МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет
імені Петра Могили
Факультет комп'ютерних наук
Кафедра інтелектуальних інформаційних систем

ДОПУЩЕНО ДО ЗАХИСТУ
Завідувач кафедри інтелектуальних
інформаційних систем, д-р техн. наук, проф.
_____Ю. П. Кондратенко
«____»_____2022 р.

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

ІНТЕЛЕКТУАЛЬНА СИСТЕМА ДЛЯ ВИЗНАЧЕННЯ ЛОКАЦІЙ МЕТОДОМ МАШИННОГО НАВЧАННЯ

Спеціальність 122 «Комп'ютерні науки»

122 – МКР – 601.2180304

Студент _____ Є. І. Скакун

«__» _____ 2022 р.

Консультант _____ О. П. Гожий
д-р техн. наук, проф. кафедри ІІС

«__» _____ 2022 р.

Миколаїв – 2022

Чорноморський національний університет ім. Петра Могили
Факультет комп'ютерних наук
Кафедра інтелектуальних інформаційних систем

Освітньо-кваліфікаційний рівень **магістр**
Галузь знань **12 «Інформаційні технології»**
(шифр і назва)
Спеціальність **122 «Комп'ютерні науки»**
(шифр і назва)

ЗАТВЕРДЖУЮ
Завідувач кафедри інтелектуальних інформаційних систем, д-р техн. наук, проф.
_____ Ю. П. Кондратенко
«____» _____2022 р.

ЗАВДАННЯ
на магістерську кваліфікаційну роботу
_____ Скакуну Євгенію Ігоровичу _____
(прізвище, ім'я, по батькові)

1. Тема магістерської кваліфікаційної роботи
Інтелектуальна система для визначення локацій методом машинного навчання

Керівник роботи Гожий Олександр Петрович д-р техн. наук, проф. кафедри ІІС
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
Затв. наказом Ректора ЧНУ ім. Петра Могили від «___» ____ 20__ р. № _____

2. Строк подання студентом роботи «___» ___ 2022 р.

3. Вхідні (початкові) дані до роботи: Документ користувача, документ локації, рейтинги локацій, історія користувача, інтереси користувача.

Очікуваний результат роботи: Список рекомендованих локацій для кожного користувача.

4. Зміст пояснювальної записки (перелік питань, які потрібно розглянути):
Аналіз предметної сфери, об`єкту та предмету дослідження, огляд наявних результатів. Моделювання та технічне проектування. Розробка програмного

забезпечення та перевірка працездатності інтелектуальної системи визначення локацій методом машинного навчання. Спеціальна методична частина. Спеціальна частина з охорони праці. Висновки.

5. Перелік графічних матеріалів: Графічний матеріал наведений, магістерська робота містить 72 сторінок (без додатків), 22 рис., 4 табл., 2 додатки та 44 джерел посилання.

6. Завдання до спеціальної частини: Аналіз умов праці та забезпечення безпеки при надзвичайних ситуаціях персоналу ТОВ «Горизонт».

7. Консультанти:

Розділ	**Прізвище, ініціали та посада консультанта**	**Підпис**
Методична частина	Гожий О. П. д-р техн. наук, проф.	
Частина з охорони праці	Щербак Ю. Г. канд. техн. наук, доцент	

Керівник роботи _____д-р техн. наук, проф. Гожий О. П._____
(наук. ступінь, вчене звання, прізвище та ініціали)

_____
(підпис)

Завдання прийнято до виконання _____Скакун Є.І._____
(прізвище та ініціали)

_____
(підпис)

Дата видачі завдання «____» _____ 2022 р.

КАЛЕНДАРНИЙ ПЛАН
Виконання магістерської кваліфікаційної роботи

Тема: «Інтелектуальна система для визначення локацій методом машинного навчання»

№	Найменування роботи	Початок	Закінчення	Примітки
1	Визначення керівника і теми МКР. Подання заяви на затвердження теми МКР	01.09.2021	10.10.2021	Виконано
2	Отримання завдання на виконання МКР	19.10.2021	22.10.2021	Виконано
3	Складання календарного плану на період виконання МКР	23.10.2021	26.10.2021	Виконано
4	Огляд літератури за темою дослідження	27.10.2021	10.11.2021	Виконано
5	Проходження переддипломної практики, збір та аналіз матеріалів до МКР	22.11.2021	11.12.2021	Виконано
6	Аналіз предметної області та розробка технічного завдання. Моделювання результатів	16.12.2021	12.01.2022	Виконано
7	Опис фахової частини МКР, зокрема дослідження публікацій щодо систем рекомендацій, огляд існуючих систем, реалізація системи рекомендації з аналізом отриманих результатів	13.01.2022	25.01.2022	Виконано
8	Розробка спеціальної частини з охорони праці та методичної частини	26.01.2022	30.01.2022	Виконано
9	Попередній захист МКР на засіданні комісії кафедри	31.01.2022	31.01.2022	Виконано
10	Корегування роботи за результатами попереднього захисту	01.02.2022	03.02.2022	Виконано
11	Остаточне оформлення пояснювальної записки та слайдів доповіді для захисту	04.02.2022	06.02.2022	Виконано
12	Подання МКР рецензенту	09.02.2022	10.02.2022	Виконано
13	Рецензування МКР	11.02.2022	12.02.2022	
14	Подання МКР, її електронної копії та інших документів (відгуку, рецензії) до захисту	14.02.2022	15.02.2022	
15	Захист МКР перед екзаменаційною комісією (ЕК)	21.02.2022	22.02.2022	

Розробив студент Скакун Є.І. _____
(прізвище та ініціали) (*підпис*)

Керівник роботи д-р техн. наук, проф. Гожий О. П. _____
(наук. ступінь, вчене звання, прізвище та ініціали) *(підпис)*

«____» _____ 2022 р.

АНОТАЦІЯ

до магістерської кваліфікаційної роботи
«Інтелектуальна система для визначення локацій методом машинного навчання»
Студент: Скакун Євгеній Ігорович
Керівник: д-р техн. наук, проф. Гожий О. П.

Робота присвячена дослідженню та розробці інтелектуальної системи для визначення локацій методом машинного навчання.

Об'єкт дослідження – процес функціонування інтелектуальної системи за допомогою методів машинного навчання.

Предмет дослідження – інтелектуальна система для підбору локацій за допомогою методів машинного навчання.

Метою роботи є створення швидкої та ефективної інтелектуальної системи для підбору рекомендованих локацій за допомогою методів машинного навчання на базі існуючого мобільного застосунку.

Практична значимість розробленої інтелектуальної системи полягає в тому, щоб покращити досвід користувачів в пошуку локацій, які можуть їх зацікавити. Добре продумана система рекомендацій звільняє користувача від фільтрації великої кількості даних, що в свою чергу дозволяє користувачеві отримувати якісний контент за короткий проміжок часу.

МКР складається з фахового розділу, спеціалтної методичної частини та спеціальної частини з охорони праці та безпеки у надзвичайних ситуаціях.

Пояснювальна записка до фахової частини роботи складається із вступу, трьох розділів, висновків, переліку джерел посилання та двох додатків.

У вступі визначається актуальність теми та проводиться короткий огляд поставленої задачі.

У першому розділі проводиться аналіз предметної області, об`єкту та предмету дослідження. Огляд та аналіз уже існуючих аналогів ІС.

Другий розділ присвячено моделюванню та технічному проектуванню.

У третьому розділі наведений опис процесу розробки ІС та перевірка працездатності системи.

В спеціальній методичній частині підготовано 2 практичні роботи по темі ДР.

В спеціальній частині з охорони праці та безпеки у надзвичайних ситуаціях проведено аналіз умов праці та забезпечення безпеки при надзвичайних ситуаціях персоналу ТОВ «Горизонт».

У висновках проводиться аналіз проведеної роботи та отриманих результатів.

В цілому магістерська робота містить 72 сторінок (без додатків), 22 рис., 4 табл., 2 додатки та 44 джерел посилання.

Ключові слова: *KNN, рекомендація, локація, інтелектуальна система, матрична факторизація, алгоритм.*

ABSTRACT

of the Master's Thesis

"Intelligent system for determining locations by machine learning"

Undergraduate: **Skakun Yevhenii Ihorovych**

Supervisor of thesis: D.Sc., professor **Gozhyj A. P.**

The work is devoted to the research and development of an intelligent system for determining locations by machine learning.

The object of study - the process of functioning of the intelligent system using machine learning methods.

The subject of research is an intelligent system for selection of locations using machine learning methods.

The practical significance of the developed intelligent system is to improve the user experience in finding locations that may interest them. A well-designed system of recommendations frees the user from filtering large amounts of data, which in turn allows the user to receive quality content in a short period of time.

Thesis consists of a professional section, a special methodological part and a special part on labor protection and safety in emergencies.

The explanatory note to the professional part of the thesis consists of an introduction, three sections, conclusions, a list of reference sources and two appendices.

The introduction determines the relevance of the topic and provides a brief overview of the task.

In the first section the analysis of subject area, object and subject of research is carried out. Review and analysis of existing IS analogues.

The second section is devoted to modeling and technical design.

The third section describes the process of IS development and verification of the system.

In a special methodical part 2 practical works on the topic of MW were prepared.

In the special part on labor protection and safety in emergency situations the analysis of working conditions and safety in case of emergencies of the personnel of LLC "Horizon" is carried out.

The conclusions analyze the work done and the results obtained.

In total, the master's thesis contains 72 pages (without appendices), 22 figures, 4 tables, 2 appendices and 44 references.

Keywords: *KNN, recommendation, location, intelligent system, matrix factorization, algorithm.*

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ І СКОРОЧЕНЬ 4
ВСТУП 6
1 АНАЛІЗ ПРЕДМЕТНОЇ СФЕРИ. ПОСТАНОВКА ЗАДАЧІ 8
1.1 Опис предметної сфери 8
1.1.1 Опис процесу діяльності 8
1.1.2 Загальна функціональна модель (схема) 9
1.1.3 Загальні підходи фільтрації 10
1.2 Огляд та аналіз наявних аналогів та публікацій 11
1.2.1 Аналоги 11
1.2.2 Публікації 13
1.3 Постановка задачі 17
Висновки до розділу 1 18
2 МАТЕМАТИЧНІ МОДЕЛІ, МЕТОДИ, ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ДЛЯ ВИРІШЕННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ 19
2.1 Загальні поняття 19
2.1.1 Типи рекомендаційних систем 19
2.1.2 Проблеми спільної фільтрації 22
2.1.3 Baseline 23
2.2 Алгоритми прогнозування 26
2.2.1 Найпростіші алгоритми 26
2.2.2 Алгоритм найближчих сусідів 27
2.2.3 Алгоритми матричної факторізації 30
2.2.4 Інші алгоритми 36
2.3 Оцінка подібності 36
2.4 Оцінка точності прогнозної моделі та прогнозів 40
Висновки до розділу 2 45
3 МОДЕЛЮВАННЯ ТА ДОСЛІДЖЕННЯ ОТРИМАНИХ РЕЗУЛЬТАТІВ 46
3.1 Опис даних прогнозування 46

3.2 Опис інтелектуальної системи 51
3.2.1 Блок попереднього розрахунку рейтингів 52
3.2.2 Блок розрахунку рекомендацій 54
3.2.3 Формування рекомендації 56
3.3 Аналіз результатів дослідження 59
Висновки до розділу 3 64
4 МЕТОДИЧНА ЧАСТИНА 67
5 АНАЛІЗ УМОВ ПРАЦІ ТА ЗАБЕЗПЕЧЕННЯ БЕЗПЕКИ ПРИ НАДЗВИЧАЙНИХ СИТУАЦІЯХ ПЕРСОНАЛУ ТОВ «ГОРИЗОНТ» 77
ВИСНОВКИ 88
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 90
ДОДАТОК А Код програмного забезпечення 95
ДОДАТОК Б Матеріали апробації роботи 98

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ І СКОРОЧЕНЬ

БД – База даних

ОПМ – Обробка природної мови

ALS – Alternating Least Squares

BPMF – Bayesian Probabilistic Matrix Factorization

CF – Collaborative Filtering

EM – Expectation-maximization

FCP – Fraction of Concordant Pairs

IBCF – Item Based Collaborative Filtering

MAE – Mean Absolute Error

MF – Matrix Factorization

MSD – Mean Squared Difference

MSE – Mean Square Error

NMF – Non-Negative Matrix Factorization

PMF – Positive Matrix Factorisation

RMSE – Root Mean Square Error

SVD – Singular Value Decomposition

UBCF – User Based Collaborative Filtering

Пояснювальна записка

до магістерської кваліфікаційної роботи

на тему:

«ІНТЕЛЕКТУАЛЬНА СИСТЕМА ДЛЯ ВИЗНАЧЕННЯ ЛОКАЦІЙ МЕТОДОМ МАШИННОГО НАВЧАННЯ »

Спеціальність 122 «Комп’ютерні науки»

122 – МКР – 601.2180304

Студент _____ Є. І. Скакун

«__» _____ 2022 р.

Консультант _____ О. П. Гожий
д-р техн. наук, проф. кафедри ІІС

«__» _____ 2022 р.

Миколаїв – 2022

ВСТУП

Зі збільшенням кількості локації в системі на користувача збільшується інформаційний тиск, що не дуже гарно сприяє на користувацький досвід та відгуки про систему. Для уникнення подібних ситуацій є безліч варіантів вирішення проблеми. Найпростіший шлях – запропонувати йому додаткові локації, але якщо радити все поспіль, є ризик втратити відвідувача через нав'язливість. Однак, якщо не запропонувати нічого, то менші шанси зацікавити користувача. Знайти "золоту середину" допомагають рекомендаційні системи.

Системи рекомендацій допомагають користувачам отримувати персоналізовані рекомендації, допомагають користувачам приймати правильні рішення під час серфінгу, підвищувати продажі та перевизначати досвід перегляду локацій, утримувати користувачів, покращувати їхній досвід бронювання. Проблема перевантаження інформацією вирішується пошуковими системами, але вони не забезпечують персоналізацію даних. Механізми рекомендацій забезпечують персоналізацію. Існують різні типи рекомендаційних систем, таких як система рекомендацій на основі вмісту, спільна фільтрація, гібридна система рекомендацій, демографічна система та система рекомендацій на основі ключових слів.

Тому було запропоновано створити інтелектуальну систему для підбору рекомендованих локацій за допомогою методів машинного навчання, що значно поліпшить процес пошуку користувачами цікавих локацій для відпочинку.

Метою роботи є створення швидкої та ефективної інтелектуальної системи для підбору рекомендованих локацій за допомогою методів машинного навчання на базі існуючого мобільного застосунку.

Об’єктом дослідження є процес функціонування інтелектуальної системи за допомогою методів машинного навчання.

Предметом дослідження є інтелектуальна система для підбору локацій за допомогою методів машинного навчання.

Для досягнення зазначеної мети необхідно виконати наступні завдання:

– проаналізувати схожі програмні продукти для підбору рекомендованих локацій;

– розробити інтелектуальне та програмне забезпечення системи для збору та підготовки даних;

– розробити інтелектуальну систему для підбору рекомендованих локацій;

– протестувати точність системи.

Практична значимість розробленої інтелектуальної системи полягає в тому, щоб покращити досвід користувачів в пошуку локацій, які можуть їх зацікавити. Більшість користувачів знають про існування індивідуальних рекомендацій на великих інтернет-платформах, однак важливість і домінування цієї технології, можливо, більш важливі, ніж багато хто думає. Оскільки добре продумана система рекомендацій звільняє користувача від фільтрації великої кількості даних, що в свою чергу дозволяє користувачеві отримувати якісний контент за короткий проміжок часу.

1 АНАЛІЗ ПРЕДМЕТНОЇ СФЕРИ. ПОСТАНОВКА ЗАДАЧІ

1.1 Опис предметної сфери

1.1.1 Опис процесу діяльності

Рекомендаційні системи – це спеціальні алгоритми, які пропонують користувачеві місця, що підходять йому за тими чи іншими критеріями. Рекомендації допомагають кристувачу, по-перше, розібратися, що йому потрібно, а по-друге, швидше ухвалити рішення про вибір цікавого місця. В результаті його лояльність підвищується з великою ймовірністю він повернеться на ресурс за новими враженнями.

А ось ще один приклад: якщо користувач сумніваєтеся в виборі того чи іншого місця, то, швидше за все, перегляне блок «Вам також може сподобатися» або «Подібні локації». З усього цього можна сміливо зробити висновок про те, що користувачі люблять отримувати рекомендації, що ґрунтуються на їхніх інтересах та потребах, тому що це скорочує час на пошуки потрібної локації на ресурсі та полегшує процес бронювання за рахунок спочатку релевантних та персоналізованих пропозицій. Але дати такі рекомендації – це не просто показати кілька подібних локацій. Важливо формувати пул місць розумно, ґрунтуючись на зібраних користувачів даних і враховуючи максимальну кількість факторів таких як: ціна, популярність, місність, аналоги і т. д., щоб підвищити ймовірність бронювання.

На сьогодні є актуальна тема рекомендаційних систем. І було вирішено створити інтелектуальну систему, яка допоможе швидко і зручно підбирати локації для користувачі. Перед тим, як приступити до розробки такої інформаційної – системи було проаналізовані аналоги та публікації інтелектуальної системи, які детально розглянуті в першому розділі другого підрозділу.

1.1.2 Загальна функціональна модель (схема)

Система рекомендацій генерує скомпільований список елементів, які можуть бути зацікавлені користувачем у взаємності їх поточного вибору елемента(ів). Рекомендації розширює пропозиції користувачів без будь-яких перешкод або монотонності, і він не рекомендує елементи, які користувач вже знає.

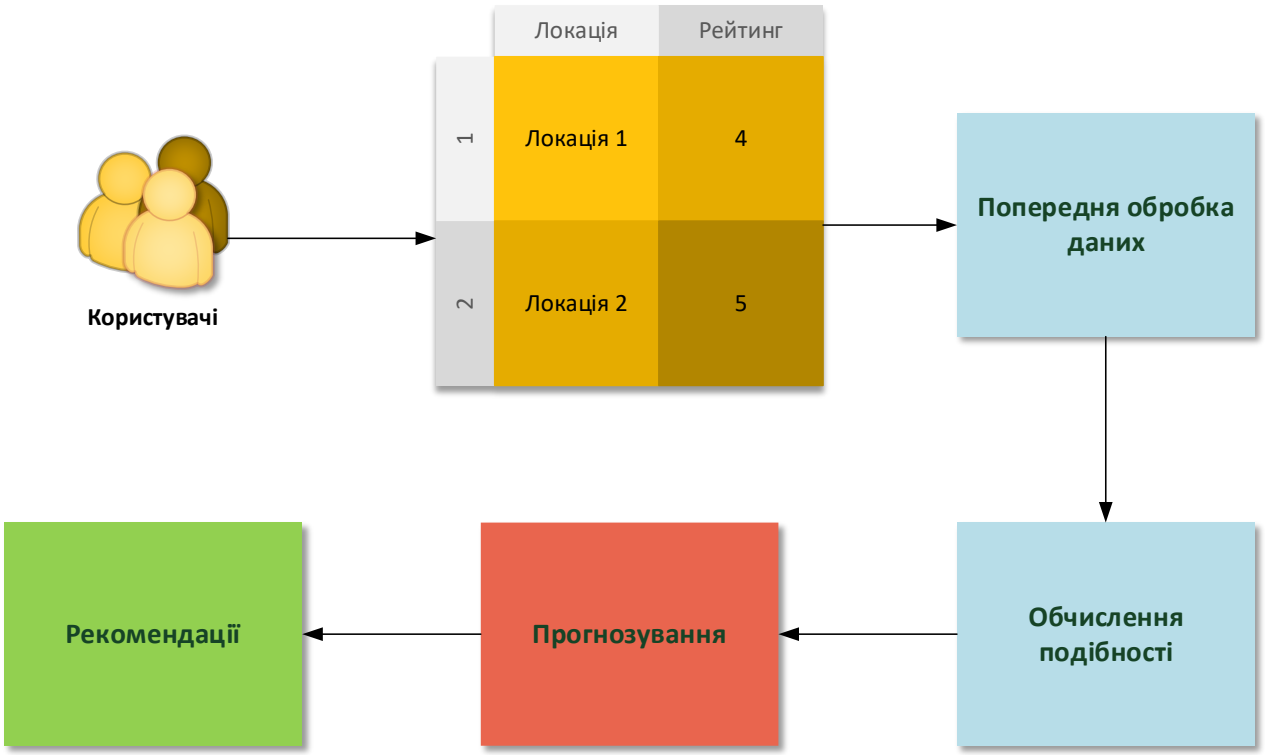

Рис. 1.1. Схема системи рекомендацій

Робочий процес системи рекомендацій показаний на діаграмі вище (рис. 1.1.), показує співпрацю користувача щодо рейтингів різних локацій. Нові користувачі отримують свої рекомендації на основі рекомендацій існуючих користувачів або на базі їхнього профайлу.

Системи рекомендацій — це системи на основі машинного навчання, які сканують усі можливі варіанти та надають передбачення чи рекомендації. Однак створення рекомендаційної системи має такі складності:

1) обсяг даних великий і включає значний список локацій, історії, профілів та інтересів клієнтів, рейтингів та інших точок даних;

2) нові зареєстровані клієнти мають дуже обмежену інформацію;
3) прогнозування в режимі реального часу для користувачів;
4) старі користувачі можуть мати надлишок інформації;
5) не слід показувати предмети, які дуже відрізняються чи надто схожі;
6) дані користувачів є взаємозамінними;
7) користувачі можуть різко змінити свої смаки, інтереси.

1.1.3 Загальні підходи фільтрації

Collaborative filtering – є, мабуть, найвідомішим підходом до рекомендацій до такої міри, що іноді його вважають синонімом поля. Основна ідея полягає в тому, що користувачі отримують матрицю уподобань для елементів, які використовуються для прогнозування відсутніх уподобань і рекомендації елементів з високим прогнозом. Все, що вам потрібно для початку – це ідентифікатори користувачів і елементів, а також уявлення про переваги користувачів щодо елементів (оцінки, перегляди тощо).

Content-based filtering – на основі вмісту надають користувацькі переваги для елементів і рекомендують подібні елементи на основі уявлення про вміст елемента, специфічного для домену. Цей підхід також природно поширюється на випадки, коли доступні метадані елементів.

Social and demographic – ці рекомендації пропонують товари, які подобаються друзям, друзям друзів та демографічно схожим людям. Такі рекомендації не потребують будь-яких уподобань користувача, якому даються рекомендації, що робить їх дуже потужними.

Contextual - алгоритми контекстної рекомендації рекомендують елементи, які відповідають поточному контексту користувача. Це дозволяє їм бути більш гнучкими та адаптивними до поточних потреб користувача, ніж методи, які ігнорують контекст. Отже, контекстні алгоритми з більшою ймовірністю викликають відповідь, ніж підходи, які базуються лише на історичних даних.

1.2 Огляд та аналіз наявних аналогів та публікацій

1.2.1 Аналоги

Рекомендаційні системи використовуються в різних областях і найчастіше вважаються генераторами списків відтворення для відео та музичних сервісів, таких як Netflix, YouTube та Spotify, рекомендації щодо продуктів для таких сервісів, як Amazon, або рекомендації щодо утримання для платформ соціальних мереж, таких як Facebook та Twitter. Ці системи можуть працювати, використовуючи один вхід, наприклад музику або кілька входів всередині та між платформами, такими як новини, книги та пошукові запити.

Netflix – американський провайдер медійних послуг та продюсерська компанія зі штаб-квартирою в Лос-Гатос, Каліфорнія [1].

Система реєструє, що відбувається, коли користувач заходить в сервіс Netflix, і постійно перенавчає алгоритми відповідно до цих нових даних, щоб більш точно прогнозувати, що може сподобатися користувачеві. Бази даних, алгоритми та обчислювальні системи сервісу щільно пов'язані один з одним.

На рис. 1.2. представлена фотографія головної сторінки «Netflix» [1].

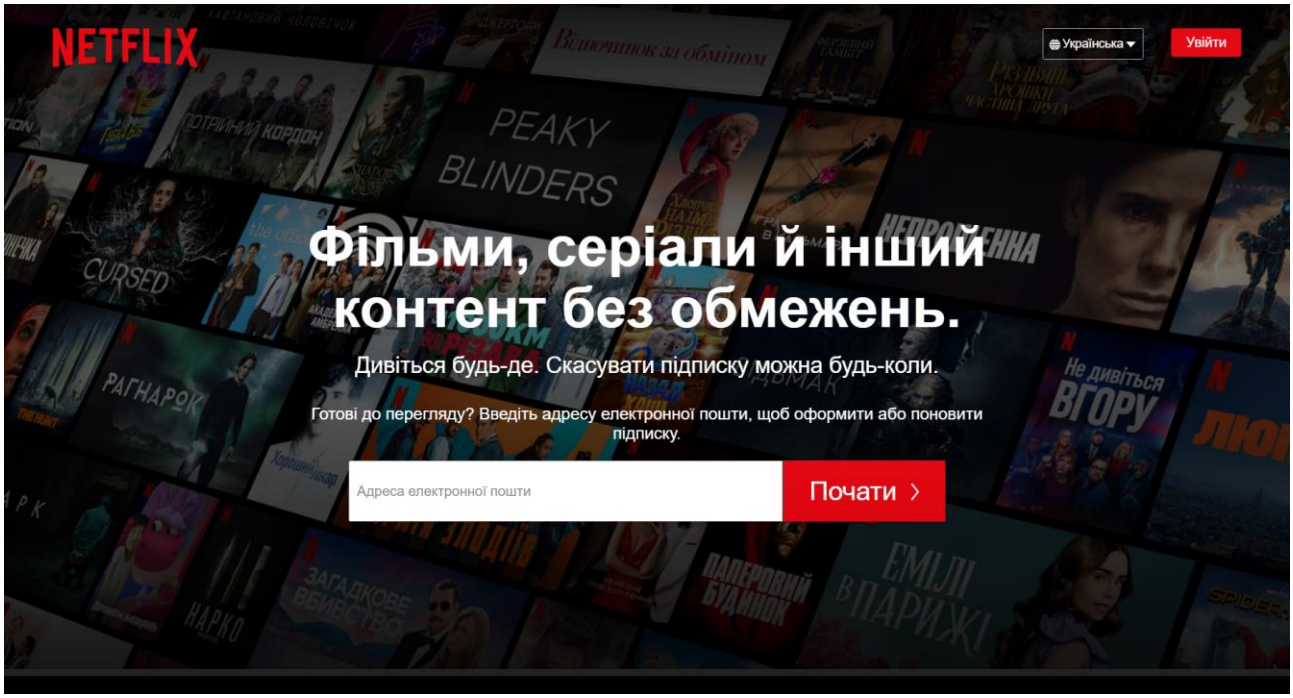

Рис. 1.2. Головна сторінка «Netflix» [1]

YouTube – популярний відеохостинг, що надає послуги розміщення відеоматеріалів. Заснований 14 лютого 2005 року трьома працівниками PayPal: Чадом Герлі, Стівеном Чені та Джаведом Карімом. YouTube є підрозділом компанії Google.

Рекомендації YouTube – це ролики, які Ютуб підбирає на основі ваших інтересів та пропонує до перегляду. Вони схожі на ті, що ви дивилися раніше або шукали в пошуку. Попадання в рекомендації дає власникам каналів приплив додаткового трафіку, причому безкоштовного, адже їхнє відео просуває сам відеохостинг.

На рис. 1.3. представлена фотографія головної сторінки «YouTube» [2].

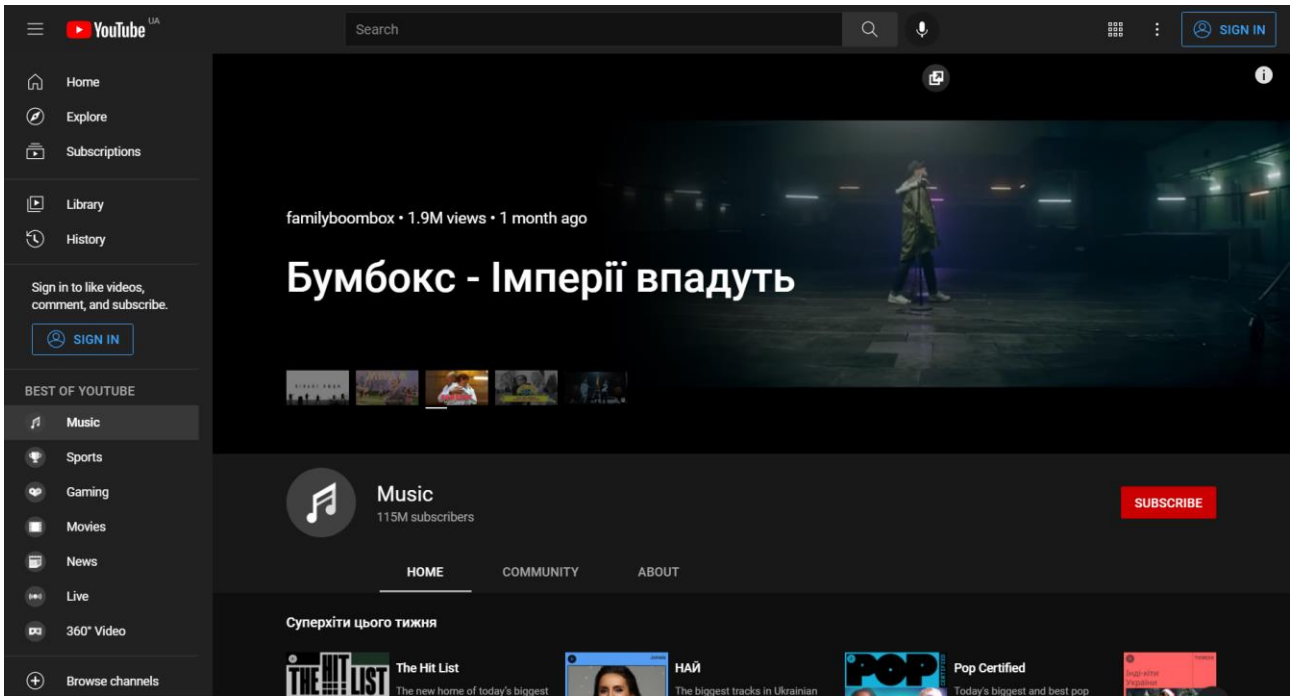

Рис. 1.3. Головна сторінка «YouTube» [2]

Spotify – інтернет-сервіс потокового аудіо, що дозволяє легально й безкоштовно прослуховувати музичні композиції. Надає послуги легального онлайнового стримінгу аудіозаписів основних світових і незалежних лейблів, в тому числі BBC, Sony, EMI, Warner Music Group та Universal.

Інтернет-сервіс поєднала кращі стратегії рекомендацій інших сервісів. Вийшов унікальний та потужний дослідницький механізм.

Для створення Discovery Weekly сервіс використовує три моделі:

1) колаборативна фільтрація (метод Last.fm), яка аналізує вашу модель поведінки та інших;

2) обробка природної мови (ОПМ) для аналізу тексту;

3) аудіомоделі, які аналізують аудіофайли.

На рис. 1.4. представлена фотографія головної сторінки «Spotify» [3].

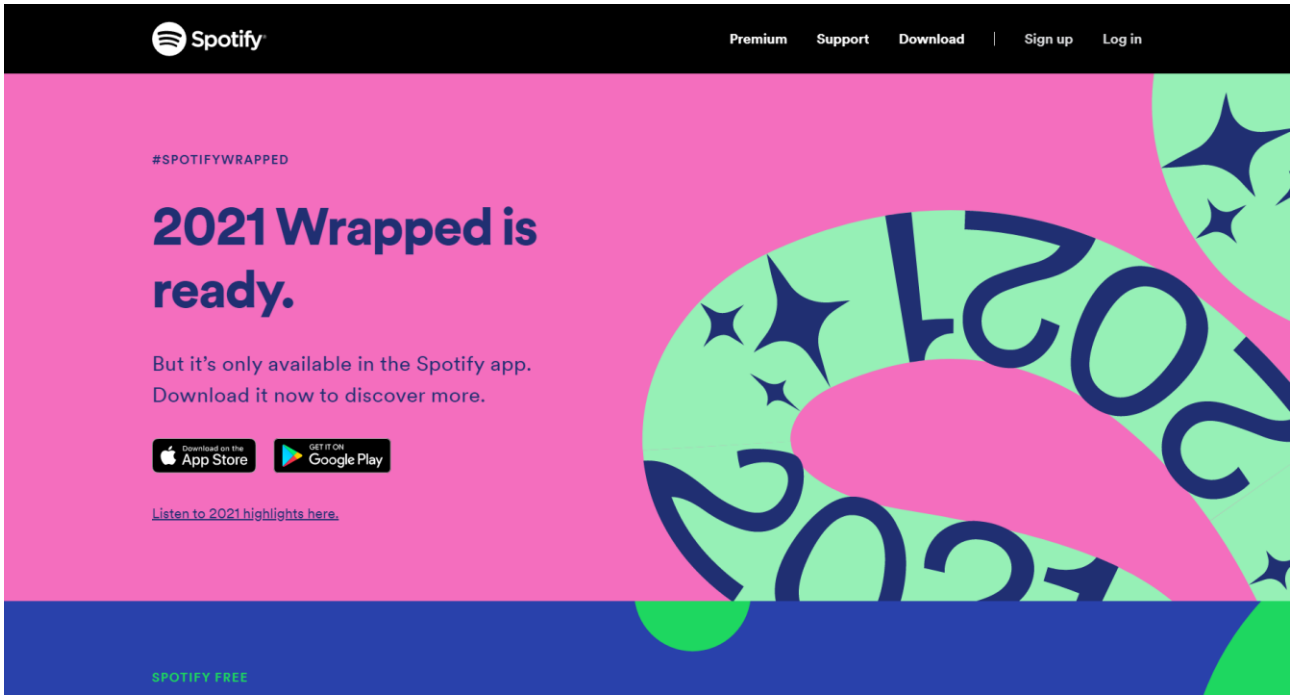

Рис. 1.4. Головна сторінка резервування "Spotify" [3]

1.2.2 Публікації

На сьогоднішній день існує певний ряд наукових робіт, які так чи інакше спрямовані на вирішення поставлених проблем. Розглянемо декілька із них.

Підхід до спільної фільтрації на основі наївного байєсового класифікатора. В роботі [4] запропоновано байєсівську модель, яка не тільки надає такі ж рекомендації, як і моделі матричної факторізації, але ті передбачення також можна пояснити. Модель базується на підходах спільної фільтрації, як на основі користувачів, так і на основі елементів, що рекомендує елементи з використанням схожої інформації про користувачів і елементів

відповідно. Експерименти проведені з використанням чотирьох наборів даних дають хороші результати порівняно з кількома найсучаснішими базовими лініями, досягаючи найкращої продуктивності за допомогою вимірювання якості нормалізованого дисконтованого кумулятивного підсилення (nDCG), а також покращуючи точність прогнозу в деяких наборах даних.

Спільна фільтрація на основі вбудовування графіка знань. Поряд із швидким зростанням масових онлайн-даних рекомендаційні системи використовувалися, як ефективний підхід для фільтрації корисної інформації, які були широко прийняті в багатьох веб-додатках. У цій роботі [5] автори пропонують новий підхід до спільної фільтрації з неявним зворотним зв’язком, заснований на вбудовуванні графа знань. Основним ідеалом є моделювання взаємодії між користувачами та елементами, як графіка знань взаємодії з одним відношенням, векторне представлення, якого вивчається за допомогою вбудовування графа знань. На основі вивченого представлення проблема спільної фільтрації перетворюється на передбачення зв’язку в графі знань взаємодії. KGECF – нейронна мережа для спільної фільтрації на основі вбудовування графів знань, запропонована на основі цього ідеалу та моделі вбудовування графів знань RotatE. Результати експерименту на п’яти наборах даних з різними характеристиками показують, що модель KGECF досягає найсучаснішої продуктивності (SOTA) для всіх наборів даних. І на відміну від інших моделей SOTA, які мають явне падіння продуктивності на наборах даних AMusic і AToy, продуктивність моделі дуже стабільна для всіх наборів даних.

Рекомендована система: уважна нейронна спільна фільтрація. В останні роки нейронні мережі досягли величезного успіху в розпізнаванні мовлення, комп’ютерному зору та обробці природної мови. Проте дослідження нейронних мереж на рекомендаційних системах зазнало відносно менше уваги. В роботі [6] розроблено методи, засновані на нейронних мережах для вирішення ключової проблеми в рекомендаціях – спільної фільтрації – на основі неявного зворотного зв’язку. Хоча деякі нещодавні роботи використовували глибоке навчання для рекомендацій, вони в основному

використовували його для моделювання допоміжної інформації, такої як текстові описи предметів і акустичні особливості музики. Коли справа доходить до моделювання ключового фактора спільної фільтрації – взаємодії між користувачами та функціями елемента, вони все ще вдалися до матричної факторізації та застосували внутрішній продукт до прихованих характеристик користувачів і елементів. А сигнал співпраці прихований у взаємодії користувача та елемента не кодується під час процесу вбудовування. Таким чином отримане вбудовування може бути недостатнім для захоплення ефекту спільної фільтрації. Замінивши внутрішній продукт нейронною архітектурою, яка може вивчати довільну функцію з даних, представлено загальний метод під назвою ANCF (Спільна фільтрація нейронної мережі уваги). ANCF фіксує сигнали спільної фільтрації та уточнює вбудовування користувачів і елементів відповідно до структури графіка. Впроваджуючи механізм уваги, вектор користувача та вектор елемента вивчаються на графіку взаємодії користувача з елементом, інформація взаємодії сусідів об’єднується для кодування, а вбудовування поширюється на графік взаємодії користувача та елемента. Це дає можливість явно вводити сигнали взаємодії користувача та елемента в процес вбудовування. Великі експерименти, проведені на двох реальних наборах даних показують, що відкликання ANCF і ndcg зросли на 30% і 35%, тому запропонований метод ANCF був значно вдосконалений порівняно з найсучаснішим методом. Емпіричні дані показують, що використання більш глибоких шарів нейронних мереж забезпечує кращу продуктивність рекомендацій.

Алгоритм міждоменної спільної фільтрації на основі користувачів на основі моделі лінійної декомпозиції. В роботі [7] запропоновано користувацький міждомений CF-алгоритм на основі моделі лінійної декомпозиції. Автори об’єднали елементи разом і вивчили модель лінійного розкладання, щоб досліджувати зв’язок між повною схожістю та локальною схожістю різних доменів. Спочатку створюється навчальні зразки, обчислюючи схожість будь-яких двох користувачів у різних доменах. Потім розв’язується

лінійна задача найменших квадратів, щоб отримати коефіцієнти розкладання. Нарешті, обчислюється локальна подібність у цільовій області за допомогою моделі декомпозиції. Оскільки обчислення подібність у цільовій області за допомогою розширених рейтингів в інших доменах, можна очікувати, що ця подібність буде точнішою, ніж виміряна подібність обчислена за допомогою розріджених рейтингів у цільовій області. Автори провели великі експерименти, щоб показати, що запропонований алгоритм є ефективним у вирішенні проблеми розрідженості даних у порівнянні з багатьма найсучаснішими методами CF.

Рекомендаційна система для пошуку нерухомості за допомогою методу фільтрації на основі контенту. Розвиток технологій змушує багато галузей бізнесу перейти від офлайнових бізнес-систем до світу електронної комерції. Однією з найпопулярніших електронних комерцій, яку відвідують потенційні покупці, є сайт нерухомості. Враховуючи, що нерухомість є однією з основних вимог для життя, а також одним із найдорожчих активів, які можна мати. У роботі [8] автори розробили веб-систему рекомендацій щодо вибору нерухомості за допомогою методу фільтрації на основі вмісту. Система рекомендацій надає інформацію про властивості на основі поведінки користувача шляхом пошуку рекламного вмісту, який раніше шукав користувач. Кожного разу, коли користувач вибирає вміст оголошення для показу, ця інформація зберігатиметься в базі даних для подальшої обробки для надання рекомендації. Система додатків представить ту саму рекомендацію щодо товару, відповідно до профілю / критеріїв та переваг потенційного покупця. Таким чином, рекомендаційна система допоможе потенційним покупцям визначитися з вибором продукту нерухомості, який вони хочуть придбати, і цей процес може бути забезпечений системою рекомендацій за короткий час.

1.3 Постановка задачі

Є нескінченна кількість локацій, які можна відвідати, але щоб знайти саме ті локації, які можуть зацікавити користувачів, витрачається багато часу на пошук. Тому було запропоновано створити інтелектуальну систему для підбору рекомендованих локацій за допомогою методів машинного навчання, що значно поліпшить процес пошуку користувачами цікавих локацій для відпочинку.

Метою роботи є створення швидкої та ефективної інтелектуальної системи для підбору рекомендованих локацій за допомогою методів машинного навчання на базі існуючого мобільного застосунку.

Об’єктом дослідження є процес функціонування інтелектуальної системи за допомогою методів машинного навчання.

Предметом дослідження є інтелектуальна система для підбору локацій за допомогою методів машинного навчання.

Для досягнення зазначеної мети необхідно виконати наступні завдання:

– проаналізувати схожі програмні продукти для підбору рекомендованих локацій;

– розробити інтелектуальне та програмне забезпечення системи для збору та підготовки даних;

– розробити інтелектуальну систему для підбору рекомендованих локацій;

– протестувати точність системи.

Практична значимість розробленої інтелектуальної системи полягає в тому, щоб покращити досвід користувачів в пошуку локацій, які можуть їх зацікавити. Більшість користувачів знають про існування індивідуальних рекомендацій на великих інтернет-платформах, однак важливість і домінування цієї технології, можливо, більш важливі, ніж багато хто думає. Оскільки добре продумана система рекомендацій звільняє користувача від фільтрації великої

кількості даних, що в свою чергу дозволяє користувачеві отримувати якісний контент за короткий проміжок часу.

Висновки до розділу 1

В даному розділу був проведений докладний опис предметної сфери, побудовані функціональні моделі інтелектуальної системи для визначення локацій методом машинного навчання.

Проаналізовано публікації інших інтелектуальних систем.

Виконана постановка задачі, описані об’єкт, предмет, мета та завдання МКР.

2 МАТЕМАТИЧНІ МОДЕЛІ, МЕТОДИ, ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ДЛЯ ВИРІШЕННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ

2.1 Загальні поняття

Рекомендаційна система або система рекомендацій – це підклас системи фільтрації інформації, яка шукає передбачення "оцінки" або "уподобання", яке користувач дав би елементу.

Рекомендовані системи використовуються в різних галузях, при цьому зазвичай приклади використовуються у вигляді творців плейлистів для відео та музичних послуг, рекомендацій щодо продуктів для інтернет-магазинів або рекомендацій щодо контенту для платформ соціальних мереж та рекомендацій щодо відкритого веб-контенту. Ці системи можуть працювати з використанням одного входу, наприклад, музики, або кількох входів всередині та між платформами, таких як новини, книги та пошукові запити. Є також популярні рекомендаційні системи для конкретних тем, як ресторани та онлайн знайомства. Були також розроблені системи рекомендацій для вивчення дослідницьких статей та експертів, співробітників та фінансових послуг [9].

Наведемо загальний опис типів рекомендаційних систем у наступному підрозділі.

2.1.1 Типи рекомендаційних систем

Фільтрація вмісту (Content Filtering)

Фільтрація вмісту передбачає додаткову інформацію, таку як властивості локації (назва локації, місцеположення, особливості тощо). Системи рекомендацій працюють добре, навіть, якщо до бази додаються нові елементи [10]. Алгоритм рекомендаційної системи очікує включати всі побічні властивості елементів своєї бази.

Важливий аспект фільтрації вмісту:

1) очікує інформацію про предмет;

2) інформація про пункт має бути документом.

Спільна фільтрація (Collaborative Filtering)

Ідея спільної фільтрації полягає в тому, щоб враховувати думки користувачів щодо різних локацій та рекомендувати найкраще місця кожному користувачеві на основі попередніх рейтингів користувача та думки інших подібних типів користувачів [11] (рис. 2.1).

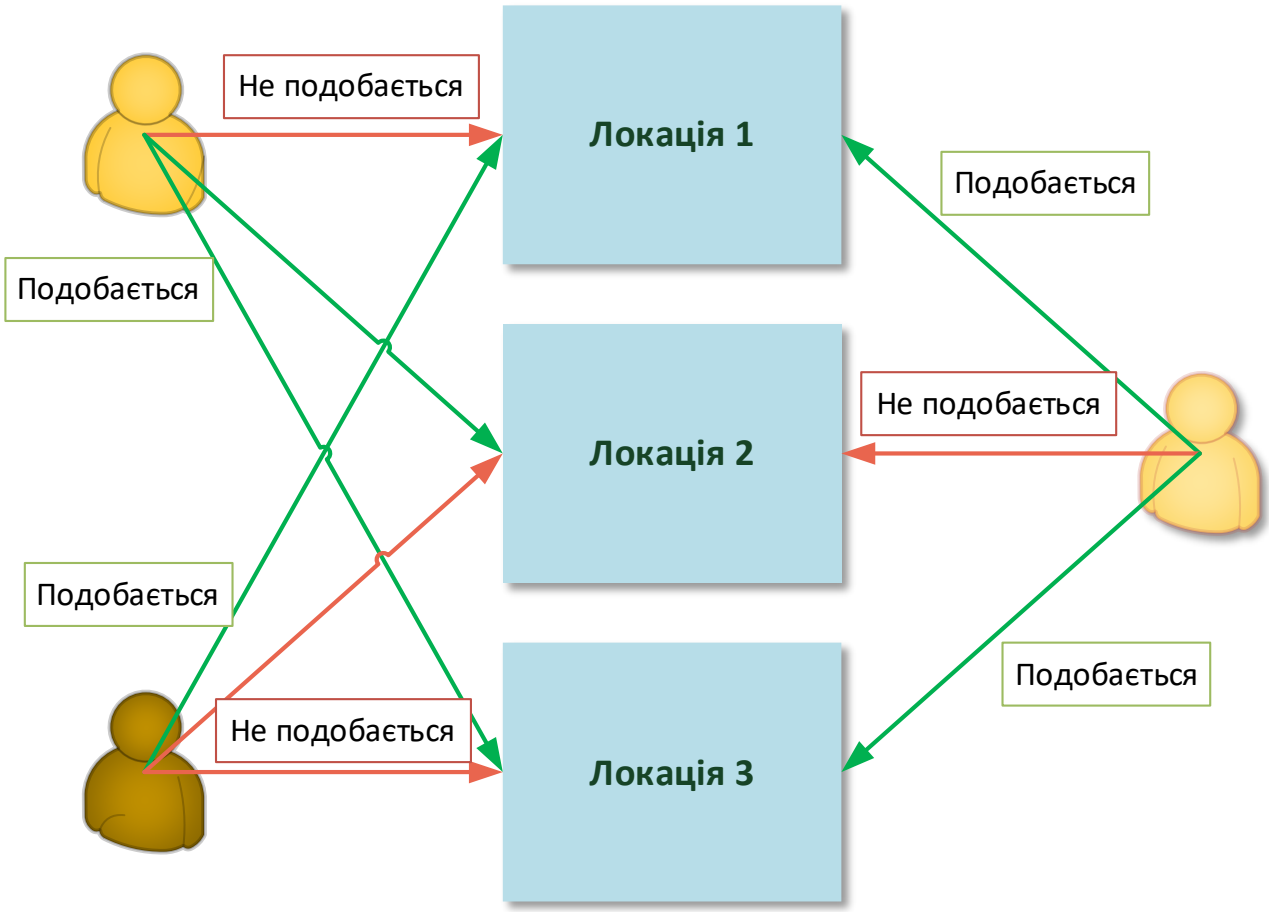

Рис. 2.1. Схема інтересів

Даний тип фільтрації має наступні переваги:

1) для цього не потрібні сторонні знання локацій, та іншого;

2) підхід використовує інформацію, зібрану від інших користувачів, щоб рекомендувати нові елементи поточному користувачеві.

Недоліки:

1) підхід не дає рекомендації щодо нової локації, які не мають рейтингів;

2) це вимагає спільноти користувачів і може мати проблему розрідженості.

Різні техніки спільної фільтрації

Неімовірнісні алгоритми:

1) найближчий сусід на основі користувача (User-based nearest neighbor);
2) найближчий сусід на основі предмета (Item-based nearest neighbor);
3) зменшення розмірності (Reducing dimensionality).

Імовірнісні алгоритми:

1) байєсівська мережева модель (Bayesian-network model);
2) алгоритм EM (EM algorithm).

Матрична факторізація

Матрична факторізація – це спосіб створення прихованих ознак при множенні двох різних типів сутностей. Спільна фільтрація – це застосування матричної факторізації для визначення зв’язку між об’єктами елементів і користувачів [12].

Наприклад, у нас є таблиця рейтингу клієнтів із 6 користувачів і 2 локації, а рейтинги є цілими числами від 1 до 5, матриця наведена на рис. 2.2.

Оскільки не кожен користувач дає оцінки всім локаціям у матриці є багато відсутніх значень, і це призводить до розрідженої матриці. Отже, нульові значення, які не надаються користувачами будуть заповнені значенням 0, щоб заповнені значення були надані для множення. Як можно побачити два користувачі дали високі оцінки локації 2. Отже, за допомогою матричної факторізації ми можемо виявити приховані закономірності, щоб дати прогноз щодо рейтингу подібності в уподобаннях та взаємодії користувачів.

	Юзер 1	Юзер 2	Юзер 3	Юзер 4	Юзер 5	Юзер 6
Локація 1	3	2	**2**	**5**		3
Локація 2		**1**		**5**	5	4

Рис. 2.2. Матриця рейтингів

2.1.2 Проблеми спільної фільтрації

Існує кілька проблем для спільної фільтрації, які можуть суттєво впливати на результати рекомендацій.

Розрідженість

Розріджена матриця – це матриця, в якій більшість значень дорівнює нулю. Частка нульових елементів до ненульових елементів називається розрідженістю матриці. Розрідженість даних виникає через те, що користувачі зазвичай оцінюють лише обмежену кількість елементів[13].

Розрідженість даних визначається співвідношенням порожніх і загальних записів у матриці елемента користувача (рис. 2.3.) (2.1).

$$Sparsity = 1 - |R| \,/\, |I| * |U| \qquad (2.1)$$

де R – рейтинг;

I – локація;

U – користувачі.

	Юзер 1	Юзер 2	Юзер 3	Юзер 4	Юзер 5	Юзер 6
Локація 1	0	0	**2**	**5**	**2**	0
Локація 2	0	**1**	0	**5**	0	0

Рис. 2.3. Матриця рейтингів

Холодний старт

Ця проблема виникає, коли система не має інформації, щоб дати рекомендації для нових користувачів. Як наслідок, методи матричної факторізації не можуть застосовуватися [14].

Ця проблема викликає два спостереження:

1) як рекомендувати користувачам нову локацію;
2) яку локацію рекомендувати новим користувачам.

Для вирішення даної проблеми є декілька рішень:

1) запропонуйте або попросіть користувачів оцінити локацію;
2) голосування за замовчуванням для локації;
3) на початковому етапі використовувати інші методи, наприклад, на основі вмісту або демографічну.

2.1.3 Baseline

Базовий рівень (Baseline) – це метод, який використовує евристики, просту підсумкову статистику, випадковість або машинне навчання для створення прогнозів для набору даних. Ви можете використовувати ці прогнози для вимірювання ефективності базової лінії (наприклад, точності) – цей

показник стане тим, з чим ви порівнюєте будь-який інший алгоритм машинного навчання. Базовий рівень є результатом дуже базової моделі/рішення. Зазвичай розробник створюєте базову лінію, а потім намагаєтеся зробити більш складні рішення, щоб отримати кращий результат.

Розглянемо декілька моделей для створення базового рівня.

Alternating Least Squares

Базова оцінка для невідомого рейтингу r_{ui} позначається b_{ui} і враховує вплив користувача та елемента:

$$b_{ui} = \mu + b_u + b_i \quad (2.2)$$

Параметри b_u і b_i вказують на спостережувані відхилення користувача u і пункту і відповідно від середнього. Наприклад, припустимо, що нам потрібна базова оцінка рейтингу фільму «Титанік» користувача Джона. Тепер скажімо, що середня оцінка всіх фільмів, μ, становить 3.7. Крім того, «Титанік» кращий за звичайний фільм, тому він, як правило, оцінюється на 0.5 зірки вище середнього. З іншого боку, Джон є критикованим користувачем, який, як правило, оцінює на 0.3 зірки нижчий, ніж середній. Таким чином, базова оцінка рейтингу «Титаніка» Джона складала би 3.9 зірки з розрахунку 3.7 – 0.3 + 0.5. Щоб оцінити b_u і b_i, можна розв'язати задачу найменших квадратів [39]:

$$\min_{b_*} \sum_{(u,i)\in\mathcal{K}} (r_{ui} - \mu - b_u - b_i)^2 + \lambda_1 (\sum_u b_u^2 + \sum_i b_i^2) \quad (2.3)$$

$$b_i = \frac{\sum_{u:(u,i)\in\mathcal{K}} (r_{ui} - \mu)}{\lambda_2 + |\{u|(u,i)\in\mathcal{K}\}|} \quad (2.4)$$

$$b_u = \frac{\sum_{i:(u,i)\in\mathcal{K}} (r_{ui} - \mu - b_i)}{\lambda_3 + |\{i|(u,i)\in\mathcal{K}\}|} \quad (2.5)$$

де $\min_{b_*} \sum_{(u,i)\in\mathcal{K}} (r_{ui} - \mu - b_u - b_i)^2$ – пошук b_u і b_i, який може підійти даному рейтингу;

$\lambda_1(\sum_u b_u^2 + \sum_i b_i^2)$ – регулюючий термін, дозволяє уникати перенавчання, зменшуючи величини параметрів;

r_{ui} – справжня оцінка користувача u для елемента i;

$\lambda_1, \lambda_2, \lambda_3$– регулюючий параметр;

μ – середнє значення всіх оцінок.

Stochastic Gradient Descent

Стохастичний градієнтний спуск (часто скорочено SGD) — це ітераційний метод оптимізації цільової функції з відповідними властивостями гладкості (наприклад, диференційована або субдиференційована). Його можна розглядати як стохастичне наближення оптимізації градієнтного спуску, оскільки воно замінює фактичний градієнт (розрахований з усього набору даних) його оцінкою (розрахованою з випадково вибраної підмножини даних) [40]. Стохастичний градієнтний спуск розраховує навчальний приклад на кожній ітерації та оновлює виграний параметр лише на основі цього прикладу:

$$b_i = \sum_{(i)\in\mathcal{K}} l_r(e - \lambda * b_i) \quad (2.6)$$

$$b_u = \sum_{(u)\in\mathcal{K}} l_r(e - \lambda * b_u) \quad (2.7)$$

де $e = r_{ui} - (\mu + b_u + b_i)$ – помилка;

λ – регулюючий параметр;

r_{ui} – справжня оцінка користувача u для елемента i;

μ – середнє значення всіх оцінок.

2.2 Алгоритми прогнозування

В алгоритмах рекомендацій показники подібності є основними компонентами, і їх ефективність матиме прямий вплив на створені рекомендації. Вибір відповідних заходів подібності буде основним фактором підвищення продуктивності методів аналізу даних. В основному вони використовуються для пошуку подібності між користувачами або предметами відповідно до вимог для різних цілей. Розглянемо ряд алгоритмів для прогнозування рейтингів локацій.

2.2.1 Найпростіші алгоритми

NormalPredictor

Алгоритм прогнозування випадкового рейтингу на основі розподілу навчального набору, який вважається нормальним [15].

Прогноз $\hat{r}_{ui}$ генерується з нормального розподілу $\mathrm{N}(\hat{\mu}, \hat{\sigma}^2)$, де $\hat{\mu}$ та $\hat{\sigma}$ оцінюються на основі даних навчання з використанням оцінки максимального правдоподібності:

$$\hat{\mu} = \frac{1}{|R_{train}|} \sum_{r_{ui} \in R_{train}} r_{ui} \tag{2.8}$$

$$\hat{\sigma} = \sqrt{\sum_{r_{ui} \in R_{train}} \frac{(r_{ui} - \hat{\mu})^2}{|R_{train}|}} \tag{2.9}$$

де r_{ui} – справжня оцінка користувача u для елемента i;

R_{train} – навчальна множина.

BaselineOnly

Алгоритм прогнозування базової оцінки для даного користувача та елемента [15].

Якщо користувач u невідомий, то зміщення b_u вважається рівним нулю. Те ж саме стосується пункту i з b_i.

$$\hat{r}_{ui} = b_{ui} = \mu + b_u + b_i \quad (2.10)$$

де b_{ui} – базовий рейтинг користувача u для елемента i;

μ – середнє значення всіх оцінок.

2.2.2 Алгоритм найближчих сусідів

Метод k-найближчих сусідів — метричний алгоритм для автоматичної класифікації об'єктів. Основним принципом методу найближчих сусідів є те, що об'єкт приписується класу, найпоширенішому серед його сусідів. Сусіди беруться, виходячи з множини об'єктів, класи яких уже відомі, і, виходячи з ключового для даного методу значення k, вираховується найчисленніший серед них клас. Кожен об'єкт має кінцеву кількість атрибутів (розмірностей) [41].

KNNBasic

Базовий алгоритм спільної фільтрації. Прогноз $\hat{r}_{ui}$ встановлюється як:

$$\hat{r}_{ui} = \frac{\sum_{v \in N_i^k(u)} sim(u,v) * r_{vi}}{\sum_{v \in N_i^k(u)} sim(u,v)} \quad (2.11)$$

або

$$\hat{r}_{ui} = \frac{\sum_{j \in N_u^k(i)} sim(i,j) * r_{uj}}{\sum_{j \in N_u^k(i)} sim(i,j)} \quad (2.12)$$

де $N_u^k(i)$ – k найближчих сусідів елемента i, які оцінюються користувачем u. Цей набір обчислюється за допомогою метрики подібності;

$N_i^k(u)$ – k найближчих сусідів користувача u, які оцінили пункт i. Цей набір обчислюється за допомогою метрики подібності;

r_{uj} – справжня оцінка користувача u для елемента j;

sim(i, j), sim(u, v) – метрика подібності.

KNNWithMeans

Базовий алгоритм спільної фільтрації, що враховує середні оцінки кожного користувача. Прогноз $\hat{r}_{ui}$ встановлюється як:

$$\hat{r}_{ui} = \mu_u + \frac{\sum_{v \in N_i^k(u)} sim(u,v) * (r_{vi} - \mu_v)}{\sum_{v \in N_i^k(u)} sim(u,v)} \quad (2.13)$$

або

$$\hat{r}_{ui} = \mu_i + \frac{\sum_{j \in N_u^k(i)} sim(i,j) * (r_{uj} - \mu_j)}{\sum_{j \in N_u^k(i)} sim(i,j)} \quad (2.14)$$

де $N_u^k(i)$ – k найближчих сусідів елемента i, які оцінюються користувачем u. Цей набір обчислюється за допомогою метрики подібності;

$N_i^k(u)$ – k найближчих сусідів користувача u, які оцінили пункт i. Цей набір обчислюється за допомогою метрики подібності;

r_{uj} – справжня оцінка користувача u для елемента j;

sim(i, j), sim(u, v) – метрика подібності;

μ_u – середнє значення всіх оцінок, наданих користувачем u;

μ_i – середнє значення всіх оцінок, наданих пункту i.

KNNWithZScore

Базовий алгоритм спільної фільтрації, що враховує нормалізацію z-показника кожного користувача. Прогноз $\hat{r}_{ui}$ встановлюється як:

$$\hat{r}_{ui} = \mu_u + \sigma_u \frac{\sum_{v \in N_i^k(u)} sim(u,v) * \frac{r_{vi} - \mu_v}{\sigma_v}}{\sum_{v \in N_i^k(u)} sim(u,v)} \quad (2.15)$$

або

$$\hat{r}_{ui} = \mu_i + \sigma_i \frac{\sum_{j \in N_u^k(i)} sim(i,j) * \frac{r_{uj} - \mu_j}{\sigma_j}}{\sum_{j \in N_u^k(i)} sim(i,j)} \quad (2.16)$$

де $N_u^k(i)$ – k найближчих сусідів елемента i, які оцінюються користувачем u. Цей набір обчислюється за допомогою метрики подібності;

$N_i^k(u)$ – k найближчих сусідів користувача u, які оцінили пункт i. Цей набір обчислюється за допомогою метрики подібності;

r_{uj} – справжня оцінка користувача u для елемента j;

sim(i, j), sim(u, v) – метрика подібності;

μ_u – середнє значення всіх оцінок, наданих користувачем u;

μ_i – середнє значення всіх оцінок, наданих пункту i;

σ_u – стандартне відхилення всіх оцінок, наданих користувачем u;

σ_i – стандартне відхилення всіх оцінок, наданих пункту i.

KNNBaseline

Базовий алгоритм спільної фільтрації з урахуванням базового рейтингу. Прогноз $\hat{r}_{ui}$ встановлюється як:

$$\hat{r}_{ui} = b_{ui} + \frac{\sum_{v \in N_i^k(u)} sim(u,v) * (r_{vi} - b_{vi})}{\sum_{v \in N_i^k(u)} sim(u,v)} \quad (2.17)$$

або

$$\hat{r}_{ui} = b_{ui} + \frac{\sum_{j \in N_u^k(i)} sim(i,j) * (r_{uj} - b_{uj})}{\sum_{j \in N_u^k(i)} sim(i,j)} \quad (2.18)$$

де $N_u^k(i)$ – k найближчих сусідів елемента i, які оцінюються користувачем u. Цей набір обчислюється за допомогою метрики подібності;

$N_i^k(u)$ – k найближчих сусідів користувача u, які оцінили пункт i. Цей набір обчислюється за допомогою метрики подібності;

r_{uj} – справжня оцінка користувача u для елемента j;

sim(i, j), sim(u, v) – метрика подібності;

b_{ui} – базовий рейтинг користувача u для елемента i.

2.2.3 Алгоритми матричної факторізації

Матричні декомпозиції – це методи, які розкладають матрицю на складові частини, що полегшують обчислення більш складних матричних операцій. Методи розкладання матриці, також звані методами матричної факторізації, є основою лінійної алгебри в комп’ютерах, навіть для основних операцій, таких як розв’язування систем лінійних рівнянь, обчислення оберненого і обчислення визначника матриці.

SVD

Відомий алгоритм SVD популяризований Саймоном Функом під час премії Netflix. Якщо базові лінії не використовуються, це еквівалентно імовірнісній матриці факторізації [16].

Якщо користувач u невідомий, то зміщення b_u і фактори p_u вважаються рівними нулю. Те ж саме стосується пункту і з b_i та q_i.

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^T p_u \quad (2.19)$$

Щоб оцінити все невідоме, ми мінімізуємо наступну регуляризовану квадратну помилку:

$$\sum_{r_{ui} \in R_{train}} (r_{ui} - \hat{r}_{ui})^2 + \lambda(b_i^2 + b_u^2 + \|q_i\|^2 + \|p_u\|^2) \quad (2.20)$$

Мінімізація виконується за допомогою дуже простого стохастичного градієнтного спуску:

$$b_u \leftarrow b_u + \gamma(e_{ui} - \lambda b_u) \quad (2.21)$$

$$b_i \leftarrow b_i + \gamma(e_{ui} - \lambda b_i) \quad (2.22)$$

$$p_u \leftarrow p_u + \gamma(e_{ui} * q_i - \lambda p_u) \quad (2.23)$$

$$q_i \leftarrow q_i + \gamma(e_{ui} * p_u - \lambda q_i) \quad (2.24)$$

$$e_{ui} = r_{ui} - \hat{r}_{ui} \quad (2.25)$$

де p_u – фактори користувача;

q_i – фактори предмета;

b_i – упередження для предмету;

b_u – упередження для користувача.

Ці кроки виконуються для всіх рейтингів тренувального набору і повторюються N разів. Базові лінії ініціалізуються на 0. Фактори користувача та елемента ініціалізуються випадковим чином відповідно до нормального розподілу.

NMF

Алгоритм спільної фільтрації заснований на факторізації невід'ємної матриці [17]. Цей алгоритм дуже схожий на SVD. Прогноз $\hat{r}_{ui}$ встановлюється як:

$$\hat{r}_{ui} = q_i^T p_u \tag{2.26}$$

де коефіцієнти користувачів і елементів залишаються позитивними.

Процедура оптимізації являє собою (регуляризований) стохастичний градієнтний спуск з конкретним вибором розміру кроку, що забезпечує невід'ємність факторів, за умови, що їх початкові значення також додатні.

На кожному кроці процедури SGD коефіцієнти f або користувача u та елемент і оновлюються таким чином:

$$p_{uf} \leftarrow p_{uf} * \frac{\sum_{i \in I_u} q_{if} * r_{ui}}{\sum_{i \in I_u} q_{if} * \hat{r}_{ui} + \lambda_u |I_u| p_{uf}} \tag{2.27}$$

$$q_{if} \leftarrow q_{if} * \frac{\sum_{u \in U_i} p_{uf} * r_{ui}}{\sum_{u \in U_i} p_{uf} * \hat{r}_{ui} + \lambda_i |U_i| q_{if}} \tag{2.28}$$

де p_u – фактори користувача;

q_i – фактори предмета;

b_i – упередження для предмету;

b_u – упередження для користувача;

λ_i, λ_u – параметри регуляризації.

PMF

Модель імовірнісної матричної факторізації (PMF), яка лінійно масштабується з кількістю спостережень і, що більш важливо добре працює на великому, розрідженому та дуже незбалансованому наборі даних Netflix [18].

Припустимо, що у нас є M локацій, N користувачів і цілі оцінки від 1 до K^1. Нехай R_{ij} представляє рейтинг користувача і для фільму j, $U \in R^{D\times N}$ та $V \in R^{D\times N}$ — приховані матриці функцій користувача та локації, причому вектори стовпців U_i та V_j представляють специфічні для користувача та специфічні для локації латентні вектори функцій відповідно. Визначимо умовний розподіл за спостережуваними рейтингами як:

$$p(R|U,V,\ \sigma^2) = \prod_{i=1}^{N}\prod_{j=1}^{M}\left[\mathcal{N}(R_{ij}|U_i^T V_j,\sigma^2)\right]^{I_{ij}} \qquad (2.29)$$

де $\mathcal{N}(x|\mu,\sigma^2)$ – функція щільності ймовірності розподілу Гаусса із середнім μ та дисперсією σ^2;

I_{ij} – є індикаторною функцією, яка дорівнює 1, якщо користувач і оцінив фільм j, і дорівнює 0 в іншому випадку.

Також розміщується сферичні гауссові априори з нульовим середнім [1, 11] на векторах функцій користувача та локації:

$$p(U|\ \sigma_U^2) = \prod_{i=1}^{N}\mathcal{N}(U_i|0,\sigma_U^2 I) \qquad (2.30)$$

$$p(V|\ \sigma_V^2) = \prod_{j=1}^{M}\mathcal{N}(V_j|0,\sigma_V^2 I) \qquad (2.31)$$

Журнал апостеріорного розподілу за функціями користувача та фільму дається за допомогою:

$$In\,p(U,V|R,\sigma^2,\sigma_V^2,\sigma_U^2) = -\frac{1}{2\sigma^2}\sum_{i=1}^{N}\sum_{j=1}^{M}I_{ij}\left(R_{ij}-U_i^T V_j\right)^2 - \frac{1}{2\sigma_U^2}\sum_{i=1}^{N}U_i^T U_i - \frac{1}{2\sigma_V^2}\sum_{j=1}^{M}V_j^T V_j - \frac{1}{2}\Big(\Big(\sum_{i=1}^{N}\sum_{j=1}^{M}I_{ij}\Big)In\,\sigma^2 + ND\ In\,\sigma_U^2 + MD\ In\,\sigma_V^2\Big) + C \qquad (2.32)$$

де C – є константою, яка не залежить від параметрів.

BPMF

Моделі на основі факторів широко використовуються в області спільної фільтрації для моделювання уподобань користувачів. Ідея Байєсівської імовірнісної матричної факторизації моделей полягає в тому, що переваги користувача визначаються невеликою кількістю неспостережуваних факторів. У лінійній факторній моделі оцінка користувача елементу моделюється за допомогою внутрішнього добутку вектора фактора елемента та вектора фактора користувача [19].

Наведемо вибірку Гіббса для байєсівського PM де для кожної ітерації $t = 1, \dots, T$ виконується наступний алгоритм:

1) зразок гіперпараметра (2.33);

2) для кожного $i = 1, \dots, N$ паралельних зразків функцій користувача (2.28);

3) для кожного $i = 1, \dots, M$ паралельних зразків функцій локації (2.34).

Умовний розподіл за гіперпараметрами користувача, обумовлений матрицею характеристик користувача U, визначається розподілом Гаусса-Вішарта:

$$p(\mu_U | \mu_0^*, (\beta_0^* \Lambda_U)^{-1}) W(\Lambda_U | W_0^*, v_0^*) \qquad (2.33)$$

де $\mu_0^* = \frac{\beta_0 \mu_0 + N\bar{U}}{\beta_0 + N}$ – гіперпараметр;

$\beta_0^* = \beta_0 + N,\ v_0^* = v_0 + N;$

$[W_0^*]^{-1} = W_0^{-1} + N\bar{S} + \frac{\beta_0 N}{\beta_0 + N}(\mu_0 - \bar{U})(\mu_0 - \bar{U})^T;$

$\bar{U} = \frac{1}{N}\sum_{i=1}^{N} U_i,\ \ \bar{S} = \frac{1}{N}\sum_{i=1}^{N} U_i U_i^T;$

Умовний розподіл за вектором характеристик користувача Ui, зумовлений особливостями локації, спостережуваною матрицею оцінок користувачів R, а значення гіперпараметрів є гауссовськими:

$$p(U_i|R,\ V, \theta_U,\ \sigma) = \mathcal{N}(\mathrm{U}_i|\mu_i^*, [\Lambda_i^*]^{-1})$$
$$\sim \prod_{j=1}^{M} \left[\mathcal{N}\left(\mathrm{R}_{ij}\middle|U_i^T V_i, \sigma^{-1}\right)\right]^{I_{ij}} p(\mathrm{U}_i|\mu_U, \Lambda_U) \qquad (2.34)$$

де $\Lambda_i^* = \Lambda_U + \sigma \sum_{i=1}^{M} \left[V_j V_j^T\right]^{I_{ij}}$;

$\mu_i^* = [\Lambda_i^*]^{-1}(\sigma \sum_{i=1}^{M} \left[V_j R_{ij}\right]^{I_{ij}} + \Lambda_U \mu_U)$.

ALS

Чергування найменших квадратів (ALS) – це ітераційний алгоритм альтернативних найменших квадратів із зваженою λ-регуляризацією (ALS-WR) для вирішення проблеми наближення низького рангу. Системи рекомендацій намагаються рекомендувати товари зацікавленим потенційним клієнтам на основі наявної інформації. Успішна система рекомендацій може значно підвищити дохід компаній електронної комерції або полегшити взаємодію користувачів в онлайн-спільнотах. Серед рекомендаційних систем підходи на основі вмісту аналізують вміст елементів, щоб визначити пов’язані елементи, тоді як спільна фільтрація використовує сукупну поведінку/смак великої кількості користувачів, щоб пропонувати відповідні елементи для конкретних користувачів [20].

Для розв'язання задачи розкладки матриці низького рангу необхідно виконати наступні кроки:

Крок 1. Ініціалізувати матрицю M, призначивши середній рейтинг для цієї локації як перший рядок, і невеликі випадкові числа для решти записів.

Крок 2. Виправити M, розв’язати U, мінімізуючи цільову функцію (сума квадратів помилок);

Крок 3. Зафіксувати U, озв'язати M, мінімізуючи цільову функцію аналогічним чином;

Крок 4. Повторювати кроки 2 і 3, доки не буде задоволено критерій зупинки.

Матриці регуляризації зваженої λ-регуляризація працює:

$$f(U,M) = \sum_{(i,j)\in I}\left(r_{ij} - u_i^T m_j\right)^2 + \lambda\left(\sum_i n_{u_i}\|u_i\|^2 + \sum_j n_{m_j}\left\|m_j\right\|^2\right) \quad (2.35)$$

де n_{u_i}, n_{m_j} – позначають кількість оцінок користувача і та локації j відповідно.

Даний стовпець U, скажімо, u_i , визначається шляхом розв'язування регуляризованої лінійної задачі найменших квадратів, що включає відомі рейтинги користувача і та вектори характеристик m_j локації, які оцінив користувач і.

$$\frac{1}{2}\frac{\partial f}{\partial_{u_{ki}}} = 0, \forall i,k \quad \Longrightarrow \sum_{j\in I_j}\left(u_i^T m_j - r_{ij}\right)m_{kj} + \lambda n_{u_i}u_{ki} = 0, \forall i,k \quad \Longrightarrow$$

$$\sum_{j\in I_j} m_{kj}m_i^T u_i + \lambda n_{u_i}u_{ki} = \sum_{j\in I_i} m_{kj}r_{ij}\,, \forall i,k \Longrightarrow \left(M_{I_i}M_{I_i}^T + \lambda n_{u_i}E\right)u_i = M_{I_i}R^T(i,I_i), \forall i \Longrightarrow u_i = A_i^{-1}V_i, \forall i \quad (2.36)$$

де $A_i = M_{I_i}M_{I_i}^T + \lambda n_{u_i}E, V_i = M_{I_i}R^T(i,I_i), E$– це $n_f \times n_f$ тотожна матриця;

M_{I_i}– позначає підматрицю M, де вибрані стовпці $i \in I_i$;

$R(i,I_i)$ — це вектор-рядок, де беруться стовпці $j \in I_i$ і-го рядка R.

2.2.4 Інші алгоритми

SlopeOne

Простий, але точний алгоритм спільної фільтрації [21]. Прогноз $\hat{r}_{ui}$ встановлюється як:

$$\hat{r}_{ui} = \mu_u + \frac{1}{|R_i(U)|}\sum_{j \in R_i(u)} dev(i,j) \quad (2.37)$$

де $dev(i,j)$ – визначається як середня різниця між рейтингами i та j;

$R_i(u)$– це набір релевантних елементів, тобто набір елементів j, оцінених u, які також мають принаймні одного спільного користувача з i;

μ_u – середнє значення всіх оцінок, наданих користувачем u.

$$dev(i,j) = \frac{1}{|U_{ij}|}\sum_{u \in U_{ij}} r_{ui} - r_{uj} \quad (2.38)$$

де r_{ui} – справжня оцінка користувача u для елемента i.;

U_{ij}– набір усіх користувачів, які оцінили обидва пункти i та j.

CoClustering

Алгоритм спільної фільтрації на основі спільної кластеризації. В основному користувачам і елементам призначаються деякі кластери C_u, C_i, а також деякі спільні кластери C_{iu} [22]. Прогноз $\hat{r}_{ui}$ встановлюється як:

$$\hat{r}_{ui} = \overline{C_{ui}} + \left(\mu_u - \overline{C_u}\right) + \left(\mu_i - \overline{C_i}\right) \quad (2.39)$$

де $\overline{C_{iu}}$ – середній рейтинг спільного кластера C_{iu};

$\overline{C_u}$, $\overline{C_\imath}$– середня оцінка кластера користувачів та предметів;

μ_u – середнє значення всіх оцінок, наданих користувачем u;

μ_i – середнє значення всіх оцінок, наданих пункту i.

2.3 Оцінка подібності

У сенсі Data Mining міра подібності – це відстань з розмірами, що описують особливості об’єкта. Це означає, що якщо відстань між двома точками даних мала, то існує високий ступінь подібності між об’єктами і навпаки. Подібність суб’єктивна і сильно залежить від контексту та застосування. Наприклад, схожість овочів можна визначити за смаком, розміром, кольором тощо.

Cosine

Косинусная подібність – це математичне обчислення, яке повідомляє нам подібність двох векторів A і B. Фактично, ми обчислюємо косинус кута тета між цими двома векторами. Функція повертає значення від -1, що вказує повністю протилежні вектори до 1, що вказує на той самий вектор. 0 вказує на відсутність кореляції між векторами, а проміжні значення вказують на проміжні рівні подібності [23]. Косинусна подібність визначається як:

$$sim(u,v) = \frac{\sum_{i \in I_{uv}} r_{ui} * r_{vi}}{\sqrt{\sum_{i \in I_{uv}} r_{ui}^2 *} \sqrt{\sum_{i \in I_{uv}} r_{vi}^2}} \quad (2.40)$$

де r_{ui} – справжня оцінка користувача u для елемента j;

Функція подібності косинуса лінійно збільшується за складністю, коли ми збільшуємо розміри A і B. Точковий добуток A і B зажадає n+t більше обчислень, якщо ми додамо ще t значень до A і B, і величина кожного з них також зросте лінійно.

MSD

Середня квадратична різниця (MSD) обчислює лише середню різницю між обома користувачами, але ігнорує частку загальних оцінок. Це може

призвести до низької точності[24]. Середньоквадратична різниця визначається як:

$$msd(u,v) = \frac{1}{|I_{uv}|} * \sum_{i \in I_{uv}} (r_{ui} - r_{vi})^2 \quad (2.41)$$

$$msd\_sim(u,v) = \frac{1}{msd(u,v) + 1} \quad (2.42)$$

де r_{ui} – справжня оцінка користувача u для елемента j;

$msd(u,v)$ – обчислює помилку, тобто визначає, наскільки різні оцінки користувачів;

$msd\_sim(u,v)$ – обчислює подібність.

Pearson

Коефіцієнти кореляції використовуються для вимірювання того, наскільки сильний зв'язок між двома змінними. Існує кілька типів коефіцієнта кореляції, але найпопулярнішим є коефіцієнт Пірсона. Кореляція Пірсона (також називається R Pearson) – це коефіцієнт кореляції, який зазвичай використовується в лінійній регресії [25]. Кореляція Пірсона визначається як:

$$sim(u,v) = \frac{\sum_{i \in I_{uv}} (r_{ui} - \mu_u) * (r_{vi} - \mu_v)}{\sqrt{\sum_{i \in I_{uv}} (r_{ui} - \mu_u)^2} * \sqrt{\sum_{i \in I_{uv}} (r_{vi} - \mu_v)^2}} \quad (2.43)$$

де r_{ui} – справжня оцінка користувача u для елемента j;

μ_u – середнє значення всіх оцінок, наданих користувачем u;

μ_v – середнє значення всіх оцінок, наданих пункту v;

Pearson Baseline

Скорочений коефіцієнт кореляції Пірсона з базовою лінією, який базується на базовій схожості Пірсона (тобто ми беремо базові прогнози замість середньої оцінки користувача/елемента) [26]. Кореляція Пірсона з базовою лінією визначається як:

$$sim(u,v) = \hat{p}_{uv} = \frac{\sum_{i \in I_{uv}} (r_{ui} - b_{ui}) * (r_{vi} - b_{vi})}{\sqrt{\sum_{i \in I_{uv}} (r_{ui} - b_{ui})^2} * \sqrt{\sum_{i \in I_{uv}} (r_{vi} - b_{vi})^2}} \qquad (2.44)$$

Згорнутий коефіцієнт кореляції Пірсона та базової лінії тоді визначається як:

$$shrunk\_sim_{(u,v)} = \frac{|I_{uv}| - 1}{|I_{uv}| - 1 + shrinkage} * \hat{p}_{uv} \qquad (2.46)$$

де r_{ui} – справжня оцінка користувача u для елемента j;

$shrinkage$ – параметр усадки (цуй параметр допомагає уникнути перенавчання, коли доступно лише кілька оцінок);

b_{ui} – базовий рейтинг користувача u для елемента i.

2.4 Оцінка точності прогнозної моделі та прогнозів

Точність прогнозної моделі – близькість розрахункових значень до фактичних спостережень за період апроксимації. Про точність прогнозу прийнято судити по величині погрішності (помилки) прогнозу – різниці між прогнозованим і фактичним значенням досліджуваної перемінної. Однак такий підхід до оцінки точності можливий тільки в двох випадках. По-перше, коли період попередження вже закінчився і дослідник має фактичні значення перемінної. При короткостроковому прогнозуванні це цілком реально. По-друге, коли прогноз розробляється ретроспективно, тобто прогнозування

здійснюється для деякого моменту часу в минулому, для якого вже маються фактичні дані. Так роблять у тих випадках, коли перевіряється розроблена методика прогнозу.

RMSE

Середньоквадратичне відхилення (RMSD) або середньоквадратична помилка (RMSE) є часто використовуваною мірою відмінностей між значеннями (вибірковими або сукупними значеннями), передбаченими моделлю або оцінювачем, і спостережуваними значеннями. RMSD являє собою квадратний корінь з моменту другої вибірки відмінностей між прогнозованими і спостережуваними значеннями або середнє квадратичне цих відмінностей. Ці відхилення називаються залишками, коли обчислення виконуються за вибіркою даних, яка була використана для оцінки, і називаються помилками (або помилками передбачення), коли обчислюються поза вибіркою. RMSD служить для об'єднання величин помилок у прогнозах для різних точок даних в єдиний вимір потужності прогнозування. RMSD є мірою точності, щоб порівняти помилки прогнозування різних моделей для певного набору даних, а не між наборами даних, оскільки він залежить від масштабу [27].

$$RMSE = \sqrt{\frac{1}{|\hat{R}|}\sum_{\hat{r}_{ui}\in\hat{R}}(r_{ui} - \hat{r}_{ui})^2} \qquad (2.46)$$

де r_{ui} – справжня оцінка користувача u для елемента j;

$\hat{r}_{ui}$– приблизний рейтинг користувача u для елемента i.

MSE

У статистиці середня квадратична помилка (MSE) або середнє квадратичне відхилення (MSD) оцінювача (процедури оцінки неспостережуваної величини) вимірює середнє квадратів помилок, тобто середню квадратичну різницю між розрахунковими і фактичними значеннями. MSE є функцією ризику, що відповідає очікуваному значенню квадрата втрат помилки. Той факт, що MSE майже завжди є строго додатним (а не нульовим), пояснюється випадковістю або тому, що оцінювач не враховує інформацію, яка могла б дати більш точну оцінку [28].

$$MSE = \frac{1}{|\hat{R}|}\sum_{\hat{r}_{ui}\in\hat{R}}(r_{ui} - \hat{r}_{ui})^2 \quad (2.47)$$

де r_{ui} – справжня оцінка користувача u для елемента j;

$\hat{r}_{ui}$– приблизний рейтинг користувача u для елемента i.

MAE

У статистиці середня абсолютна помилка (MAE) – це міра помилок між парними спостереженнями, що виражають одне й те саме явище. Приклади Y проти X включають порівняння прогнозованого та спостережуваного, наступного часу та початкового часу, а також одну методику вимірювання та альтернативну методику вимірювання [29].

$$MAE = \frac{1}{|\hat{R}|}\sum_{\hat{r}_{ui}\in\hat{R}}|r_{ui} - \hat{r}_{ui}| \quad (2.48)$$

де r_{ui} – справжня оцінка користувача u для елемента j;

$\hat{r}_{ui}$– приблизний рейтинг користувача u для елемента i.

FCP

Частка конкордантних пар – це просто частка конкордантних пар серед усіх пар (сума за всіма користувачами) [30]. Концепція полягає в наступному. Припустимо, що користувач оцінив n продуктів, тоді існує n*(n-1)/2 унікальних пар оцінок. Якщо продукт A отримує від користувача вищу оцінку, ніж продукт B, і модель прогнозує те саме, A і B є конкордантною парою, в іншому випадку – дискордантною парою (чим більше значення, тим краще).

$$FCP = \frac{\sum_u n_c^u}{\sum_u n_c^u + \sum_u n_d^u} \quad (2.49)$$

де $n_c^u = |\{(i,j)|\hat{r}_{ui} > \hat{r}_{uj}\ and\ r_{ui} > r_{uj}\}|$;

$\hat{r}_{ui}$– приблизний рейтинг користувача u для елемента i;

r_{ui} – справжня оцінка користувача u для елемента j.

Precision & Recall

Precision (також називається позитивною прогнозною цінністю) — це частка релевантних екземплярів серед вилучених екземплярів, тоді як Recall (також відоме як чутливість) — це частка релевантних екземплярів, які були отримані. Тому і точність, і відкликання базуються на релевантності (рис. 2.4.) [31].

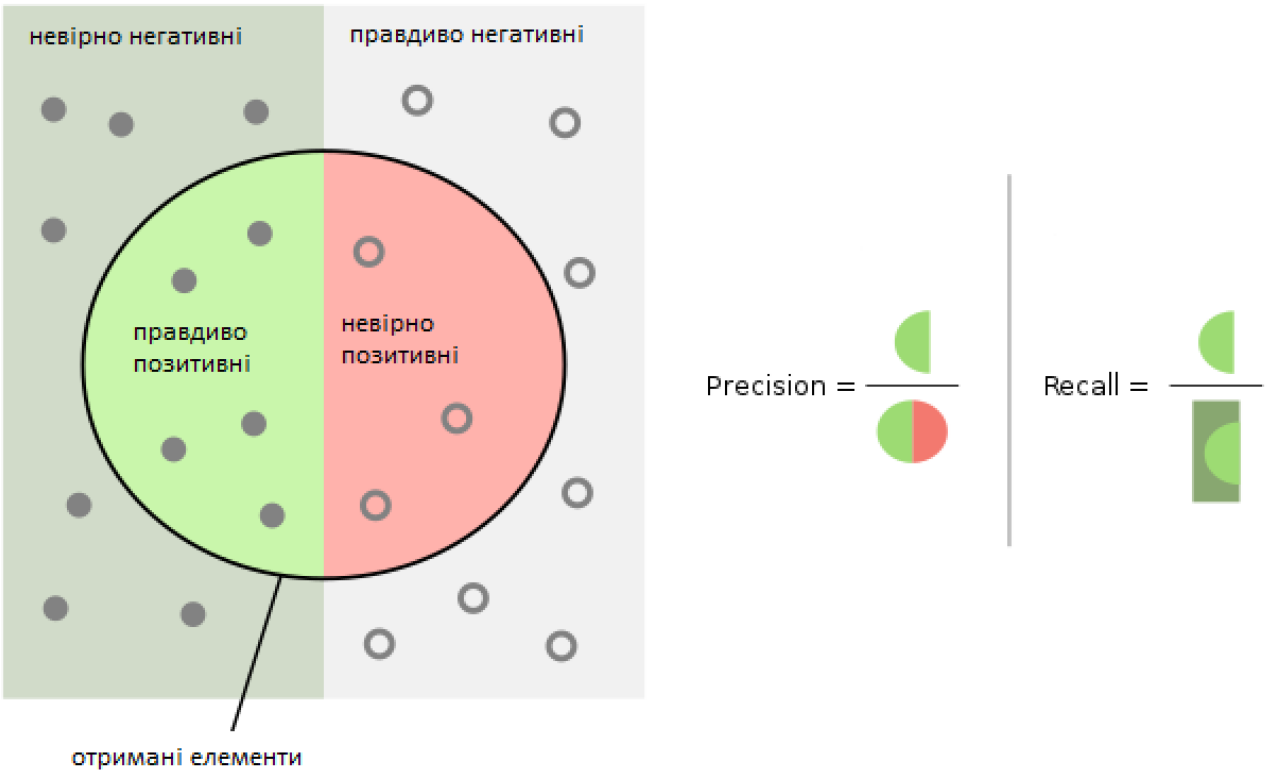

Рис. 2.4. Precision and recall

Precision намагається відповісти на таке запитання: «Яка частка позитивних ідентифікацій була насправді правильною?». Математично Precision визначається за формулою 2.43.

Recall спроби відповісти на таке запитання: «Яка частка фактичних позитивних результатів була визначена правильно?». Математично Recall визначається за формулою 2.44.

$$Precision = \frac{TP}{TP + FP} \tag{2.50}$$

$$Recall = \frac{TP}{TP + FN} \tag{2.51}$$

де TP– коректно позитивний прогноз;

FN– неправильный негативний прогноз;

FP – коректно негативний прогноз.

F1 Score

У статистичному аналізі бінарної класифікації F-оцінка або F-міра є мірою точності тесту. Він розраховується на основі Precision та Recall тесту. Оцінка F1 – це середнє гармонійне значення Precision та Recall [32].

Найвище можливе значення F-показу становить 1.0, що вказує на ідеальну Precision і Recall, а найменше можливе значення дорівнює 0, якщо точність або відкликання дорівнює нулю.

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (2.52)$$

Висновки до розділу 2

В даному розділу був проведений докладний опис алгоритмів прогнозування, оцінки подібності, оцінки точності прогнозної моделі та прогнозів.

Розкрито загальні поняття рекомендаційних систем та описано проблеми спільної фільтрації.

3 МОДЕЛЮВАННЯ ТА ДОСЛІДЖЕННЯ ОТРИМАНИХ РЕЗУЛЬТАТІВ

3.1 Опис даних прогнозування

Перш ніж почати опрацьовувати результати роботи систем рекомендацій необхідно зробити опис вхідних даних. Дану вибірку даних було взято з відкритих ресурсів та продезінфіковано від чутливих даних [33][34]. Набір даних складається з 34643 документів локацій та 5648 документів тестових користувачів після попередньої підготовки даних.

Розпочнемо опис з параметрів документа локацій (табл. 3.1).

Таблиця 3.1

Параметри документа локації

Параметр	Тип даних	Опис
ID	int	Унікальний ідентифікатор
NAME	string	Ім'я локації
DESCRIPTION	string	Опис локації
CATEGORY	enum	Категорія локації
TYPE	enum	Тип локації
REVIEW_COUNT	int	Кількість відгуків
RATE	int	Середня оцінка з усіх відгуків
LATITUDE	double	Координати: Широта
LONGITUDE	double	Координати: Довгота
IS_GATHERING	boolean	Це подія
IS_LOCATION	boolean	Це звичайне місце
IS_SHARE_PROPERTY	boolean	Це місце яке можно забронювати
LIKES	int	Кількість людей, яким сподобалося це місце
PHOTOS	int	Кількість фотографій
FEATURES	string[]	Список доступних можливостей
ACTIVITIES	string[]	Список доступних розваг
REPORTS	int	Кількість скарг

Розглянемо більш детально характер даних локацій. Базуючись на рисунку 3.1. можна побачити абсолютну перевагу типу «CAMP_SITE» та категорії «Organized camping» над усіма іншими даними. Це свідчить про те, що в більшості випадків локації будуть схожі між собою.

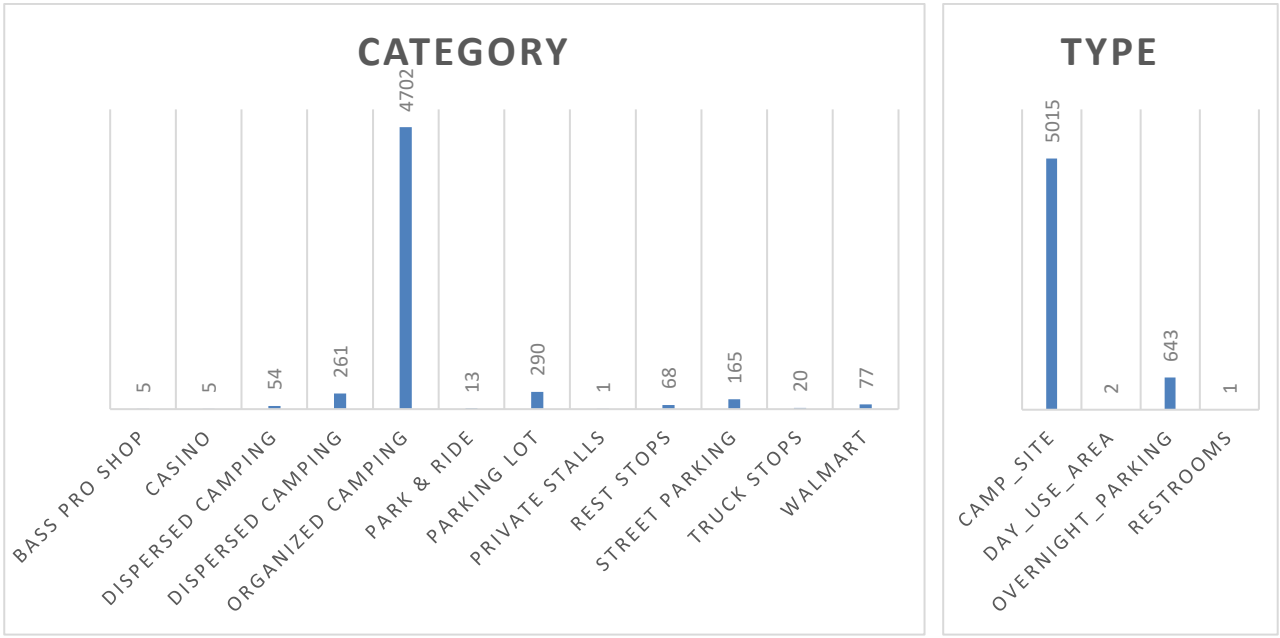

Рис. 3.1. «Категорія» та «Тип» локації

Спираючись на рисунок 3.2, можна припустити, що користувачі недостатньо активно залишають свої враження від локацій. Дана ситуація може свідчити про наявність певної розрідженості даних.

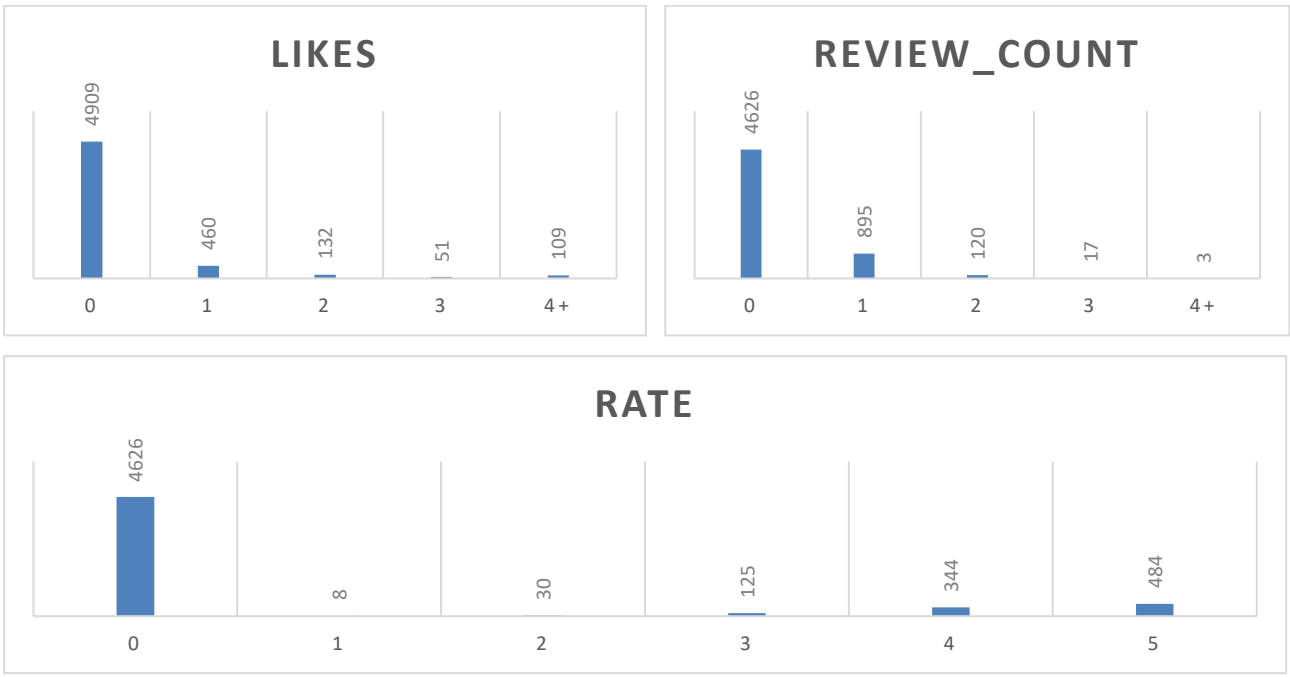

Рис. 3.2. «Рейтинг», «Кількість рейтингів» та «Кількість лайків» локації

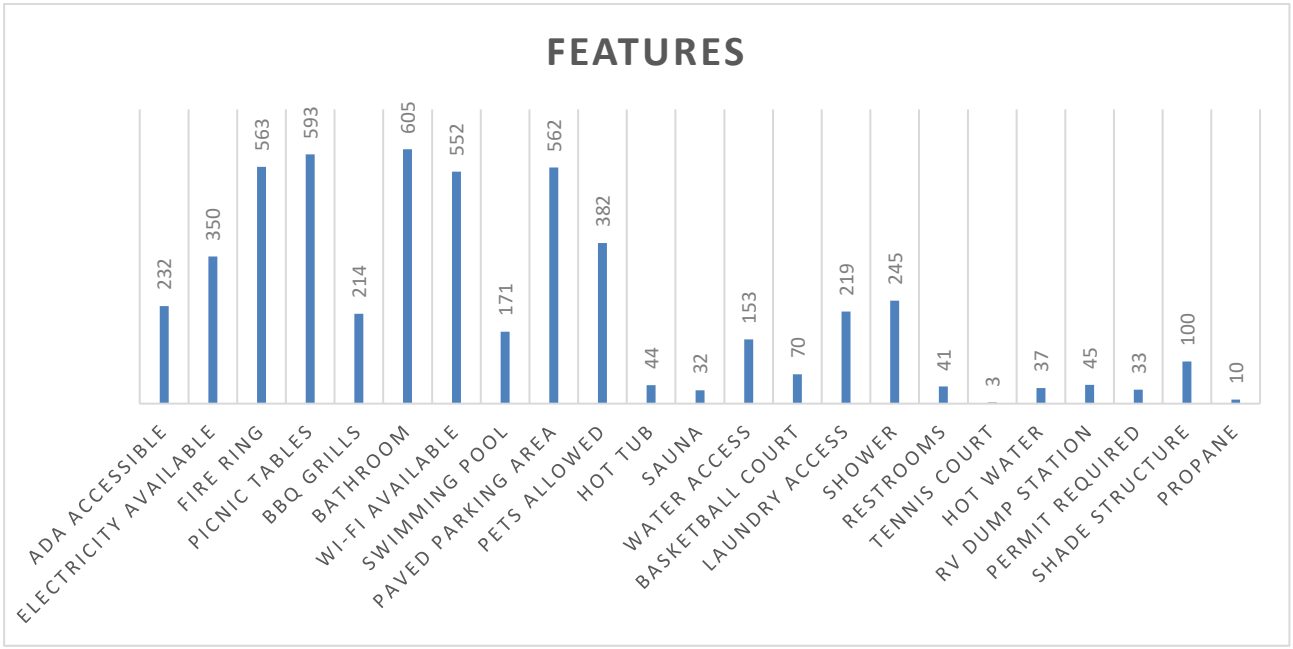

Рис. 3.3. «Особливості» локації

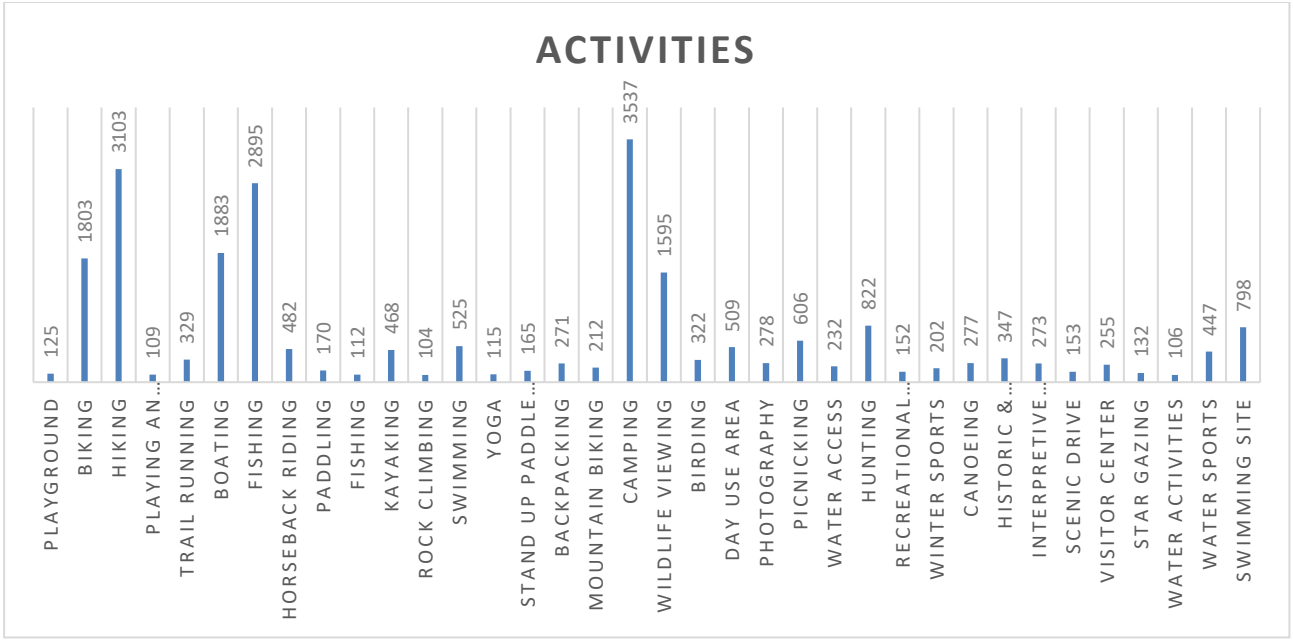

Рис. 3.4. «Активності» локації

Дивлячись на дані, які представлені на рисунках 3.3. та 3.4. можно припустити, що локації мають більш-менш рівномірний розподіл розваг та можливостей. Але не варто забувати про таку можливість, що деякі локації будуть мати всі ці дані, а деякі ні, що може призвести до тотальної переваги одних локацій над іншими.

Розглянемо параметри документа користувача (табл. 3.2). Поля «SAVED_PLACES» та «HISTORY» являються основними параметрами для формування автоматичного рейтингу локацій.

Таблиця 3.2

Параметри документа користувача

Параметр	Тип даних	Опис
ID	int	Унікальний ідентифікатор
TRAVEL_STYLE	enum	Стиль подорожі
VEHICLE_BUILD	enum	Тип транспорту
RELATIONSHIP_STATUS	enum	Сімейний стан
EMPLOYMENT_STATUS	enum	Зайнятість
LATITUDE	double	Координати: Широта
LONGITUDE	double	Координати: Довгота
FRIENDS	int[]	Список друзів
INTERESTS	string[]	Список інтересів
PETS	string[]	Список домашніх тварин
SAVED_PLACES	object[]	Список локацій, які користувач оцінив
HISTORY	object[]	Список переглянутих локацій

З рисунку 3.5. можно припустити, що статус користувачів являється звичайним для всього населення та немає ніяких аномальних викидів.

Рис. 3.5. «Зайнятість» та «Сімейний стан» користувача

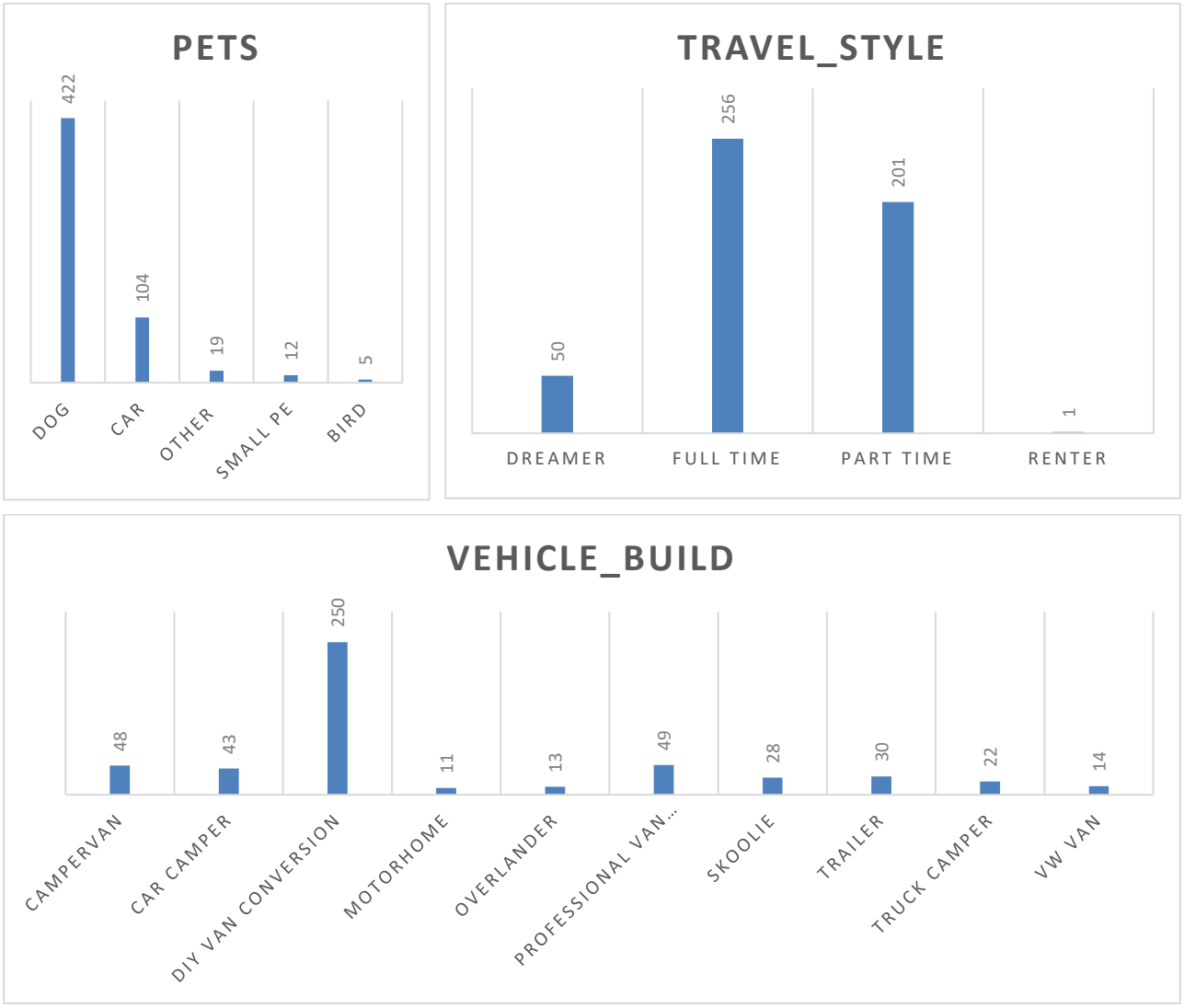

Рис. 3.6. «Домашні тварини», «Стиль подорожі» та «Тип транспорту» користувача

На рисунку 3.6. зображена достатньо цікава ситуація. Більшість користувачів воліють мати саморуч створені транспортні засоби «DIY Van Conversion»[35] та подорожувати на них весь свій час або його частину зі своїми домашніми тваринами. Даний факт, як правило свідчить про можливість користувачів відвідувати достатньо віддалені локації та залишатись там на доволі довгий період часу. В подальшому ця інформація може допомогти в вирішенні задачі стосовно виявлення оптимальної дистанції в списках рекомендованих локацій.

3.2 Опис інтелектуальної системи

Система рекомендацій генерує скомпільований список елементів, які можуть бути зацікавлені користувачем у взаємності їх поточного вибору елемента(ів). Рекомендації розширює пропозиції користувачів без будь-яких перешкод або монотонності. Інтелектуальна система складається з трьох основних блоків, які сильно зв'язані між собою та не можуть працювати один без одного.

Інтелектуальна система має наступну структуру:

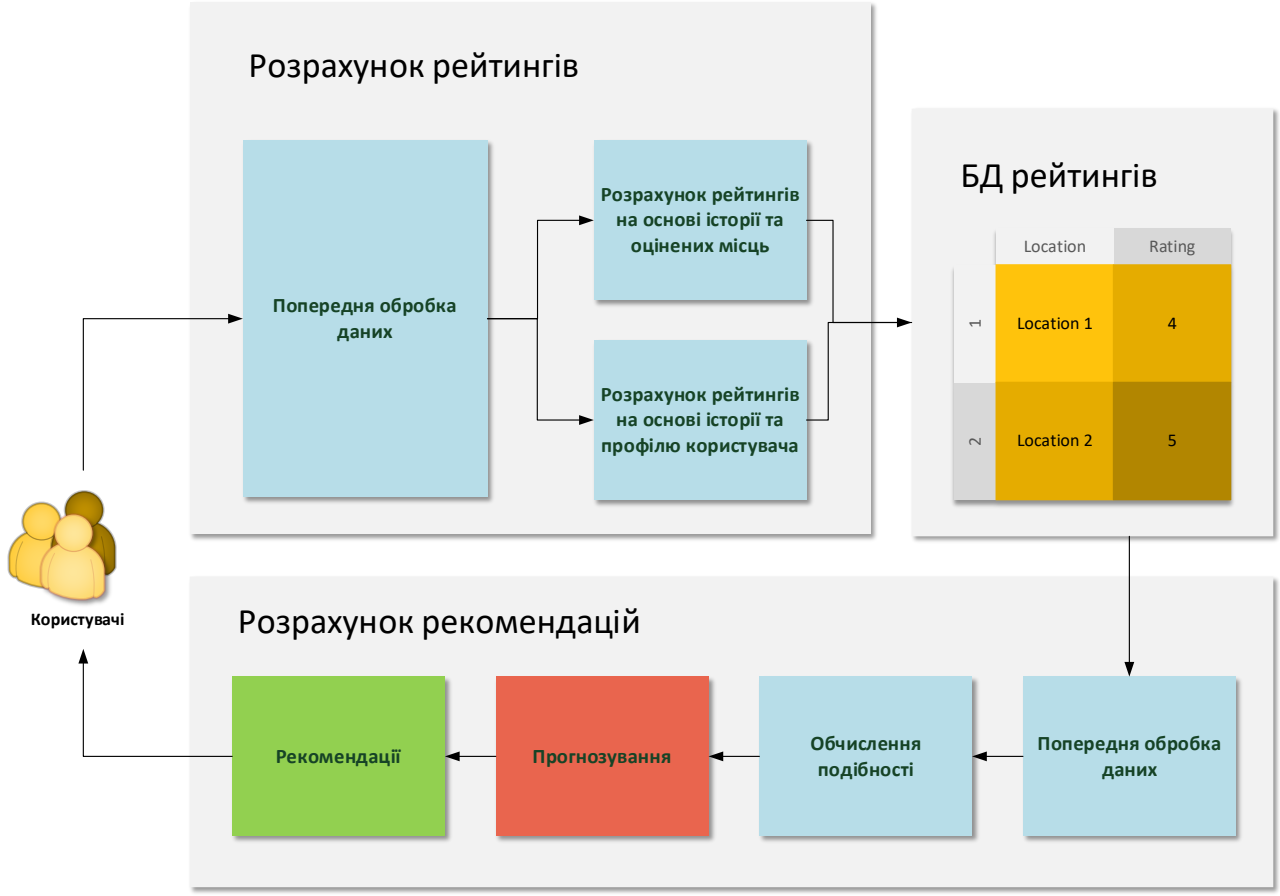

Рис. 3.7. Структура системи

Процес роботи інтелектуальної системи, показаний на діаграмі вище (рис. 3.7.), показує залежності структурних блоків та користувачів між собою.

Розглянемо кожну структурну одиницю більш детально.

3.2.1 Блок попереднього розрахунку рейтингів

Перед тим, як дані потрапляють до бази даних рейтингів вони проходять так званий етап очистки даних.

Очищення даних – процес обробки вибірки інтелектуального аналізу інформації (Data Mining) з допомогою алгоритмів машинного навчання (Machine Learning). Цей етап, на якому виконується виявлення та видалення помилок та невідповідностей у даних з метою покращення якості датасета, також називається data cleaning, data cleansing або scrubbing. Некоректна, дублююча чи втрачена інформація може стати причиною неадекватної статистики та невірних висновків у контексті бізнесу. Тому очищення даних є обов'язковою процедурою Data Preparation [36].

Перелік заходів призначених для забезпечення адекватних даних перед їхньою обробкою та додавання в базу даних рейтингів:

1) перевірка на дублювання записів в базі даних;
2) наявність всіх необхідних параметрів;
3) відкидання всіх документів, які мають скарги або мають помітку про зупинення роботи;
4) фільтрація списку можливостей та розваг локацій, дозволяються лише ті, які є як мінімум у 10 інших локаціях;
5) обробка екстремальних викидів [37], тобто ми обрізаємо значення деяких параметрів на певний ліміт. Це означає, що всі значення, які були більше ліміта тепер мають значення ліміта.

Після етапу очищення даних документи потрапляють на етап автоматичного формування рейтингів.

Автоматичне формування рейтингів – процес формування рейтингів на основі відвіданих локацій або профайлу користувача. Даний підхід дозволяє формувати обмежану кількість попередньо розрахованих рейтингів для боротьби з такими проблемами, як «Розрідженість» [13] та «Холодний старт» [14].

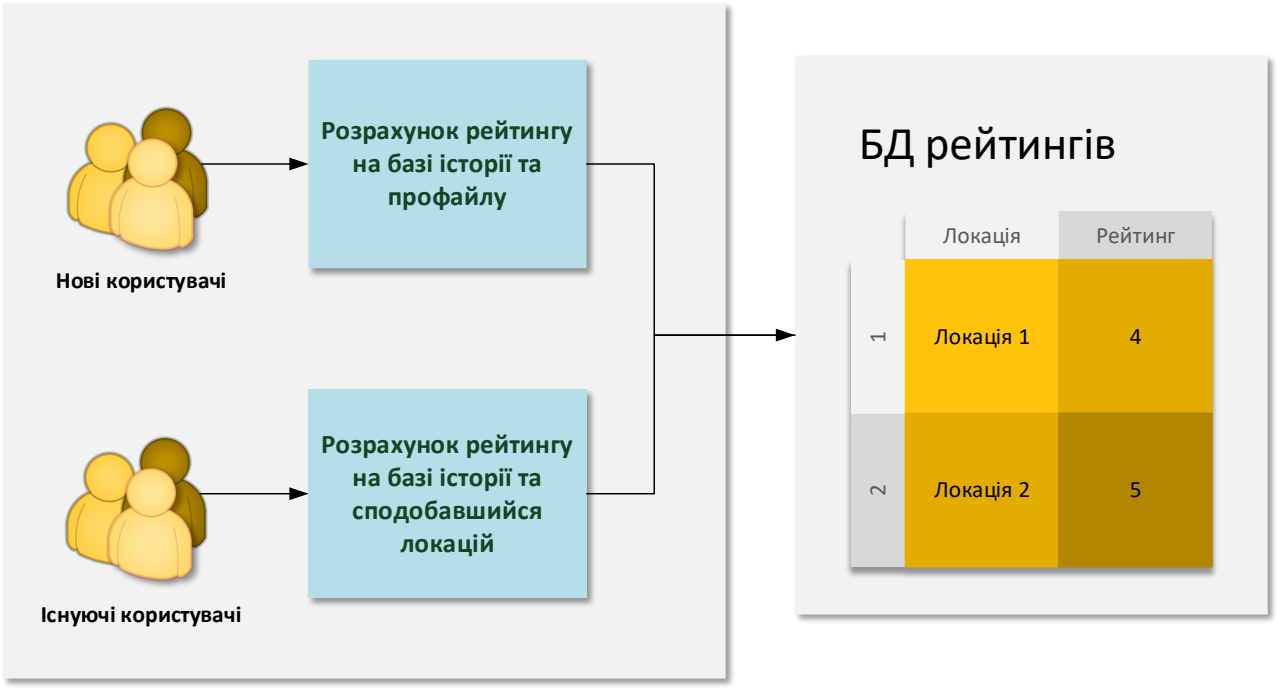

Рис. 3.8. Структура автоматичного формування рейтингів

На рисунку 3.8. зображено, як саме розподіляється алгоритм формуються рейтингів для різних груп користувачів.

Для всіх нових користувачів, які ще не залишили відгук локаціям формується обмежана кількість попередньо розрахованих рейтингів на основі профайлу користувача (інтересів користувача) та історії відвідування, ця операція дозволяє боротись з проблемою «Холодний старт» та «Розрідженість».

Для всіх існуючих користувачів, які вже мають достатньо даних формується обмежана кількість попередньо розрахованих рейтингів на основі локацій з відгуками та історії відвідування, ця операція дозволяє боротись з проблемою «Розрідженість».

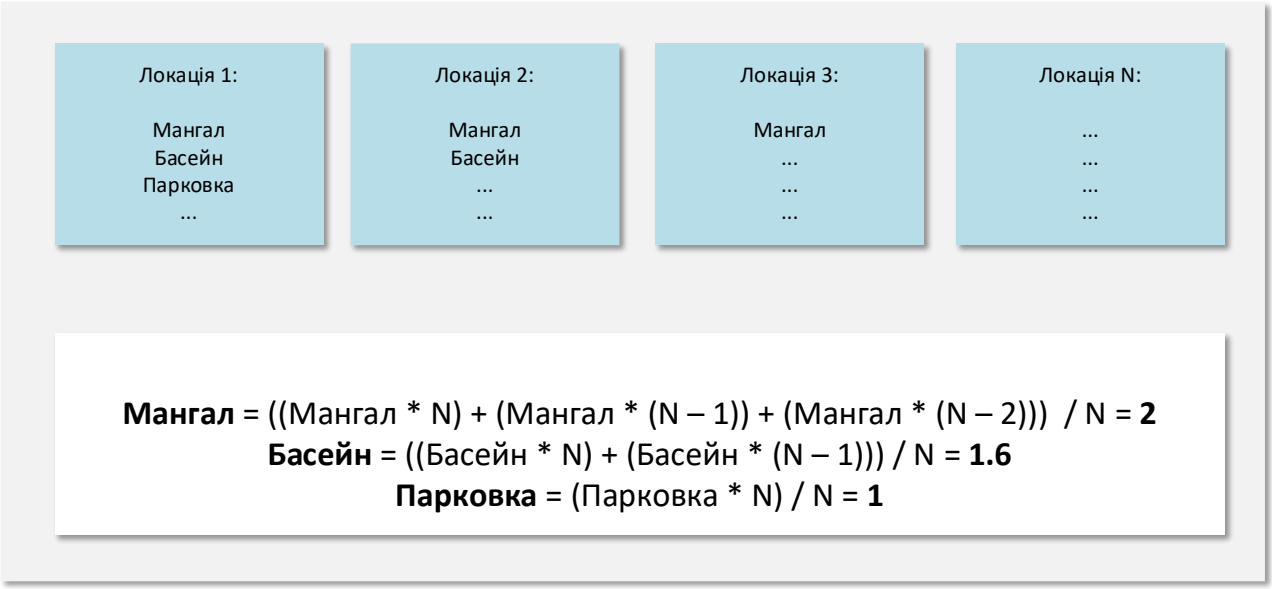

Рис. 3.9. Візуалізація розрахунку історичних записів

Формування рейтингів на основі історії відвідування локацій базується на простому принципу виявлення ключових параметрів в історії відвідування за певний період часу, які вже впливають на формування інших рейтингів. Наприклад (рис. 3.9.), ми маємо N історичних записів за останній місяць, програма розпочинає ітеративний підрахунок ваги параметрів в залежності від їх новизни.

Для боротьби з проблемами надлишок даних за конкретними користувачами ми беремо в обробку тільки ті записи, які молодші за певний період часу. Всі отримані автоматично згенеровані рейтинги нормалізуються в проміжку від 1 до 5, щоб не виникало колізії між автоматично сгенерованими рейтингами та реальними. Після проходження всіх зазначених етапів вище, рейтинги локацій записуються в базу даних.

3.2.2 Блок розрахунку рекомендацій

На рисунку 3.10. представлено список алгоритмів розбитих по схожості, які були використані в МКР.

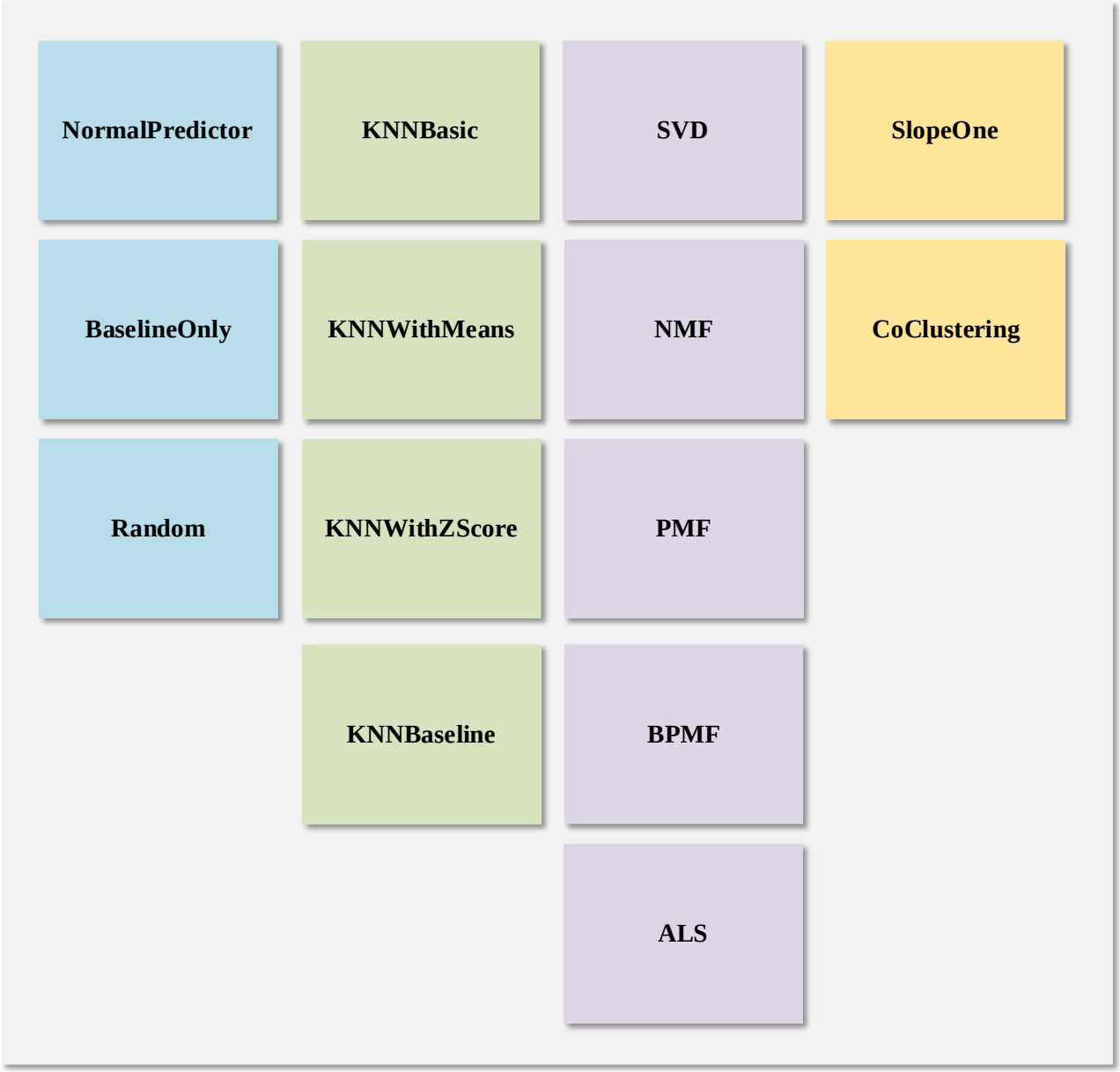

Рис. 3.10. Алгоритми рекомендацій

Група алгоритмів блакитного кольору представляє найпростіші алгоритми, які не виконують багато роботи і є найшвидшими з усіх, але все ще корисні для порівняння точності.

Група алгоритмів зеленого кольору представляє алгоритми, які безпосередньо походять від базового підходу до найближчих сусідів та активно застосовують алгоритми оцінки подібності.

Група алгоритмів фіолетового кольору представляє алгоритми матричної факторізації, що працюють шляхом розкладання матриці взаємодії користувача з елементом на добуток двох прямокутних матриць меншої розмірності.

Група алгоритмів помаранчевого кольору представляє алгоритми, які не належать до жодної з перерахованих вище груп.

Всі перераховані вище алгоритми працюють на базі пакета «Surprise»[38] за наступним принципом:

1) застосовується тренувальна вибірка розмірністю 75% та тестова 25%;

2) обраний алгоритм навчається на тренувальному набору;

3) на основі навченого алгоритма прогнозується рейтинги для тестового набору;

4) для отриманого результата розраховується параметри точності такі, як «RMSE», «MSE», «MAE», «FCP», «Precision», «Recall», «F1 Score» та час виконання.

3.2.3 Формування рекомендації

Для формування рекомендацій застосовано 4 різні групи алгоритмів розглянемо принцип роботи деяких з них.

Алгоритми найближчих сусідів

Дана група алгоритмів працює за наступним принципом. Де з самого початку розраховується подібність елементів між собою, тобто математично обчислюється наскільки подібність всі елементи між собою за допомогою таких алгоритмів подібності, як: "Косинусная подібність", "Середня квадратична різниця" та "Коефіцієнт кореляції Пірсона".

Після виконання вище наведеного кроки, безпосередньо розпочинається процес пошуку найближчих сусідів для обраного елемента, де в якості даних для пошуку найближчого сусіда береться вище розрахована оцінка подібності всіх елементи між собою та реальні рейтинги.

Отримавши вибірку найближчих сусідів виконується процес прогнозування вірогідного рейтингу елемента v для користувача u за допомогою алгоритмів "KNNBasic", "KNNWithMeans", "KNNWithZScore" та "KNNBaseline".

Перевагою застосування даної групи алгоритмів є їхня добра математична обґрунтованість, простота та швидкість роботи, недоліком – низька здатність пояснення результату та неможливість роботи при дуже великих обсягах даних.

Алгоритми матричної факторизації

Факторизація матриці працює з матрицею, де осі x та y представляють рейтинги. У наборі даних, який ми використовуватимемо, вісь x представляє ідентифікатор фільму, а вісь y представляє ідентифікатор користувача.

	Юзер 1	Юзер 2	Юзер 3	Юзер 4	Юзер 5	Юзер 6
Локація 1	0	0	**2**	**5**	**2**	0
Локація 2	0	**1**	0	**5**	0	0

Рис. 3.11. Матриця рейтингів

Значення кожного осередку представляє рейтинг локації від 1 до 5. Цілком очевидно, що це дуже розріджена матриця, як і у випадку з реальними даними. Це система спільної фільтрації, тобто вона покладається лише на оцінки інших людей, а чи не на внутрішні атрибути локацій.

Факторизація матриці виконується так (рис. 3.12.):

1) ініціалізується дві випадкові матриці X та Y з розмірами m на j та j на n так, щоб при множенні їх розмір відповідав вихідній матриці R (яка має розміри m на n);

2) перемножається матриця X на Y, щоб отримати оцінку R;

3) віднімається R від z для відомих значень R або будь-якої іншої функції втрат, щоб оцінити, наскільки далеко оцінка від реальної матриці;

4) застосовується формули градієнтного спуску, щоб відрегулювати кожне зі значень X і Y у правильному напрямку;

5) повторюється кроки 2-4 кілька разів, поки помилка не досягне розумного значення;

6) перемножується X на Y та отримується оцінка R, яка не тільки близько відповідає відомим значенням R, але також дає оцінку невідомих значень.

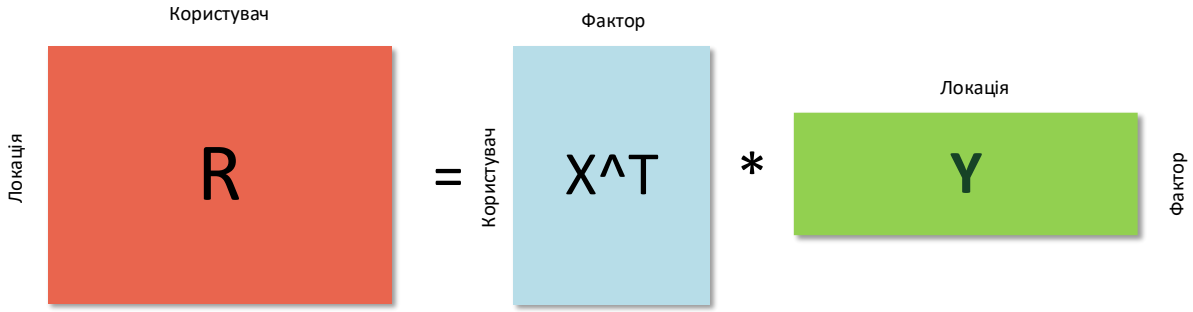

Рис. 3.12. Факторизація

Перевагою застосування даної групи алгоритмів є можливість обробка великомасштабних даних швидше та зручніше, недоліком – є те, що вони страждають від проблеми холодного запуску, оскільки не можуть надати рекомендацію новим користувачам, які не мають історії.

3.3 Аналіз результатів дослідження

Метою роботи є створення швидкої та ефективної інтелектуальної системи для підбору рекомендованих локацій за допомогою методів машинного навчання. Для того, щоб вирішити поставлену задачу необхідно перевірити якість роботи різних алгоритмів рекомендацій та обрати найкращий.

Для проведення якісного аналізу використаємо два набори даних. Перший набір буде представляти з себе вибірку даних лише з рейтингів користувачів, а другий буде складатись з рейтингів користувачів та обмеженій кількісті попередньо розрахованих рейтингів на основі профайлу користувача (інтересів користувача) та історії відвідування.

Розпочнемо наш аналіз з першої вибірки та для кожного з 14 алгоритмів проведемо пошук оптимального набору параметрів прогнозування та оцінено якість результату за допомогою різних методів оцінки точності (рис. 3.13.).

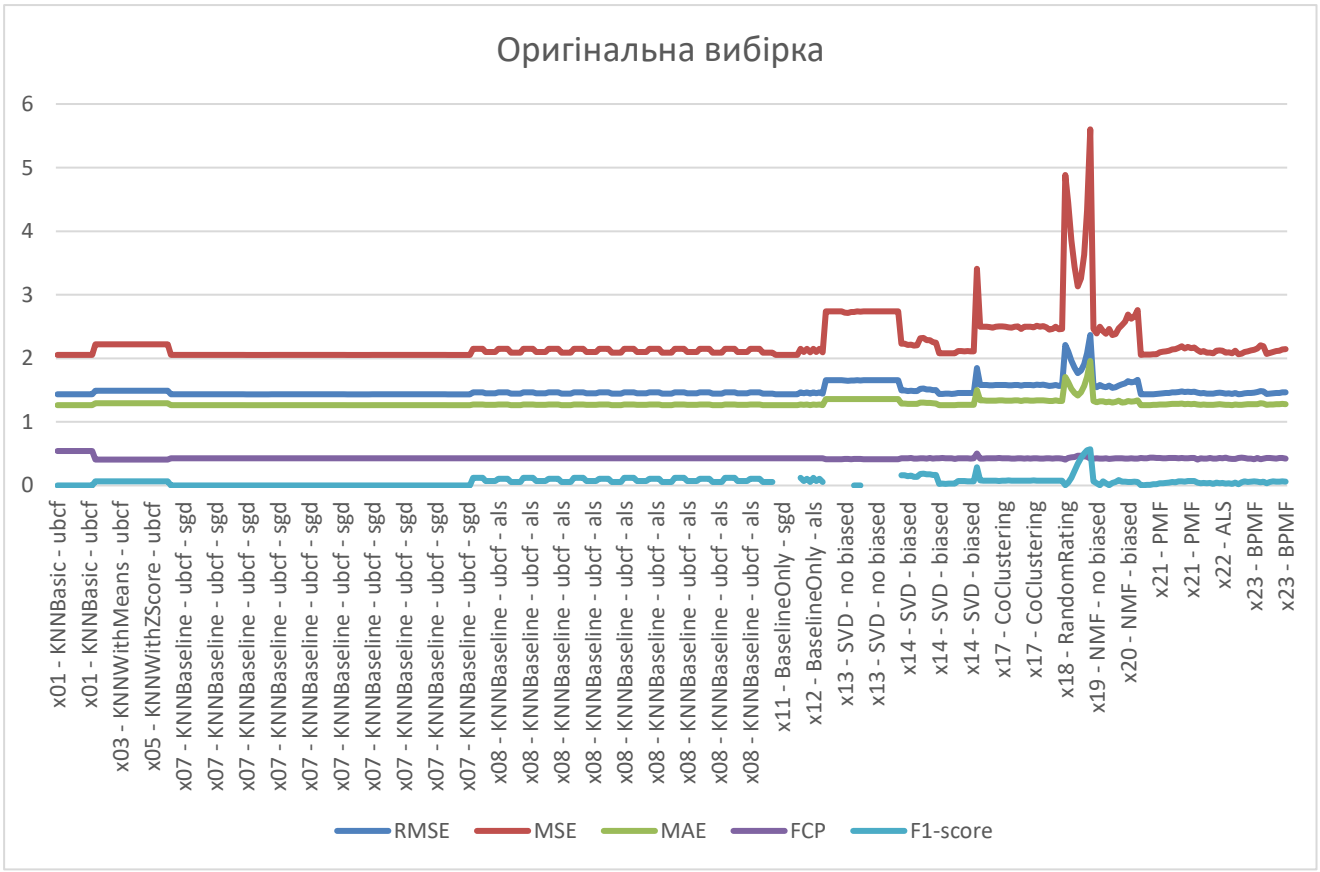

Рис. 3.13. Візуалізація RMSE & MSE & MAE & FCP & F1-score для першої вибірки

Проведено 392 експеримента з різними наборами параметрів на вибірці розмірністю 14257 елемента та затрачено 2.1 години. З рисунку 3.13. видно, що деякі групи алгоритмів майже не змінюють свою точність при різних наборах параметрів. Для більш кращого порівняння сформулюємо таблицю агрегацій алгоритмів по кожному з методів оцінки точності прогнозування (табл. 3.3).

Таблиця 3.3

Агрегація результатів алгоритмів

Name	F1-score		RMSE		MSE		MAE		FCP	
	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min
x01 - KNNBasic - ubcf	0.000468	0.000468	1.432903	1.432903	2.053212	2.053212	1.263122	1.263122	0.542353	0.542353
x03 - KNNWithMeans - ubcf	0.066157	0.066157	1.490458	1.490458	2.221464	2.221464	1.29208	1.29208	0.409014	0.409014
x05 - KNNWithZScore - ubcf	0.066157	0.066157	1.490458	1.490458	2.221464	2.221464	1.29208	1.29208	0.409014	0.409014
x07 - KNNBaseline - ubcf - sgd	0.000468	0.000468	1.432891	1.432875	2.053176	2.05313	1.263117	1.263105	0.429305	0.429305
x08 - KNNBaseline - ubcf - als	0.119429	0.053426	1.466922	1.446249	2.151859	2.091637	1.27412	1.265198	0.428952	0.427292
x11 - BaselineOnly - sgd	0	0	1.433029	1.433013	2.053573	2.053525	1.263421	1.263409	0.429305	0.429305
x12 - BaselineOnly - als	0.119429	0.052958	1.466997	1.446344	2.15208	2.091912	1.274347	1.265451	0.428952	0.427292
x13 - SVD - no biased	0.000468	0	1.655424	1.647374	2.740429	2.713841	1.36131	1.357471	0.419319	0.413206
x14 - SVD - biased	0.183663	0.026459	1.523442	1.442009	2.320877	2.079391	1.304556	1.264533	0.434189	0.420019
x15 - NormalPredictor	0.286709	0.286709	1.845889	1.845889	3.407306	3.407306	1.502463	1.502463	0.503451	0.503451
x17 - CoClustering	0.079039	0.072148	1.584909	1.566361	2.511936	2.453488	1.340226	1.328961	0.431118	0.420694

Name	F1-score		RMSE		MSE		MAE		FCP	
	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min
x18 - RandomRating	0.570035	0.007016	2.366837	1.769039	5.601917	3.1295	1.958158	1.417017	0.469893	0.4068
x19 - NMF - no biased	0.085783	0.0044 1	1.581018	1.537445	2.499619	2.363737	1.334311	1.3058 6	0.428747	0.418859
x20 - NMF - biased	0.059469	0.053878	1.660272	1.588426	2.756504	2.523098	1.340464	1.304387	0.429816	0.420547
x21 - PMF	0.07203	0.003049	1.480139	1.433704	2.190812	2.055507	1.288137	1.264198	0.436568	0.425364
x22 - ALS	0.046976	0.022791	1.458826	1.441416	2.128173	2.07768	1.276495	1.265002	0.435464	0.407615
x23 - BPMF	0.063813	0.021755	1.484932	1.436781	2.205023	2.06434	1.292914	1.266321	0.435691	0.41477
Середня оцінка	**0.075234**		**1.551061**		**2.438931**		**1.326529**		**0.43606**	
Best	**x15 - NormalPredictor**		**x07 - KNNBaseline - ubcf - sgd**		**x07 - KNNBaseline - ubcf - sgd**		**x07 - KNNBaseline - ubcf - sgd**		**x01 - KNNBasic - ubcf**	

З таблиці 3.3. та рисунку 3.13. видно, що алгоритми групи «Найближчих сусідів» не змінюйте свою точність з різними наборами параметрів. Середня оцінка помилки алгоритмів більш ніж одиниця, що обумовлено високою розрідженістю даних. Виходячи з перелічених даних можно сказати что якість результатів алгоритмів незадовільна.

Проведемо аналіз роботи алгоритмів на іншій вибірці, яка буде складатись з рейтингів користувачів та обмеженій кількісті попередньо розрахованих рейтингів на основі профайлу користувача (інтересів користувача) та історії відвідування.

Проведено 246 експеримента з різними наборами параметрів на вибірці розмірністю 298997 елемента та затрачено 10.2 години (рис. 3.14.). В ході роботи алгоритмів найближчих сусідів, які застосовували оцінки подібності

«Cosine», «Pearson» та «Pearson Baseline» трапилась помилка, що була обумовлена нестачею оперативної пам'яті, така помилка також траплялась з алгоритмами найближчих сусідів спільної фільтрація на основі елементів. Можно припустити, що алгоритми найближчих сусідів не підходять для великих вибірок даних та погано масштабуються. В подальшому будемо ігнорувати результати роботи цих алгоритмів.

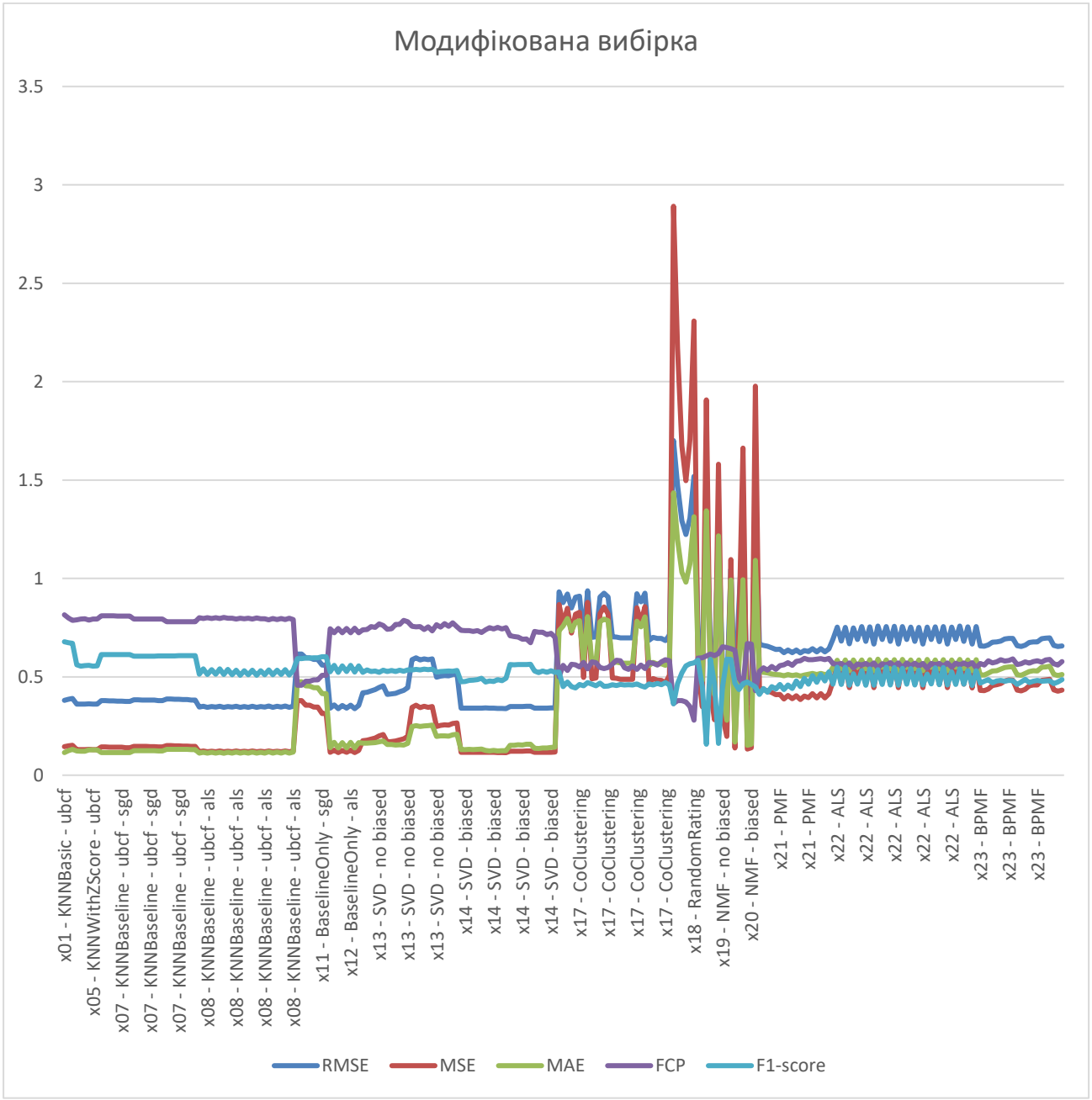

Рис. 3.14. Візуалізація RMSE & MSE & MAE & FCP & F1-score для другої вибірки

Для більш кращого порівняння сформулюємо таблицю агрегацій алгоритмів по кожному з методів оцінки точності прогнозування (табл. 3.4).

З таблиці 3.4 можно виділити той факт, що різниця між максимумом і мінімумом оцінки точності алгоритмів достатньо сильно відрізняються порівняно з першою вибіркою, що може свідчити про те, що алгоритми з даною вибіркою рейтингів чутливі до налаштувань параметрів.

Таблиця 3.4

Агрегація результатів алгоритмів

Name	F1-score		RMSE		MSE		MAE		FCP	
	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min
x01 - KNNBasic - ubcf	0.678106	0.670933	0.390022	0.380875	0.152117	0.145066	0.132061	0.114712	0.815102	0.786769
x03 - KNNWithMeans - ubcf	0.560437	0.55475	0.361101	0.360773	0.130394	0.130157	0.122469	0.122681	0.795033	0.789908
x05 - KNNWithZScore - ubcf	0.558479	0.553492	0.36202	0.3614	0.131059	0.13061	0.128667	0.128046	0.79396	0.787631
x07 - KNNBaseline - ubcf - sgd	0.613312	0.604386	0.38723	0.374271	0.149947	0.140079	0.13137	0.114822	0.809513	0.780076
x08 - KNNBaseline - ubcf - als	0.540347	0.508043	0.35112	0.34446	0.123285	0.118653	0.118511	0.111595	0.801081	0.790034
x11 - BaselineOnly - sgd	0.602013	0.59202	0.615955	0.559694	0.379401	0.313257	0.4744	0.414906	0.507904	0.457056
x12 - BaselineOnly - als	0.558363	0.523097	0.356479	0.33831	0.127077	0.114454	0.166606	0.137979	0.74639	0.723129
x13 - SVD - no biased	0.539196	0.522548	0.5961	0.410276	0.355335	0.168327	0.254401	0.15268	0.786929	0.732906
x14 - SVD - biased	0.563919	0.472937	0.350893	0.338221	0.123126	0.114393	0.157707	0.123174	0.751295	0.673436
x15 - NormalPredictor	0.523488	0.523488	0.931123	0.931123	0.86699	0.86699	0.735035	0.735035	0.499072	0.499072
x17 - CoClustering	0.473302	0.44369	0.937084	0.684664	0.878126	0.468765	0.805234	0.556423	0.586885	0.532715

Name	F1-score		RMSE		MSE		MAE		FCP	
	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min
x18 - RandomRating	0.570662	0.363025	1.700351	1.223394	2.891192	1.496694	1.435036	0.981376	0.379139	0.279441
x19 - NMF - no biased	0.589511	0.157214	1.381195	0.443466	1.907701	0.196662	1.34376	0.279256	0.652404	0.594705
x20 - NMF - biased	0.473377	0.435245	1.406122	0.364298	1.97718	0.132713	1.091325	0.151994	0.669004	0.429418
x21 - PMF	0.525673	0.411087	0.664305	0.620599	0.441301	0.385143	0.525744	0.503978	0.593798	0.527884
x22 - ALS	0.548383	0.455378	0.758099	0.66621	0.574715	0.443835	0.590049	0.530703	0.571935	0.542651
x23 - BPMF	0.487562	0.461309	0.697465	0.653486	0.486457	0.427044	0.553498	0.505827	0.590577	0.559234
Середня оцінка	**0.519376**		**0.626535**		**0.51436**		**0.424443**		**0.642238**	
Best	**x19 - NMF - no biased**		**x14 - SVD - biased**		**x14 - SVD - biased**		**x14 - SVD - biased**		**x13 - SVD - no biased**	

Середня оцінка помилки алгоритмів менше ніж одиниця, що обумовлено достатньою кількістю рейтингів локаціям. З перелічених вище фактів, можно припустити, що подібні показники є результатом роботи блоку попереднього розрахунку рейтингів на основі профайлу користувача (інтересів користувача) та історії відвідування.

Переваги застосування даного способу формування рейтингів заключається в тому, що система здатна формувати рекомендації новим користувачам та гнучко оновлювати список рекомендацій базуючись на історії користувача.

Висновки до розділу 3

В даному розділу був проведений докладний опис моделі даних користувача та локацій, описано схему роботи інтелектуальної системи та проведено аналіз результатів дослідження.

Найбільш оптимальним алгоритмом та набором параметрів для вибірки з попередньо розрахованими рейтингами буде «SVD» з наступними параметрами:

1) з використанням базової лінії «SGD»;
2) з швидкістю факторів «50»;
3) з швидкістю навчання для всіх параметрів «0.02»;
4) параметр регуляризації функції вартості «0.005»;
5) кількість ітерацій процедури SGD «25».

ВИСНОВКИ

В результаті виконання магістерської роботи було реалізована швидка та ефективна інтелектуальна системи для підбору рекомендованих локацій за допомогою методів машинного навчання.

Зазначену мету досягнуто завдяки виконання наступних завдань:

– проаналізувати схожі програмні продукти для підбору рекомендованих локацій;

– розробити інтелектуальне та програмне забезпечення системи для збору та підготовки даних;

– розробити інтелектуальну систему для підбору рекомендованих локацій;

– протестовано точність системи.

Проведений докладний опис предметної сфери, побудовані функціональні моделі інтелектуальної системи для визначення локацій методом машинного навчання. Проаналізовано публікації інших інтелектуальних систем. Проведений докладний опис алгоритмів прогнозування, оцінки подібності, оцінки точності прогнозної моделі та прогнозів. Розкрито загальні поняття рекомендаційних систем та описано проблеми спільної фільтрації.

Проаналізовано роботу алгоритмів, які безпосередньо походять від базового підходу до найближчих сусідів та активно застосовують алгоритми оцінки подібності, алгоритми матричної факторізації, що працюють шляхом розкладання матриці взаємодії користувача з елементом на добуток двох прямокутних матриць меншої розмірності та інші.

Перевірено якість роботи системи при використанні оригінальної вибірки та вибірки, яка складається з автоматично сформованих рейтингів на основі відвіданих локацій або профайлу користувача. Даний підхід дозволяє формувати обмежану кількість попередньо розрахованих рейтингів для боротьби з такими проблемами, як «Розрідженість» та «Холодний старт».

Проведено докладний опис моделі даних користувача та локацій, описано схему роботи інтелектуальної системи та проведено аналіз результатів дослідження. Найбільш оптимальним алгоритмом для вибірки з попередньо розрахованими рейтингами є «SVD».

Практична значимість розробленої інтелектуальної системи полягає в тому, щоб покращити досвід користувачів в пошуку локацій, які можуть їх зацікавити. Більшість користувачів знають про існування індивідуальних рекомендацій на великих інтернет-платформах, однак важливість і домінування цієї технології, можливо, більш важливі, ніж багато хто думає. Оскільки добре продумана система рекомендацій звільняє користувача від фільтрації великої кількості даних, що в свою чергу дозволяє користувачеві отримувати якісний контент за короткий проміжок часу.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Netflix: веб-сайт. URL: https://www.netflix.com/ (дата звернення: 01.01.2022).

2. YouTube: веб-сайт. URL: https://www.youtube.com/ (дата звернення: 01.01.2022).

3. Spotify: веб-сайт. URL: https://www.spotify.com/ (дата звернення: 01.01.2022).

4. P. Valdiviezo-Diaz, F. Ortega, E. Cobos and R. Lara-Cabrera, "A Collaborative Filtering Approach Based on Naïve Bayes Classifier," in IEEE Access, vol. 7, pp. 108581-108592, 2019, doi: 10.1109/ACCESS.2019.2933048.

5. Y. Zhang, J. Wang and J. Luo, "Knowledge Graph Embedding Based Collaborative Filtering," in IEEE Access, vol. 8, pp. 134553-134562, 2020, doi: 10.1109/ACCESS.2020.3011105.

6. Y. Guo and Z. Yan, "Recommended System: Attentive Neural Collaborative Filtering," in IEEE Access, vol. 8, pp. 125953-125960, 2020, doi: 10.1109/ACCESS.2020.3006141.

7. X. Yu, F. Jiang, J. Du and D. Gong, "A User-Based Cross Domain Collaborative Filtering Algorithm Based on a Linear Decomposition Model," in IEEE Access, vol. 5, pp. 27582-27589, 2017, doi: 10.1109/ACCESS.2017.2774442.

8. T. Badriyah, S. Azvy, W. Yuwono and I. Syarif, "Recommendation system for property search using content based filtering method," 2018 International Conference on Information and Communications Technology (ICOIACT), 2018, pp. 25-29, doi: 10.1109/ICOIACT.2018.8350801.

9. Introduction to recommender systems. URL: https://towardsdatascience.com/introduction-to-recommender-systems-6c66cf15ada. (дата звернення: 02.01.2022).

10. What Content-Based Filtering is and Why You Should Use It. URL: https://www.upwork.com/resources/what-is-content-based-filtering. (дата звернення: 02.01.2022).

11. Collaborative Filtering In Recommender Systems: Learn All You Need To Know. URL: https://www.iteratorshq.com/blog/collaborative-filtering-in-recommender-systems/. (дата звернення: 02.01.2022).

12. Recommender System — Matrix Factorization. URL: https://towardsdatascience.com/recommendation-system-matrix-factorization-d61978660b4b. (дата звернення: 02.01.2022).

13. Sparse Matrix. URL: https://www.sciencedirect.com/topics/mathematics/sparse-matrix/. (дата звернення: 02.01.2022).

14. Solving Cold User problem for Recommendation system using Multi-Armed Bandit. URL: https://towardsdatascience.com/solving-cold-user-problem-for-recommendation-system-using-multi-armed-bandit-d36e42fe8d44/. (дата звернення: 02.01.2022).

15. Basic algorithms. URL: https://surprise.readthedocs.io/en/stable/basic_algorithms.html (дата звернення: 02.01.2022).

16. SVD. URL: https://surprise.readthedocs.io/en/stable/matrix_factorization.html#unbiased-note (дата звернення: 02.01.2022).

17. Tang, Bing & Kang, Linyao & Zhang, Li & Guo, Feiyan & He, Haiwu. (2021). Collaborative Filtering Recommendation Using Nonnegative Matrix Factorization in GPU-Accelerated Spark Platform. Scientific Programming. 2021. 1-15. 10.1155/2021/8841133.

18. Salakhutdinov, R. & Mnih, A.. (2008). Probabilistic matrix factorization. Neural Information Processing Systems(NIPS08). 1257-1264.

19. Salakhutdinov, Ruslan & Mnih, Andriy. (2008). Bayesian probabilistic matrix factorization using Markov chain Monte Carlo. Proceedings of the 25th International Conference on Machine Learning. 25. 880-887. 10.1145/1390156.1390267.

20. Zhou, Yunhong & Wilkinson, Dennis & Schreiber, Robert & Pan, Rong. (2008). Large-Scale Parallel Collaborative Filtering for the Netflix Prize. 337-348. 10.1007/978-3-540-68880-8_32.

21. Lemire, Daniel & Maclachlan, Anna. (2007). Slope One Predictors for Online Rating-Based Collaborative Filtering. Proceedings of the 2005 SIAM International Conference on Data Mining, SDM 2005. 5. 10.1137/1.9781611972757.43.

22. George, Thomas & Merugu, Srujana. (2005). A scalable collaborative filtering framework based on co-clustering. 4 pp.-. 10.1109/ICDM.2005.14.

23. Rahutomo, Faisal & Kitasuka, Teruaki & Aritsugi, Masayoshi. (2012). Semantic Cosine Similarity.

24. Liu, Haifeng & Hu, Zheng & Mian, Ahmad & Tian, Hui & Zhu, Xuzhen. (2014). A new user similarity model to improve the accuracy of collaborative filtering. Knowledge-Based Systems. 56. 156–166. 10.1016/j.knosys.2013.11.006.

25. Benesty, Jacob & Chen, Jingdong & Huang, Yiteng & Cohen, Israel. (2009). Pearson Correlation Coefficient. 10.1007/978-3-642-00296-0_5.

26. Pearson-baseline correlation coefficient. URL: https://surprise.readthedocs.io/en/stable/similarities.html#surprise.similarities.pearson_baseline (дата звернення: 02.01.2022).

27. Hyndman, Rob & Koehler, Anne. (2006). Another look at measures of forecast accuracy. International Journal of Forecasting. 22. 679-688. 10.1016/j.ijforecast.2006.03.001.

28. Alexander, S.. (1986). The Mean Square Error (MSE) Performance Criteria. 10.1007/978-1-4612-4978-8_2.

29. Chai, Tianfeng & Draxler, R.. (2014). Root mean square error (RMSE) or mean absolute error (MAE)?. Geosci. Model Dev.. 7. 10.5194/gmdd-7-1525-2014.

30. Koren, Yehuda & Sill, Joseph. (2013). Collaborative filtering on ordinal user feedback. 3022-3026.

31. Ting, Kai. (2011). Precision and Recall. 10.1007/978-0-387-30164-8_652.

32. Srivastava, Saurabh & Singh, Girdhari. (2016). F1 Score Analysis of Search Engines. S.K.I.T Research Journal. 6. 1-6.

33. Recreation Information Database. URL: https://ridb.recreation.gov/docs (дата звернення: 04.01.2022).

34. The most comprehensive guide to Federal, State, Provincial and Local campgrounds. URL: http://www.uscampgrounds.info (дата звернення: 04.01.2022).

35. Build Journal. URL: https://faroutride.com/build-journal/ (дата звернення: 05.01.2022).

36. The Ultimate Guide to Data Cleaning. URL: https://towardsdatascience.com/the-ultimate-guide-to-data-cleaning-3969843991d4 (дата звернення: 06.01.2022).

37. Handling extreme outliers. URL: https://developers.google.com/machine-learning/crash-course/representation/cleaning-data#handling-extreme-outliers (дата звернення: 06.01.2022).

38. Welcome to Surprise' documentation!. URL: https://surprise.readthedocs.io/en/stable/index.html (дата звернення: 06.01.2022).

39. Factor in the Neighbors: Scalable andAccurate Collaborative Filtering. URL: https://courses.ischool.berkeley.edu/i290-dm/s11/SECURE/a1-koren.pdf (дата звернення: 07.01.2022).

40. Bottou, Léon & Bousquet, Olivier. (2007). The Tradeoffs of Large Scale Learning.. Optimization for Machine Learning. 20.

41. Piryonesi, S Madeh & El-Diraby, Tamer. (2020). Role of Data Analytics in Infrastructure Asset Management: Overcoming Data Size and Quality Problems. Journal of Transportation Engineering, Part B: Pavements. 146. 04020022. 10.1061/JPEODX.0000175.

42. Санітарні норми мікроклімату виробничих приміщень: ДСН 3.3.6.042–99 [Чинний від 1999-12-01]. Київ : Мінрегіонбуд України, 1999. 12 с.

43. Про боротьбу з тероризмом : Закон України від 20.03.2003 № 638-IV // База даних «Законодавство України» / Верховна Рада України. URL: https://zakon.rada.gov.ua/go/638-15 (дата звернення: 29.01.2022).

44. Як поводитися під час терористичної загрози? URL: https://uman-rda.gov.ua/yak-povoditisya-pid-chas-teroristichnoi-zagrozi-13-27-11-25-10-2017/ (дата звернення: 29.01.2022).

ДОДАТОК А

Код програмного забезпечення

```
class MyRecommendation:
  def testRecommendations(self):
    reader = surprise.reader.Reader(
      line_format='user item rating', sep='\t')
    data = surprise.Dataset.load_from_folds(
      [('../report/train.dat', '../report/test.dat')], reader)
    totalrecommendations = []
    def runTest(trainset, testset, alg, name, params, showTop=False):
      start_time = time.time()
      alg.fit(trainset)
      predictions = alg.test(testset)
      print(name)
      print(params)
      accuracy = hlp.calcAccuracy(start_time, predictions)
      print('RMSE: \t{}'.format(accuracy['rmse']))
      print('MSE: \t{}'.format(accuracy['mse']))
      print('MAE: \t{}'.format(accuracy['mae']))
      print('FCP: \t{}'.format(accuracy['fcp']))
      print('Precisions: \t{}'.format(accuracy['precisions']))
      print('Recalls: \t{}'.format(accuracy['recalls']))
      print('F1-score: \t{}'.format(accuracy['f1-score']))
      totalrecommendations.append({
        'name': name,
        'params': params,
        'accuracy': accuracy
      })

    pkf = surprise.model_selection.PredefinedKFold()
    methods = [
      {
        'class': surprise.KNNBasic,
        'name': "x01 - KNNBasic - ubcf",
        'gridParams': {
          'k': [10, 20, 30],
          'min_k': [1],
          'sim_options': {  'user_based': [True],   'name': ['pearson_baseline', 'cosine', 'msd', 'pearson']  }
        }
      },
      {
        'class': surprise.KNNBasic,
        'name': "x02 - KNNBasic - ibcf",
        'gridParams': {
          'k': [10, 20, 30],
          'min_k': [1],
          'sim_options': {  'user_based': [False],   'name': ['pearson_baseline', 'cosine', 'msd', 'pearson']  }
        }
      },
      {
        'class': surprise.KNNWithMeans,
        'name': "x03 - KNNWithMeans - ubcf",
        'gridParams': {
          'k': [10, 20, 30],
          'min_k': [1],
          'sim_options': {  'user_based': [True],'name': ['pearson_baseline', 'cosine', 'msd', 'pearson']}
        }
      },
      {
        'class': surprise.KNNWithMeans,
        'name': "x04 - KNNWithMeans - ibcf",
        'gridParams': {
          'k': [10, 20, 30],
          'min_k': [1],
          'sim_options': {    'user_based': [False],  'name': ['pearson_baseline', 'cosine', 'msd', 'pearson']  }
        }
      }, {
        'class': surprise.KNNWithZScore,
        'name': "x05 - KNNWithZScore - ubcf",
        'gridParams': {
```

```
      'k': [10, 20, 30],
      'min_k': [1],
      'sim_options': {  'user_based': [True],  'name': ['pearson_baseline', 'cosine', 'msd', 'pearson']  }
    }
  }, {
    'class': surprise.KNNWithZScore,
    'name': "x06 - KNNWithZScore - ibcf",
    'gridParams': {
      'k': [10, 20, 30],
      'min_k': [1],
      'sim_options': {  'user_based': [False],'name': ['pearson_baseline', 'cosine', 'msd', 'pearson']}
    }
  }, {
    'class': surprise.KNNBaseline,
    'name': "x07 - KNNBaseline - ubcf - sgd",
    'gridParams': {
      'k': [10, 20, 30],
      'min_k': [1],
      'sim_options': {  'user_based': [True],'name': ['pearson_baseline', 'cosine', 'msd', 'pearson']},
      'bsl_options': {  'method': ['sgd'], 'learning_rate': [.00005, .0001],  'reg': [.02, 0.005],   'n_epochs': [15, 25],   }
    }
  }, {
    'class': surprise.KNNBaseline,
    'name': "x08 - KNNBaseline - ubcf - als",
    'gridParams': {
      'k': [10, 20, 30],
      'min_k': [1],
      'sim_options': {  'user_based': [True],'name': ['pearson_baseline', 'cosine', 'msd', 'pearson']},
      'bsl_options': {  'method': ['als'],  'n_epochs': [15, 25],  'reg_u': [4, 9],'reg_i': [3, 5]}
    }
  }, {
    'class': surprise.KNNBaseline,
    'name': "x09 - KNNBaseline - ibcf - sgd",
    'gridParams': {
      'k': [10, 20, 30],
      'min_k': [1],
      'sim_options': {  'user_based': [False],  'name': ['pearson_baseline', 'cosine', 'msd', 'pearson']  },
      'bsl_options': {  'method': ['sgd'],'learning_rate': [.00005, .0001],  'reg': [.02, 0.005],   'n_epochs': [15, 25],}
    }
  }, {
    'class': surprise.KNNBaseline,
    'name': "x10 - KNNBaseline - ibcf - als",
    'gridParams': {
      'k': [10, 20, 30],
      'min_k': [1],
      'sim_options': {  'user_based': [False],'name': ['pearson_baseline', 'cosine', 'msd', 'pearson']},
      'bsl_options': {  'method': ['als'],  'n_epochs': [15, 25],  'reg_u': [4, 9],  'reg_i': [3, 5]}
    }
  },  {
    'class': surprise.BaselineOnly,
    'name': "x11 - BaselineOnly - sgd",
    'gridParams': {
      'bsl_options': {  'method': ['sgd'],  'learning_rate': [.00005, .0001],  'reg': [.02, 0.005],  'n_epochs': [15, 25],   }
    }
  },  {
    'class': surprise.BaselineOnly,
    'name': "x12 - BaselineOnly - als",
    'gridParams': {
      'bsl_options': {'method': ['als'],   'n_epochs': [15, 25],'reg_u': [4, 9],  'reg_i': [3, 5]  }
    }
  }, {
    'class': surprise.SVD,
    'name': "x13 - SVD - no biased",
    'gridParams': {   'biased': [False],  'n_factors': [10, 25, 50],  'n_epochs': [15, 25],  'lr_all': [.02, 0.005],   'reg_all': [.02, 0.005],  }
  }, {
    'class': surprise.SVD,
    'name': "x14 - SVD - biased",
    'gridParams': {   'biased': [True],   'n_factors': [10, 25, 50],    'n_epochs': [15, 25],    'lr_all': [.02, 0.005],   'reg_all': [.02, 0.005],   }
  }, {
    'class': surprise.NormalPredictor,
    'name': "x15 - NormalPredictor",
    'gridParams': {}
  },  {
    'class': surprise.SlopeOne,
```

```
        'name': "x16 - SlopeOne",
        'gridParams': {}
      }, {
        'class': surprise.CoClustering,
        'name': "x17 - CoClustering",
        'gridParams': {'n_epochs': [15, 25, 50],   'n_cltr_i': [2, 3, 4],  'n_cltr_u': [2, 3, 4]   }
      },  {
        'class': surprise.NMF,
        'name': "x19 - NMF - no biased",
        'gridParams': {   'biased': [False],  'n_factors': [10, 15, 25],   'n_epochs': [15, 25, 50],
          'reg_pu': [.02],    'reg_qi': [.02],  'reg_bu': [.02],  'reg_bi': [.02],'lr_bu': [.02],   'lr_bi': [.02],
        }
      }, {
        'class': surprise.NMF,
        'name': "x20 - NMF - biased",
        'gridParams': {
          'biased': [True],
          'n_factors': [10, 15, 25],
          'n_epochs': [15, 25],
          'reg_pu': [.02],
          'reg_qi': [.02],
          'reg_bu': [.02],
          'reg_bi': [.02],
          'lr_bu': [.02],
          'lr_bi': [.02],
        }
      }, {
        'class': customRecomendation.PMF,
        'name': "x21 - PMF",
        'gridParams': {
          'n_feature': [10, 15, 25],
          'n_epochs': [15, 25, 50],
          'batch_size': [100000],
          'epsilon': [50.0],
          'momentum': [.8,],
          'reg': [0.06, 0.001],
          'converge': [0.00001],
        }
      }, {
        'class': customRecomendation.ALS,
        'name': "x22 - ALS",
        'gridParams': {  'n_feature': [10, 15, 25],  'n_epochs': [15, 25, 50],'converge': [0.0001, 0.00001],   'reg': [0.06, 0.001],}
      }, {
        'class': customRecomendation.BPMF,
        'name': "x23 - BPMF",
        'gridParams': {
          'n_feature': [10, 15, 25],
          'n_epochs': [15, 25, 50],
          'converge': [0.0001, 0.00001],
          'mu0_user': [0],
          'mu0_item': [0],
          'beta': [3.0, 2.0],
          'beta_user': [3.0, 2.0],
          'beta_item': [3.0, 2.0],
        }
      },
    ]
    df = pd.read_csv('../report/recommendations-copy.txt', sep="\t")
    for trainset, testset in pkf.split(data):
      for method in methods:
        gridParams = hlp.genParams(method['gridParams'])
        for param in gridParams:
          exists = df.loc[(df['Name'] == method['name']) & (df['Params'] == str(param))]
          if len(exists) == 0:
            try:
              runTest(trainset, testset,  method['class'](**param), method['name'], str(param),  )
            except:
              pass
          else:
            print('pass: {}'.format(i))
          self.writeToFile(totalrecommendations)
    self.writeToFile(totalrecommendations)
    self.analyzeRecommendationResults()
```

ДОДАТОК Б

Матеріали апробації роботи

Міністерство освіти і науки України
Чорноморський національний університет
імені Петра Могили

«Інформаційні технології та інженерія»

Всеукраїнська науково-практична конференція молодих вчених, аспірантів і студентів

ТЕЗИ

9–11 лютого 2022 року

Миколаїв – 2022

Машинне навчання та штучний інтелект

Баришніков В. О., Мєщанінов О. П. Послідовність розробки системи інтелектуального аналізу даних розумного будинку 38

3

Борисов М. В., Гожий О. П. Використання методів машинного навчання для прогнозування чистої вихідної електроенергії електростанції комбінованого циклу 40

Івченко І. О., Калініна І. О. «Застосування методів машинного навчання для вирішення задачі біологічної класифікації 43

Костиря М. А., Кондратенко Ю. П. Дослідження впливу архітектур згорткових нейронних мереж на ефективність сегментації об'єктів.... 44

Малімон О. О., Сіденко Є. В. Інтелектуальний аналіз та класифікація текстів з використанням технологій штучного інтелекту 47

Нечахін В. В. Застосування нейромережевої архітектури LSTM в системі керування сонячною електростанцією..... 50

Петрович В. І., Кондратенко Г. В. Дослідження методів машинного зору для автоматизованого діагностування хвороб шкіри..... 52

Под'ячев А. Д., Гожий О. П. Використання нечіткої логіки в ігровому штучному інтелекті 55

Попель О. О., Гожий О. П. Хмарні обчислення задач машинного навчання та штучного інтелекту з використанням інструментів Infrastructure as Code..... 57

Савчук О. А., Кондратенко Ю. П. Діагностування COVID-19 з використанням методів штучного інтелекту..... 60

Скакун Є. І., Гожий О. П. Інтелектуальна система для визначення локацій методом машинного навчання..... 63

Скубак М. Д., Калініна І. О. Інтелектуальна система аналізу контенту музичного вебсерверу для Андроїд-застосунку..... 65

Рецензія

на магістерську кваліфікаційну роботу студента групи 601 ЧНУ імені Петра Могили

Скакуна Євгенія Ігоровича

«Інтелектуальна система для визначення локацій методом машинного навчання»

Магістерська кваліфікаційна робота Скакуна Є. І., яку подано на рецензію, виконана у відповідності до завдання, в повному обсязі у встановлений термін.

Актуальність теми магістерської кваліфікаційної роботи полягає в тому, щоб покращити досвід користувачів в пошуку локацій, які можуть їх зацікавити. Добре продумана система рекомендацій звільняє користувача від фільтрації великої кількості даних, що в свою чергу дозволяє користувачеві отримувати якісний контент за короткий проміжок часу.

МКР складається з фахової частини, методичної та спеціальної частини з охорони праці.

У роботі проаналізовано схожі програмні продукти систем рекомендацій, проведено докладний опис моделі даних користувача та локацій, описано схему роботи інтелектуальної системи та проведено аналіз результатів дослідження.

Проаналізовано роботу алгоритмів, які безпосередньо походять від базового підходу до найближчих сусідів та активно застосовують алгоритми оцінки подібності, алгоритми матричної факторізації, що працюють шляхом розкладання матриці взаємодії користувача з елементом на добуток двох прямокутних матриць меншої розмірності та інші.

Зауваження до роботи: наявні несуттєві неточності у оформленні, а також виникає питання, з яким мінімальним розміром вибірки рейтингів може коректно працювати система.

Відзначений недолік не зменшує в цілому позитивне враження від роботи, яку виконано на високому рівні.

Вважаю, що робота задовольняє вимогам, які пред’являються до МКР та може бути оцінена на «відмінно», а її автор, Скакун Є. І., заслуговує присвоєння освітньої кваліфікації «Магістр з комп’ютерних наук».

Рецензент,
канд. техн. наук, доц., зав. каф. КІ
ЧНУ імені Петра Могили Я. М. Крайник

ВІДГУК
на магістерську роботу студента групи 601 ЧНУ імені Петра Могили
Скакуна Євгенія Ігоровича
«Інтелектуальна система для визначення локацій методом машинного навчання»

Зі збільшенням кількості локації в системі на користувача збільшується інформаційний тиск, що не дуже гарно сприяє на користувацький досвід та відгуки про систему. Для боротьби з подібними проблемами слід застосувати системи рекомендацій. Системи рекомендацій допомагають користувачам отримувати персоналізовані рекомендації, допомагають користувачам приймати правильні рішення під час серфінгу, підвищувати продажі та перевизначати досвід перегляду локацій, утримувати користувачів, покращувати їхній досвід бронювання.

Результатом виконання магістерської роботи є розробка інтелектуальної система для підбору локацій за допомогою методів машинного навчання.

Під час виконання магістерської роботи студент Скакун Є. І. проявив себе як самостійний, грамотний аналітик та кваліфікований програміст, який володіє методами моделювання та технічного проектування інформаційних систем, сучасними технологіями створення програмних продуктів, і може довести поставлену задачу до практичної реалізації.

Отримані в результаті виконання магістерської роботи рішення мають практичне значення та сприяють покращити досвід користувачів в пошуку локацій, які можуть їх зацікавити. Оскільки добре продумана система рекомендацій звільняє користувача від фільтрації великої кількості даних, що в свою чергу дозволяє користувачеві отримувати якісний контент за короткий проміжок часу.

Робота пройшла апробацію під час Всеукраїнської наукової конференції, є самостійною, цілісною та завершеною працею, пояснювальна записка оформлена відповідно до існуючих вимог. Матеріал у магістерській роботі викладено грамотно та структуровано, з використанням достатньої кількості графічних та схематичних матеріалів.

Враховуючі вищенаведене, вважаю за можливе допустити до захисту МКР студента гр. 601 Скакуна Є. І. та присвоїти йому освітню кваліфікацію «Магістр з комп’ютерних наук» в галузі знань 12 «Інформаційні технології» за спеціальністю 122 «Комп’ютерні науки».

Керівник МКР
д-р техн. наук, проф. кафедри ІІС
ЧНУ ім. Петра Могили О. П. Гожий

ЗВІТ
про унікальність пояснювальної записки
магістерської кваліфікаційної роботи на тему:
«Інтелектуальна система для визначення локацій методом машинного навчання»

студента спеціальності 122 «Комп'ютерні науки», 601 групи

_____Скакун Євгеній Ігорович_____
(прізвище, ім'я, по батькові)

Перевірку тексту здійснено сервісом: онлайн-сервіс Unicheck

Результат перевірки тексту магістерської кваліфікаційної роботи: схожість складає 3,65%.

Студент:

_____ Є.І. Скакун
підпис ініціали, прізвище

Керівник:
д-р техн. наук, проф.
_____ О. П. Гожий
підпис ініціали, прізвище

Дата: «___» _____ 2022 р.