших інтернет або мобільних додатках, наприклад Friendprobe. Рекомендаційні системи, що підбирають друзів у соціальних мережах, забезпечують користувачеві досвід спілкування всередині мережі. Чим краще такі системи підбирають друзів, тим кращий досвід переживає користувач і тим більше він задоволений.

Сьогодні активно відбувається пошук способів покращити якість рекомендацій таких систем. Перспективними є напрямки аналізу соціальних зв'язків (метод «друзі друзів»), врахування інтересів (явних з профілю користувача, або не явних, отриманих з активності користувача) та врахування контексту (геолокація, мова спілкування, тощо).

ОБ'ЄКТ, ПРЕДМЕТ ТА МЕТА ДОСЛІДЖЕННЯ

Об'єктом дослідження є технологія розробки програмної системи рекомендацій людей та груп за інтересами у віртуальних спільнотах.

Предметом дослідження є моделі та підходи до формування списків рекомендацій людей та груп за інтересами у віртуальних спільнотах.

Метою дослідження є створення технології розробки програмної системи рекомендацій для знайомств людей та груп за інтересами у віртуальних спільнотах шляхом використання та розвитку актуальних методів розрахунку рекомендацій людей.

НАУКОВА НОВИЗНА ТА ПРАКТИЧНЕ ЗНАЧЕННЯ

Наукова новизна:

- отримав подальший розвиток метод, на основі єдності, основна ідея якого полягає в поетапному вимірі міцності зв'язку в мережі і виявленні можливих зв'язків в цій мережі, які ще не створено; виявленні груп в мережі з використанням модулярності та рекомендації друзів, за рахунок включення фактору подібності інтересів та врахування двосторонньої сили зв'язку між користувачами, що дозволило покращити точність рекомендацій;
- набула подальшого розвитку формальна модель користувача сервісу розрахунку рекомендацій друзів, за рахунок її розширення профілями інтересів користувачів, що дало змогу покращити точність рекомендацій через точніше моделювання кожного окремого користувача;
- запропонована оригінальна технологія розробки програмної системи рекомендацій для знайомств людей та груп за інтересами у віртуальних спільнотах.

Практичне значення:

- використання запропонованого підходу всередині соціальних мереж для формування точніших рекомендацій людей;
- поліпшення якості розрахованих рекомендацій вдосконаленого методу відносно оригінального.

ПУБЛІКАЦІЇ

Протоковський А.О. Аналіз стратегій рекомендаційних систем знайомств людей та груп за інтересами у віртуальних спільнотах/ А.О. Протоковський, Ю.В. Форкун // Матеріали 9-ї міжнародної наукової конференції «Інформація, комунікація, суспільство 2020». – Львів: Видавництво Львівської політехніки.– 2020. – № 9. – С. 154-155.

Протоковський А. О. Методологія розрахунку рекомендацій в рекомендаційних системах / А. О. Протоковський, Ю. В. Форкун // Збірник наукових праць за матеріалами XII Всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних наук АПКН-2020». – 2020. – № 7. – С. 237–241.

АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ

В якості базового методу рекомендації друзів було вирішено обрати метод на основі узгодженості запропонований Shamsu Shehu в 2017 році, оскільки він враховує участь користувача у різних групах та соціальні зв'язки, що якнайкраще підходить для нашого завдання. На нашу думку цей метод має такі недоліки:

- не враховує інтереси користувачів;
- враховує лише односторонню міцність зв'язку в процесі розрахунку можливості встановлення дружби;
- не враховує дружбу користувача з сильно відмінними від нього користувачами.

ПОСТАНОВКА ЗАДАЧІ

- провести аналіз методів та інструментів для рекомендації людей;
- розробити метод розрахунку рекомендацій людей, опираючись на актуальні методи, шляхом перегляду списку факторів, які враховуються та перерахунком їх ваг у системі;
- створити технологію розробки програмної системи рекомендацій для знайомств людей та груп за інтересами у віртуальних спільнотах;
- провести практичну перевірку ефективності запропонованих нововведень в порівнянні з оригінальним методом;
- провести практичну перевірку ефективності запропонованого методу рекомендацій спільнот.

ПРОПОНОВАНІ ПОКРАЩЕННЯ

Розрахунок сили зв'язку у базовому підході

$$R(T, T_1) = \lambda_1 F_1(T, T_1) + \lambda_2 F_2(T, T_1) + \lambda_3 F_3(T, T_1)$$

$$F_1 = \frac{|Fr(T) \cap Fr(T_1)|}{|Fr(T)|}$$

$$F_2 = \frac{|Gr(T) \cap Gr(T_1)|}{|Gr(T)|}$$

$$F_3 = \frac{|Apps(T) \cap Apps(T_1)|}{|Apps(T)|}$$

Розрахунок сили зв'язку у модифікованому підході

$$R'(T, T_1) = \lambda_1 F'_1(T, T_1) + \lambda_2 F_2(T, T_1) + \lambda_3 F'_3(T, T_1)$$

$$cossim_{A,B} = \frac{\sum_{k=1}^n a_k b_k}{\sqrt{\sum_{k=1}^n a_k^2} \sqrt{\sum_{k=1}^n b_k^2}},$$

$$euclidsim_{A,B} = A - B = \sqrt{\sum_{k=1}^n (a_k - b_k)^2},$$

$$F'_3(T, T_1) = cossim_{T, T_1} \times euclidsim_{T, T_1}$$

ПРОПОНОВАНІ ПОКРАЩЕННЯ

Розрахунок можливості зв'язку та порогового значення у базовому підході

$$\frac{R(T, T_i) + R(T_i, T)}{2} > threshold$$

$$threshold = \frac{\sum_{i \in Fr(T)} R(T, T_i)}{|Fr(T)|}$$

Розрахунок можливості зв'язку та порогового значення у модифікованому підході

$$\frac{R(T, T_i) + R(T_i, T)}{2} > \text{Min}(threshold_T, threshold_{T_i})$$

$$threshold = \gamma \frac{\sum_{i \in Fr(T)} R(T, T_i)}{|Fr(T)|} + (1 - \gamma) \frac{\sum_{j \in \text{Min}R_n(T)} R(T, T_j)}{|\text{Min}R_n(T)|}$$

ВИСНОВКИ

- проведено аналіз методів та інструментів для рекомендації людей;
- розроблено метод розрахунку рекомендацій людей, опираючись на актуальні методи, шляхом перегляду списку факторів, які враховуються та перерахунком їх ваг у системі;
- створено технологію розробки програмної системи рекомендацій для знайомств людей та груп за інтересами у віртуальних спільнотах;
- проведено практичну перевірку ефективності запропонованих нововведень в порівнянні з оригінальним методом;
- проведено практичну перевірку ефективності запропонованого методу рекомендацій спільнот.



ДЯКУЮ ЗА УВАГУ

Anti-Plagiarism v-15.257

Максимальное совпадение с одним документом 0.0%

Словари проверки: en_US, ru_RU, ua_UA. Ошибок в документах: 10%

ID: 82422 Название: Технологія розробки програмної системи рекомендацій для знайомств людей та груп за інтересами у віртуальних спільнотах Добавлено в БД: 2020-12-04 Авторы: А. О. Протоковський Руководители: Ю. В. Форкун Консультанты: Оponentы:	Документ		Суммарное совпадение по Базе Данных	
	Символы	Лексемы	Символы	Лексемы
	126955	954	1556 (1%)	30 (3%)

Источник плагиата

ID	Описание	Наличие плагиата в документе	
		Символы	Лексемы



Ім'я користувача:
Kafedra IPZ

Дата перевірки:
04.12.2020 12:54:55 EET

Дата звіту:
04.12.2020 12:56:57 EET

ID перевірки:
1005362953

Тип перевірки:
Doc vs Internet + Library

ID користувача:
100005589

Назва документа: ПЗ-2.imp

Кількість сторінок: 114 Кількість слів: 18805 Кількість символів: 144897 Розмір файлу: 4.59 MB ID файлу: 1005655523

2.69%
Схожість

Найбільша схожість: 0.44% з Інтернет-джерелом (<https://scholar.google.com/citations?user=CyrE5ycAAAAJ&hl=en>)

2.44% Джерела з Інтернету

363

Сторінка 116

0.25% Джерела з Бібліотеки

31

Сторінка 118

0% Цитат

Вилучення цитат вимкнено

Вилучення списку бібліографічних посилань вимкнено

0%
Вилучень

Немає вилучених джерел

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

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

Назва: Технологія розробки програмної системи рекомендацій для знайомств людей та груп за інтересами у віртуальних спільнотах

Автор: А. О. Протоковський

Спеціальність: 121 Інженерія програмного забезпечення

Освітня програма: Інженерія програмного забезпечення

Науковий керівник: к.т.н., доцент, Форкун Ю.В.

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 (далі – зазначаються детальні та аргументовані підстави віднесення запозичень до правомірних). 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: Текст є оригінальним, виявлені запозичення не є плагіатом оскільки розміщені в розділах які не описують безпосередньо авторське дослідження, складають 2.69% та мають посилання на літературні джерела. Робота приймається до захисту.

4.11.2020

Підпис керівника

Підпис завідувача кафедри

Підпис гаранта ОП

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА ДИПЛОМНУ РОБОТУ

Дипломник студент групи ІПЗм-19-1 Протоковський Андрій Олексійович

Тема Технологія розробки програмної системи рекомендацій для знайомств людей та груп за інтересами у віртуальних спільнотах

Спеціальність 121 – Інженерія програмного забезпечення

Обсяг дипломної роботи:

Кількість листів креслень 0; кількість сторінок записки 96

1. Короткий зміст ДР та прийнятих рішень Представлена робота присвячена актуальній темі в області рекомендаційних систем. і складається з наступних розділів: вступ, аналіз сучасного стану рекомендаційних систем, моделі та методи технології розрахунку рекомендацій людей та груп, алгоритми та технологія для розрахунку рекомендацій людей та груп, реалізація програмного забезпечення для розрахунку рекомендацій людей та груп, висновки, додатки.

2. Висновок про відповідність ДР поставленому завданню Подана дипломна робота виконана у відповідності до завдання, у повному обсязі та у встановлений термін

3. Характеристика виконання кожного розділу роботи, ступінь використання останніх досягнень науки і техніки і передових методів роботи: В першому розділі роботи проведено детальний аналіз предметної області, досліджено існуючі технології рекомендаційних систем, проаналізовано особливості існуючих рішень, запропоновано способи покращення конкретного методу вирішення задачі та обґрунтовано актуальність роботи. В другому розділі було детально описано метод рекомендації людей, що вдосконалюється та детально описано запропоновані покращення. В третьому розділі було розроблено алгоритми рекомендацій людей та груп, розроблено вимоги до програмного забезпечення та виконано його проектування. В четвертому розділі було виконано програмну реалізацію розроблено системи та проведено її тестування на тестовому наборі даних, а також зроблено висновки про позитивний вплив запропонованих змін на точність рекомендацій.

4. Позитивні сторони роботи Серед позитивних сторін роботи варто виділити вибір актуальної теми дослідження, проведений глибокий аналіз технологій систем рекомендацій людей та груп, а також проведення тестування запропонованого рішення на двох різних тестових вибірках різного розміру.

5. Негативні сторони роботи На момент рецензування були виявлені незначні помилки у оформленні матеріалу, які були оперативно усунуті.

6. Оцінка графічного оформлення та пояснювальної записки роботи Надані на рецензування матеріали роботи чітко структуровані у відповідності із послідовністю виконання роботи. Виявлені у ході рецензування незначні стилістичні та орфографічні помилки були оперативно усунуті. Графічне оформлення та пояснювальна записка роботи заслуговують оцінки «добре»

7. Відгук про роботу в цілому Через зміст роботи в повній мірі розкривається обрана тема. Запропоновані покращення в достатній мірі аргументовані на теоретичному та практичному рівнях. Система із запропонованими покращеннями була протестована на підготованих наборах даних та порівняна із незміненою системою. Результатом проведення тестування стали відповідні висновки щодо впливу запропонованих покращень на роботу системи в цілому.

8. Інші зауваження _____

9. Оцінка дипломної роботи Робота заслуговує оцінки «добре», а її автор присвоєння кваліфікації «магістра» інженерії програмного забезпечення

РЕЦЕНЗЕНТ (прізвище, ім'я, по-батькові, посада, місце роботи) Говорушенко Тетяна Олександрівна, доктор технічних наук, професор, зав. кафедри комп'ютерної інженерії та системного програмування ХНУ

“02” грудня

2020 р.

(підпис)