Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет комп'ютерних наук
(повна назва)

Кафедра програмної інженерії
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Дослідження методів рекомендації систем
на основі колаборативної фільтрації
(тема)

Виконала:
здобувачка 2 року навчання,
групи ІПЗзм-23-1

Дарія АНДРЮЩЕНКО
(прізвище, ініціали)

Спеціальність 121 – Інженерія програмного забезпечення
(код і повна назва спеціальності)
Тип програми освітньо-наукова

Керівник проф. каф. ПІ, к.т.н., доц. Ігор ШУБІН
(посада, прізвище, ініціали)

Допускається до захисту
Зав. кафедри _____ Кирило СМЕЛЯКОВ
(підпис) (прізвище, ініціали)

2025 р.

Харківський національний університет радіоелектроніки

Факультет комп’ютерних наук
Кафедра програмної інженерії
Рівень вищої освіти другий (магістерський)
Спеціальність 121 – Інженерія програмного забезпечення
Тип програми освітньо-наукова програма
Освітня програма Інженерія програмного забезпечення
(шифр і назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«___» _____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачці Андрющенко Дарії Олегівні
(прізвище, ім’я, по батькові)

1. Тема роботи «Дослідження методів рекомендації систем на основі колаборативної фільтрації»

Затверджена наказом по університету від 10.04. 2025р. № 55 стз

2. Термін подання здобувачем роботи до екзаменаційної комісії 23.06.2025

3. Вихідні дані до роботи опис досліджуваних методів на основі колаборативної фільтрації, критерії та метріки оцінювання, дані для експериментів, мова програмування Java, технологія зборки проекту Maven, бібліотека CF4J, середовище розробки IntelliJ Idea 2023.3.2

4. Перелік питань, що потрібно опрацювати в роботі
аналіз та порівняння існуючих підходів колаборативної фільтрації, вибір конкретних реалізацій алгоритмів, визначення набору метрик, підготовка програмного середовища, проведення експериментів та аналіз отриманих результатів

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз предметної галузі та постановка задачі	23.01 – 14.02.25	*виконано*
2	Аналіз та вибір моделей для дослідження	15.02 – 24.02.25	*виконано*
3	Аналіз та моделювання предметної області	17.02 – 28.02.25	*виконано*
4	Планування експериментів	25.02 – 28.02.25	*виконано*
5	Програмна реалізація кожної з обраних для дослідження моделі	25.02 – 01.04.25	*виконано*
6	Експериментальні дослідження	02.04 – 20.04.25	*виконано*
7	Аналіз результатів експериментальних досліджень та розробка рекомендацій	20.04 – 23.04.25	*виконано*
8	Написання та оформлення статті та тез доповіді	17.04 – 23.04.25	*виконано*
9	Підготовка пояснювальної записки	01.04 – 26.04.25	*виконано*
10	Підготовка презентації та доповіді	26.04 – 2.05.25	*виконано*
11	Нормоконтроль	3.05 – 08.05.25	*виконано*
12	Рецензування	08.05 – 14.05.25	*виконано*
13	Занесення диплома в електронний архів	15.05.2025	*виконано*
14	Попередній захист	15.05.2025	*виконано*
15	Допуск до захисту у зав. кафедри	18.05.2025	*виконано*

Дата видачі завдання 7 квітня 2025р.

Здобувач (ка) _____ Дарія АНДРЮЩЕНКО
(підпис)

Керівник роботи _____ проф. каф. ПІ, к.т.н., доц. Ігор ШУБІН
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ / ABSTRACT

Пояснювальна записка містить: 60 с., 6 рис., 26 джерел.

КОЛАБОРАТИВНА ФІЛЬТРАЦІЯ, МЕТОДИ ПРОГНОЗУВАННЯ, МОДЕЛІ РЕКОМЕНДАЦІЙ, РЕКОМЕНДАЦІЙНІ СИСТЕМИ, CF4J.

Об’єкт дослідження – методи рекомендацій у системах на основі колаборативної фільтрації.

Мета роботи – дослідження принципів роботи рекомендаційних систем, зокрема методів колаборативної фільтрації, аналіз різних моделей прогнозування, підготовка та проведення теоретичного і практичного дослідження з метою визначення найбільш ефективного підходу до реалізації рекомендаційних систем.

Методи розробки та проектування включають аналіз існуючих рекомендаційних систем, порівняння моделей колаборативної фільтрації (user-based, item-based, гібридні), обрання релевантних метрик оцінювання якості прогнозів (RMSE, Precision, Recall), а також використання бібліотеки CF4J на Java для проведення практичного експерименту.

Результат роботи – здійснено огляд основних методів колаборативної фільтрації, реалізовано та протестовано кілька моделей за допомогою CF4J, обрано найбільш ефективну модель прогнозування на основі порівняння результатів за визначеними критеріями. Результати дослідження можуть бути застосовані для покращення персоналізації в системах рекомендацій.

COLLABORATIVE FILTERING, RECOMMENDER SYSTEMS, FORECASTING METHODS, CF4J, RECOMMENDATION MODELS.

Object of research – recommendation methods in systems based on collaborative filtering.

Purpose of the work – to study the principles of recommender systems, specifically collaborative filtering methods, analyze various forecasting models, and

conduct theoretical and practical research to determine the most effective approach for implementing recommender systems.

Development and design methods include the analysis of existing recommender systems, comparison of collaborative filtering models (user-based, item-based, hybrid), selection of relevant evaluation metrics (RMSE, Precision, Recall), and the use of the CF4J library in Java for practical experiments.

Result of the work – an overview of the main collaborative filtering methods was carried out, several models were implemented and tested using CF4J, and the most effective forecasting model was identified based on the evaluation results. The findings can be applied to improve personalization in recommender systems.

Завідувачу кафедри
ПІ_____
(скорочена назва кафедри)
проф. Кирилу СМЕЛЯКОВУ_
(вчене звання, сласне ім'я, прізвище)

ЗАЯВА
щодо самостійності виконання кваліфікаційної роботи та можливості її публікації (та/або публікації анотації кваліфікаційної роботи) в електронному архіві відкритого доступу ElAr KhNURE

Я, Андрющенко Дарія Олегівна, здобувачка гр. ІПЗзм-23-1, здобувач вищої освіти на другому (магістерському) рівні кафедри «Програмна інженерія», заявляю: моя кваліфікаційна робота на тему «Дослідження методів рекомендації систем на основі колаборативної фільтрації», що буде представлена в екзаменаційну комісію для публічного захисту, виконана самостійно, в ній не містяться елементи плагіату і вона може бути опублікована в електронному архіві відкритого доступу ElArKhNURE. Всі запозичення з друкованих та електронних джерел мають відповідні посилання.

Я ознайомлений(на) з діючим положенням «Про протидію академічному плагіату в ХНУРЕ», згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування дисциплінарних заходів.

Дата Підпис

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ

CBR – case-based recommender systems;

CF – колаборативна фільтрація;

PMF – probabilistic matrix factorization;

KNN – метод k-найближчих сусідів;

User-KNN – user-based collaborative filtering model;

Item-KNN – item-based collaborative filtering model;

NCF – neural matrix factorization;

MAE – середня абсолютна помилка;

RMSE – середньоквадратична помилка.

ВСТУП

Коли кількість інформації в інтернеті постійно зростає, стає все важче швидко знаходити потрібний контент. Саме тому рекомендаційні системи відіграють важливу роль – вони допомагають користувачам отримувати персоналізовані поради та пропозиції, ґрунтуючись на їхніх інтересах і поведінці.

Одним із найпоширеніших і ефективних підходів до створення таких систем є колаборативна фільтрація. Вона працює за принципом пошуку схожих користувачів або товарів, щоб запропонувати те, що може зацікавити конкретну людину. Незважаючи на просту ідею, існує багато способів реалізації цього підходу, кожен з яких має свої переваги та недоліки.

Ця тема є актуальною, адже правильний вибір моделі рекомендацій напряму впливає на якість сервісу та задоволеність користувачів. Колаборативна фільтрація вже давно застосовується у таких галузях, як онлайн-магазини, музичні та відео сервіси, соціальні мережі. Проте навіть сьогодні існує потреба в подальшому дослідженні цих методів і виявленні найбільш ефективних з них.

Метою даної роботи є аналіз існуючих методів колаборативної фільтрації, їх порівняння на практиці та визначення, який із них найкраще підходить для побудови сучасних рекомендаційних систем. Результати дослідження можуть бути корисними для розробників програмного забезпечення та всіх, хто працює з персоналізованими сервісами.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Аналіз рекомендаційних систем

Рекомендаційні системи почали активно розвиватись із 90-х років і з того часу стали важливим інструментом в онлайн-сервісах. Є два основні типи рекомендацій: персоналізовані та неперсоналізовані. Неперсоналізовані зазвичай пропонують найпопулярніші товари або ті, що вигідні для бізнесу, тому вони менш цікаві з дослідницької точки зору. У цій роботі ми зосередимось на персоналізованих рекомендаціях, які враховують інтереси та поведінку конкретного користувача. Основна мета таких систем – допомогти користувачу знайти те, що йому дійсно потрібно.

Серед ключових викликів – забезпечення точності, подолання проблеми "холодного старту" (коли система нічого не знає про нового користувача або товар), а також розширення набору рекомендованих товарів [2]. У випадку інтернет-магазинів додатковими завданнями є збільшення продажів, підвищення лояльності клієнтів, ефективна реалізація додаткових та супутніх продажів, а також комунікація з клієнтами після покупки. Для цього важливо правильно оцінювати схожість між користувачами і товарами – як за кількісними, так і за якісними параметрами. У великих онлайн-магазинах ефективною є робота з групами користувачів і товарів, які мають схожі ознаки [3]. Це допомагає робити рекомендації точнішими, особливо коли мова йде про маркетингові кампанії.

Також велике значення мають гібридні підходи, які поєднують різні методи прогнозування, щоб компенсувати слабкі сторони кожного окремого з них. Зокрема, такі методи дозволяють краще працювати з розрідженими даними та забезпечувати більшу різноманітність у пропозиціях. У цьому розділі ми розглянемо загальне уявлення про те, як працюють рекомендаційні системи, а далі перейдемо до аналізу конкретних підходів.

1.1.1 Колаборативні рекомендаційні системи

Суть колаборативної фільтрації полягає в тому, що користувачі з подібними смаками зазвичай цікавляться схожими товарами, а товари, які сподобались одним користувачам, можуть сподобатись і іншим із подібними вподобаннями. Для цього система використовує дані про оцінки, які користувачі вже поставили товарам чи контенту. Ці оцінки зазвичай задаються у вигляді чисел в межах певної шкали, наприклад від 0 до 5 або від 0 до 10. Іноді також використовуються прості значення типу "подобається/не подобається" (1 або 0). Щоб створити рекомендації, система аналізує схожі профілі користувачів або схожі товари й на основі цього формує прогноз. Той, хто звертається до системи, називається активним користувачем, а об’єкт, для якого робиться прогноз – активним товаром.

Колаборативна фільтрація сьогодні є найпоширенішим підходом у рекомендаційних системах. Її основна перевага в тому, що вона не вимагає знань про характеристики товарів – достатньо лише оцінок від користувачів. Проте є й недоліки: дуже велика кількість даних (матриця користувачів і товарів) і складність у формуванні рекомендацій для нових користувачів або нових товарів, для яких ще немає оцінок.

1.1.2 Рекомендаційні системи, які використовують контентну фільтрацію

Системи рекомендацій, що базуються на контентній фільтрації, формують свої прогнози, аналізуючи властивості об’єктів, з якими взаємодіяв користувач. Наприклад, якщо користувач часто ставив високі оцінки фільмам певного жанру, з конкретними акторами чи режисерами, система фіксує ці характеристики у його профілі. Цей профіль інтересів є своєрідним шаблоном, з яким потім порівнюються інші об’єкти, щоб виявити схожі за параметрами. На основі такого порівняння система визначає ступінь відповідності між новим об’єктом і сформованим профілем користувача, що дозволяє передбачити, чи буде цей об’єкт цікавим для нього.

Особливістю контентної фільтрації є те, що вона не використовує дані інших користувачів – рекомендації ґрунтуються лише на індивідуальному досвіді. Завдяки цьому, така система може ефективно працювати навіть при обмеженій кількості користувачів або в ситуації, коли предмет є новим і ще не отримав відгуків. Це вигідно для стартапів, вузькопрофільних платформ або сервісів, де важливу роль відіграє індивідуалізація.

Разом з тим, існують і певні обмеження. Контентна фільтрація повністю залежить від доступності та повноти опису кожного об’єкта. Якщо опис не містить важливої для користувача характеристики або є занадто загальним, система може видати менш релевантні рекомендації. Ще одна слабкість – вона не адаптує або не оновлює інформацію про предмети з часом, тому не здатна вловлювати зміну контексту чи вподобань користувача без явного втручання. У підсумку, хоча контентна фільтрація добре працює для індивідуального аналізу, її ефективність може знижуватись, якщо база даних об’єктів або характеристик є недостатньо гнучкою чи обмеженою.

1.1.3 Демографічні рекомендаційні системи

Системи рекомендацій на основі демографії групують користувачів за такими ознаками, як вік, стать, освіта, професія тощо. Вони будують припущення, що люди з подібними соціальними характеристиками можуть мати схожі вподобання. Наприклад, якщо 25-річному студенту подобаються певні фільми, то іншому користувачу з такою ж віковою категорією та рівнем освіти, ймовірно, також будуть цікаві ті самі фільми. Для цього використовуються профілі, де зберігаються дані про демографію та оцінки, які користувач залишив для різних об'єктів.

Розрахунок рекомендацій здійснюється схоже до методів, що працюють на принципах схожості між товарами чи користувачами, але в цьому випадку основою виступають саме демографічні характеристики. Замість звичайного аналізу оцінок, система обчислює схожість між демографічними профілями, щоб передбачити, які об’єкти можуть зацікавити користувача.

Основна перевага такого підходу полягає в тому, що він не потребує історії взаємодій або оцінок – достатньо лише даних про саму людину. Це дозволяє використовувати систему навіть тоді, коли новий користувач ще нічого не оцінював. Проте є і суттєвий недолік: ці системи не враховують індивідуальні вподобання в межах однієї демографічної групи. Наприклад, двоє людей одного віку й освіти можуть мати зовсім різні інтереси, але система все одно буде пропонувати їм однакові рекомендації. Через це точність прогнозів може бути низькою, якщо не додати інші джерела інформації.

1.1.4 Системи рекомендацій, які використовують знання

Звичайні підходи в рекомендаційних системах, наприклад, контентна фільтрація чи колаборативні методи, добре працюють у тих випадках, коли мова йде про фільми, книги, музику або інші об'єкти, які часто оцінюють багато користувачів. Проте ці методи стають неефективними, коли йдеться про речі, що купуються або використовуються рідко – наприклад, авто, квартири чи складна техніка. У таких випадках немає достатньої кількості оцінок або історії взаємодій, на основі яких можна зробити якісні рекомендації. До того ж, оцінки, зроблені рік чи більше тому, можуть втратити актуальність, що також знижує точність таких систем.

Тут на допомогу приходять системи, які базуються на знаннях. Серед них – рекомендаційні системи на основі прецедентів (CBR) та системи з використанням чітко заданих обмежень і правил (constraint-based). У першому випадку система аналізує попередні, вже вирішені ситуації, і шукає схожі на поточну, щоб запропонувати рішення, яке спрацювало раніше. У другому випадку працює набір фіксованих правил, що вказують, які властивості товарів відповідають конкретним вимогам користувача.

CBR-системи схожі на експертні системи, але замість набору логічних правил, вони використовують базу прикладів з реальними рішеннями [4]. Коли з'являється нова задача, система знаходить найближчі схожі приклади та комбінує їх, щоб запропонувати найкраще рішення. Проте такий підхід має свій мінус:

система не створює узагальнену модель або правила на майбутнє – вона працює лише з уже наявними ситуаціями, що обмежує її здатність до навчання.

1.2 Системи рекомендацій для груп користувачів

Системи рекомендацій для груп користувачів – це окремий клас рекомендаційних систем, які спрямовані не на індивідуальні вподобання, а на врахування потреб групи людей. Такі системи застосовуються у ситуаціях, коли рішення має задовольняти одразу кількох користувачів, наприклад, у виборі фільму для спільного перегляду, плануванні подорожі чи підборі товарів для родини або команди. На відміну від персоналізованих підходів, де система адаптується до інтересів окремого користувача, тут необхідно поєднати вподобання декількох людей і знайти компромісне рішення.

Основні завдання таких систем включають ідентифікацію груп зі схожими смаками, побудову агрегованого профілю групи та формування рекомендацій, які будуть прийнятними для більшості членів. Існують різні стратегії об'єднання інтересів – наприклад, обрахунок середнього рейтингу, знаходження найменш конфліктного варіанту або ж врахування "лідерів думок" у групі. Одним із викликів у цьому підході є уникнення ситуацій, коли інтереси окремих учасників повністю ігноруються. З розвитком технологій і зростанням соціальної активності в Інтернеті, групові рекомендаційні системи стають дедалі більш актуальними в багатьох прикладних сферах.

2 ПРОБЛЕМИ ТА ВИКЛИКИ КОЛАБОРАТИВНОЇ ФІЛЬТРАЦІЇ

Хоча колаборативна фільтрація є одним із найефективніших і найпопулярніших методів рекомендацій, вона має низку суттєвих проблем, що обмежують її ефективність у реальних умовах. У цьому розділі розглянуто основні виклики, з якими стикаються системи, побудовані на цьому підході.

2.1 Проблема розрідженості даних

Однією з головних проблем колаборативної фільтрації є розрідженість матриці взаємодій між користувачами та об’єктами. У великих системах користувачі зазвичай взаємодіють лише з незначною частиною об’єктів, що призводить до великої кількості порожніх (невідомих) значень у матриці. Це ускладнює обчислення схожості та знижує точність прогнозів, особливо на початкових етапах функціонування системи або при роботі з новими користувачами [5].

В умовах розрідженості важко знайти користувачів або об’єкти з достатньою кількістю спільних оцінок для якісного обчислення подібності. Це також впливає на стійкість системи до шуму та аномалій у даних.

2.2 Проблема «холодного старту»

Проблема «холодного старту» виникає при появі нових користувачів або об’єктів, для яких відсутня достатня історія взаємодії. У таких випадках система не має змоги здійснити прогнозування на основі схожості, оскільки не існує з чим порівнювати.

Подолати цю проблему дозволяє інтеграція колаборативного підходу з контентним. Наприклад, при додаванні нового об’єкта система може використати його атрибути (жанр, опис, характеристики) для початкових рекомендацій. Для нових користувачів застосовують короткі опитування або реєстраційні анкети для збору первинної інформації, а також можуть бути використані демографічні ознаки або початкові рейтинги популярних об’єктів.

2.3 Проблема розрідженості

Однією з найпоширеніших проблем є розрідженість матриці взаємодій між користувачами та об’єктами. У більшості випадків користувачі оцінюють або взаємодіють лише з невеликою частиною об’єктів, унаслідок чого велика частина елементів матриці залишається порожньою. Це ускладнює визначення схожості та знижує ефективність прогнозування.

Для вирішення цієї проблеми використовуються методи матричного розкладання, які дозволяють відновити відсутні значення на основі прихованих (латентних) факторів. Також допомагає застосування імпліцитних ознак, таких як перегляди чи кліки, які частіше зустрічаються, ніж явні оцінки. Додатково, у гібридних системах використовують контентну інформацію або метадані, щоб компенсувати брак взаємодій.

2.4 Проблема масштабованості

Із зростанням кількості користувачів та об’єктів у системі збільшується розмір матриці, а разом із цим – складність обчислення прогнозів. Особливо гостро ця проблема постає у memory-based методах, які вимагають безпосереднього порівняння великої кількості записів у процесі кожного запиту.

Для підвищення масштабованості використовуються model-based підходи, які дозволяють спершу побудувати модель на основі навчальних даних, а потім застосовувати її для прогнозів у реальному часі. Крім того, поширеним рішенням є використання розподілених обчислень, таких як Apache Spark або Hadoop, а також застосування кластеризації, індексації та інших способів зменшення обсягу даних, що обробляються одночасно.

2.5 Брак новизни у рекомендаціях

Колаборативні системи мають тенденцію рекомендувати популярні об’єкти, що вже неодноразово оцінювались великою кількістю користувачів. У результаті рекомендації втрачають різноманітність і не охоплюють нові або нішеві об’єкти, які могли б бути цікавими окремим користувачам.

Рішенням цієї проблеми є впровадження метрик новизни та різноманітності у функцію оцінки результатів. Це дозволяє моделі надавати перевагу менш популярним, але потенційно релевантним об’єктам. Також у гібридних підходах можна включати контентні характеристики, що дозволяє виділяти об’єкти за тематичною близькістю до вже вподобаних.

2.6 Уразливість до атак і маніпуляцій

Колаборативні моделі можуть бути ціллю для зловмисників, які маніпулюють оцінками через створення фальшивих профілів або масові «накрутки» рейтингів. Це призводить до зміщення рекомендацій і порушення довіри користувачів.

Запобігти подібним атакам допомагають методи виявлення аномальної активності, які аналізують нетипову поведінку або статистичні відхилення. Також застосовуються обмеження ваги нових користувачів, перевірка достовірності даних і фільтрація підозрілих оцінок. У складніших системах впроваджуються моделі з довірчими коефіцієнтами або історичною репутацією.

3 ОГЛЯД МОДЕЛЕЙ ТА АЛГОРИТМІВ КОЛАБОРАТИВНОЇ ФІЛЬТРАЦІЇ

3.1 Алгоритми колаборативної фільтрації

Серед усіх типів рекомендаційних систем найбільшу популярність нині здобули ті, що працюють на принципах колаборативної фільтрації. Суть цього підходу – у тому, щоб підказувати користувачу нові варіанти на основі того, що подобалося йому раніше, або з огляду на смаки інших людей, які мають схожі вподобання. З часом дослідники створили цілу низку методів реалізації такого підходу, які умовно можна згрупувати в три основні напрямки:

- model-based (побудовані на моделі даних);
- memory-based (побудовані на пам’яті);
- gybrid-based (поєднання model-based та memory-based).

3.2 Методи, засновані на побудові моделі даних (Model-based)

Модельно-орієнтовані рекомендаційні системи відрізняються тим, що не працюють безпосередньо з усією історією оцінок під час кожного запиту. Натомість вони навчаються на наявних даних і створюють узагальнену модель, яка містить найважливішу інформацію про вподобання користувачів та властивості об’єктів. Ця модель дозволяє швидко робити прогнози щодо того, що може сподобатися конкретному користувачу, навіть якщо він раніше не бачив певний об’єкт.

Навчання таких систем зазвичай відбувається у два етапи. Спершу виконується обчислювально складне формування моделі на основі історичних даних. Після цього сформовану модель можна багаторазово використовувати для швидкого формування рекомендацій у реальному часі. Завдяки такому підходу модельно-орієнтовані системи чудово масштабуються і підходять для роботи з великими обсягами даних.

У якості моделей найчастіше використовуються методи машинного навчання, зокрема ті, що здатні виявляти приховані закономірності у великих наборах даних [7]. Одним із найпоширеніших підходів є матрична факторизація. Її суть полягає у представленні великої розрідженої матриці оцінок у вигляді

добутку двох менших матриць – одна з яких описує користувачів, а інша – об’єкти. Це дозволяє ефективно зберігати інформацію про зв’язки між ними й робити точні прогнози.

Формально цей підхід можна записати у вигляді рівняння (див. формулу 3.1):

(3.1)

де U – матриця, що представляє латентні фактори користувачів,

V – матриця, що представляє латентні фактори елементів.

Коли ми перемножуємо матриці, створені під час матричної факторизації, ми отримуємо приблизне відтворення початкової таблиці оцінок. Це дозволяє «заповнити прогалини» – тобто передбачити оцінки в тих місцях, де користувач ще нічого не оцінював. Такий підхід особливо корисний у випадках, коли інформації про користувача або об'єкт дуже мало, але все одно потрібно зробити точну рекомендацію [9].

Матрична факторизація добре працює в умовах, коли даних багато, але вони неповні, що типово для великих систем. Вона дозволяє досягати високої точності, не вимагаючи повного заповнення всієї таблиці вподобань.

У більш сучасних підходах замість класичної факторизації часто використовуються нейронні мережі. Вони навчаються на великих наборах даних і здатні помічати складні зв’язки між користувачами та об’єктами, які не видно простими методами. Мережа «вчиться» на взаємодіях, а потім може передбачати, що саме сподобається користувачу, навіть якщо він ще не стикався з певним товаром чи фільмом.

Інформація про те, як користувачі оцінювали чи взаємодіяли з елементами, подається на вхід моделі. Після навчання вона починає передбачати можливі оцінки для нових комбінацій. У процесі навчання мережа коригує свої внутрішні параметри, підлаштовуючись під специфіку кожного користувача й набору даних, що робить рекомендації більш персоналізованими та точними.

Завдяки здатності працювати з розрідженими даними, масштабуватися та будувати точні прогнози, методи матричної факторизації й нейронні моделі сьогодні вважаються ключовими інструментами в сучасних рекомендаційних системах, побудованих на колаборативній фільтрації.

3.3 Методи, засновані на аналізі наявних оцінок (Memory-based)

Цей підхід часто називають методом найближчих сусідів, або просто KNN. Його суть полягає в тому, щоб аналізувати попередні оцінки, які користувач залишав раніше, і порівнювати їх з оцінками інших людей, чиї вподобання схожі. Якщо знайти тих, хто має подібний смак, можна припустити, що їм сподобаються схожі речі. На основі цього й формуються рекомендації.

Такі системи працюють безпосередньо з таблицею, де зібрано всі оцінки – її ще називають матрицею корисності. На першому етапі створюється модель, яка враховує саме цю таблицю. Можна умовно записати це як (див. формулу 3.2):

(3.2)

Після цього система бере дані з профілю користувача і через модель обчислює, що йому варто порекомендувати. Проте, якщо користувач новий і ще не залишав оцінок, його спершу потрібно додати до матриці, а також перерахувати всі схожості з іншими – цей процес досить затратний за часом і ресурсами.

Цей метод колаборативної фільтрації буває двох типів. Перший – коли рекомендації формуються на основі схожості між користувачами [11]. У цьому випадку система шукає тих, хто має подібні вподобання, і дивиться, що саме вони вже оцінили. Якщо декілька «схожих» користувачів позитивно відгукувались про певний об’єкт, то його й пропонують нашому користувачу.

Другий тип – це фільтрація на основі схожості між об’єктами. Тут система не шукає схожих людей, а натомість дивиться, які об’єкти подібні до тих, що вже сподобалися користувачу [10]. Наприклад, якщо він поставив високу оцінку

певному фільму, система знайде фільми з подібними оцінками та темами – і запропонує їх. Остаточна оцінка нових об’єктів визначається як середньозважене всіх рейтингів подібних до нього об’єктів.

У всіх таких методах пошук «схожості» виконується за допомогою спеціальних математичних метрик, які визначають, наскільки близькими є користувачі або об’єкти за своїми оцінками.

3.4 Гібридні методи

Гібридні алгоритми колаборативної фільтрації поєднують у собі найкраще з двох світів – ті підходи, які будуються на математичних моделях, і ті, що працюють із реальними оцінками користувачів. Іншими словами, такі системи не тільки використовують готові формули для прогнозування вподобань, а й зберігають велику базу даних про минулі дії користувачів, щоб знаходити між ними схожість [8].

Завдяки цьому поєднанню система може робити точніші й більш гнучкі рекомендації, бо вона не покладається лише на одне джерело даних. Вона вміє адаптуватися до змін – наприклад, якщо користувачі починають поводитись інакше або з’являються нові об’єкти. Такі гібридні моделі особливо корисні в ситуаціях, коли один підхід не справляється – наприклад, при розрідженості даних, нових користувачах чи дуже великому обсязі інформації [12]. У результаті виходить система, яка здатна ефективно реагувати на різні умови й давати точні персоналізовані рекомендації.

4 ПОСТАНОВКА ЗАДАЧІ ТА ВИБІР МОДЕЛЕЙ ДЛЯ ДОСЛІДЖЕННЯ

4.1 Постановка технічного завдання

Метою технічної частини цієї магістерської роботи є практичне дослідження та порівняння ефективності різних підходів до реалізації колаборативної фільтрації в рекомендаційних системах. Для досягнення цієї мети передбачено розробку та експериментальне тестування чотирьох алгоритмів, що представляють ключові напрямки в рамках колаборативної фільтрації:

- user-based collaborative filtering – підхід, що формує рекомендації на основі подібності між користувачами;
- item-based collaborative filtering – метод, що орієнтується на схожість між об'єктами;
- matrix factorization – алгоритм, що виявляє приховані зв’язки між користувачами та об’єктами шляхом зменшення розмірності матриці вподобань;
- neural collaborative filtering (ncf) – модель, яка використовує нейронні мережі для прогнозування оцінок і побудови рекомендацій.

Усі обрані алгоритми буде реалізовано в єдиному програмному середовищі з використанням одного і того ж набору даних, що дозволить забезпечити справедливе порівняння за однакових умов. Як набір даних планується використати відкриту колекцію, яка містить інформацію про вподобання користувачів, їхні оцінки об’єктів та інші релевантні дані для побудови рекомендацій.

Для кожного з алгоритмів буде здійснено навчання моделі на тренувальній частині датасету та проведено тестування на валідаційній або тестовій вибірці. Якість рекомендацій буде оцінюватися за допомогою стандартних метрик, таких як MAE (Mean Absolute Error)**,** RMSE (Root Mean Squared Error).

Результати експериментів дадуть змогу проаналізувати переваги та недоліки кожного методу, зокрема з точки зору точності прогнозування, обчислювальної складності та придатності до масштабування. Підсумком цього дослідження стане обґрунтований вибір оптимального алгоритму для подальшого

застосування або розвитку рекомендаційних систем на основі колаборативної фільтрації.

4.2 Вибір моделей прогнозування рекомендаційних систем

У цій роботі розглядаються чотири різні методи колаборативної фільтрації: user-based, item-based, матрична факторизація та нейронна колаборативна фільтрація (NCF). Саме ці методи було обрано, щоб охопити як базові, так і більш сучасні підходи до побудови рекомендацій.

User-based і item-based методи – це класичні алгоритми, які працюють досить просто: вони шукають схожих користувачів або схожі об’єкти, і на основі цього формують рекомендації. Такі методи легко реалізувати, і вони добре підходять для розуміння принципів колаборативної фільтрації. Їх часто використовують як відправну точку у створенні рекомендаційних систем.

Матрична факторизація – це вже складніший підхід, який дозволяє виявити приховані зв’язки між користувачами та об’єктами. Він працює навіть тоді, коли в матриці багато пропущених значень, тобто не всі користувачі залишили оцінки всім об’єктам. Цей метод є дуже популярним у багатьох сучасних рекомендаційних системах, бо дозволяє досягати хороших результатів у великих обсягах даних.

Нейронна колаборативна фільтрація – найсучасніший із обраних методів. Вона базується на використанні нейронних мереж і дозволяє знаходити складні закономірності в поведінці користувачів. Цей підхід здатен підлаштовуватись під специфіку даних і дає можливість отримувати точніші персоналізовані рекомендації.

Ми зосереджуємось саме на цих чотирьох методах, бо вони добре показують різні етапи розвитку колаборативної фільтрації: від простих моделей до більш потужних та адаптивних. Крім того, їх зручно порівнювати в однакових умовах, щоб зрозуміти, який з них показує кращі результати на практиці.

4.2.1 User-KNN

Алгоритм User-KNN працює так: він знаходить користувачів, у яких схожі оцінки або вподобання, і використовує цю інформацію, щоб зробити рекомендації. Щоб визначити, хто на кого схожий, система порівнює, як різні користувачі оцінювали ті самі об'єкти. Для цього використовуються спеціальні способи вимірювання схожості, наприклад, косинусна відстань або коефіцієнт Пірсона.

Цей метод належить до групи алгоритмів, які працюють з уже наявними оцінками – тобто він не створює складну модель, а орієнтується на дані, які є в базі. Суть у тому, що якщо кілька людей поділяли однакову думку в минулому, то є велика ймовірність, що вони так само оцінять інші об'єкти в майбутньому. Тому система просто бере оцінки схожих користувачів і на їх основі прогнозує, що може сподобатися поточному користувачу.

User-KNN дозволяє показати, як працює один із найстаріших і найзрозуміліших підходів у колаборативній фільтрації. Він добре ілюструє, як можна використовувати зв’язки між людьми у даних, щоб робити персоналізовані поради.

4.2.2 Item-KNN

Item-KNN – це метод, який визначає, наскільки об’єкти (наприклад, фільми, книги чи товари) схожі між собою, ґрунтуючись на тому, як їх оцінювали користувачі. Інакше кажучи, якщо два товари мають схожі оцінки від багатьох людей, система вважає їх подібними. Потім на основі цих зв’язків і формуються рекомендації.

На відміну від User-KNN, де аналізується схожість між самими користувачами, тут порівнюються саме об’єкти. Наприклад, якщо користувач поставив високу оцінку певному фільму, система може порекомендувати інший фільм, який часто отримував схожі оцінки від інших людей [13].

Щоб визначити, наскільки об’єкти схожі, система використовує спеціальні розрахунки, наприклад, косинусну подібність. Іноді застосовують модифіковані

формули, які краще враховують індивідуальні особливості оцінювання. У підсумку, рекомендації формуються, виходячи з того, наскільки подібні товари вже оцінені цим користувачем.

Цей підхід добре працює тоді, коли про самих користувачів мало інформації, але накопичено достатньо оцінок для об’єктів [14]. Він показує, як можна робити рекомендації, аналізуючи не стільки людей, скільки зв’язки між товарами, які їм подобаються.

4.2.3 Імовірнісне матричне розкладання

PMF – це метод, який використовує приховані фактори для опису взаємодії між користувачами та об'єктами, але на відміну від звичайної матричної факторизації, він працює з ймовірностями. Інакше кажучи, замість одного точного значення ця модель намагається передбачити можливий діапазон оцінок. Вона розглядає всю задачу як випадковий процес, а також використовує байєсівський підхід для кращої інтерпретації результатів.

Такий метод особливо добре підходить для ситуацій, коли даних дуже мало, тобто коли матриця оцінок сильно розріджена – що, власне, часто буває в реальних рекомендаційних системах.

У PMF застосовуються нормальні (гауссівські) розподіли, щоб врахувати похибки в оцінках, і використовуються ймовірнісні методи для розрахунку параметрів. Це дозволяє моделі "зізнаватись", коли вона не впевнена у своєму прогнозі, що є великою перевагою, якщо важлива надійність результатів.

Такий підхід демонструє, як можна застосовувати статистику у побудові рекомендацій. Він дає змогу досягати більшої точності, краще працювати з великими даними та справлятися з невизначеністю в умовах, коли інформації не вистачає.

4.2.4 Нейронна колаборативна фільтрація

NCF – це підхід, який об’єднує класичні методи матричної факторизації з можливостями нейронних мереж. У ньому поєднуються два окремих компоненти:

один відповідає за виявлення базових зв’язків між користувачами й об'єктами, а інший, побудований на багатошаровій нейромережі (MLP), вміє знаходити більш складні, нестандартні залежності. Результати обох частин поєднуються, щоб спрогнозувати оцінку користувача.

Сильна сторона NCF полягає в тому, що модель здатна навчатися навіть на неявних, складних шаблонах у даних, які класичні методи часто не помічають. Це дає змогу формувати точніші рекомендації, особливо в тих випадках, коли вподобання користувача не можна пояснити простою логікою.

Для побудови прогнозів NCF використовує вектори прихованих характеристик (факторів), які описують користувачів і об’єкти. Ці вектори подаються на вхід нейронної мережі, яка складається з кількох шарів і застосовує спеціальні функції активації, наприклад ReLU, щоб навчитися розпізнавати складні схеми взаємодії між користувачами та контентом [16].

Включення NCF у це дослідження дає можливість перевірити, наскільки ефективним може бути глибоке навчання у сфері персоналізованих рекомендацій. Це дозволяє порівняти класичні підходи з новими й зрозуміти, чи справді сучасні нейромережеві моделі краще справляються з пошуком прихованих зв’язків у даних.

5 ОПИС ВИКОРИСТАНИХ МЕТРИК ТА ТЕХНОЛОГІЙ ДЛЯ ПРОВЕДЕННЯ ДОСЛІДЖЕННЯ

5.1 Засоби проведення дослідження

У цьому дослідженні практична частина буде реалізована за допомогою мови програмування Java у середовищі Intellij Idea. Java обрана не випадково – це одна з найпоширеніших мов у світі, яка відома своєю надійністю, гарною швидкістю виконання програм і великою кількістю готових бібліотек. До того ж вона має активну спільноту, де легко знайти приклади, поради й рішення типових проблем. Ще однією перевагою Java є автоматичне керування пам’яттю, що особливо важливо для експериментів, які можуть тривати довго.

Для реалізації й тестування алгоритмів колаборативної фільтрації в рамках дослідження обрана бібліотека CF4J, яка також працює на Java. Вона надає готові засоби для створення, запуску та аналізу рекомендаційних моделей. Це дозволяє швидко оцінювати якість різних алгоритмів і зручно їх порівнювати між собою в єдиному середовищі.

Ще однією сильною стороною бібліотеки CF4J є те, що вона легко розширюється. За потреби можна додати власні алгоритми або змінити вже наявні, не витрачаючи на це багато часу. Це дуже зручно, якщо потрібно адаптувати систему під конкретні завдання чи протестувати нетипові методи. Саме така гнучкість робить CF4J оптимальним вибором для проведення прикладного дослідження в цій роботі.

5.2 Методи подібності

Методи обчислення подібності – це інструменти, які допомагають визначити, наскільки два об’єкти схожі між собою. Їх часто використовують, коли потрібно порівняти великі обсяги даних, наприклад, у задачах машинного навчання або створення рекомендацій. У таких випадках об'єкти подають у вигляді чисел, і система визначає "відстань" або "схожість" між ними за певними формулами [18].

Ці методи є важливою частиною багатьох алгоритмів, які передбачають уподобання користувачів. У нашому дослідженні вони використовуються в таких моделях, як user-KNN та item-KNN, де від подібності залежить, які рекомендації отримає користувач.

Найпопулярніші способи оцінити схожість – це косинусна подібність і кореляція Пірсона. Вони дають хороші результати і тому широко застосовуються в рекомендаційних системах. Проте існують і інші метрики, як-от міра Жаккара або кореляція Спірмена. Усі вони мають свої особливості та сильні сторони, тому вибір конкретної метрики залежить від того, з якими даними працює система і яких результатів потрібно досягти.

5.2.1 Косинусна подібність

Косинусна подібність – це спосіб зрозуміти, наскільки два об’єкти схожі між собою, якщо представити їх у вигляді числових векторів. Уявімо, що кожен об’єкт має набір характеристик, і ці характеристики формують вектор у багатовимірному просторі. Щоб дізнатися, наскільки два об’єкти подібні, система дивиться на кут між цими векторами [20]. Якщо вектори майже дивляться в одному напрямку, тобто кут між ними маленький, – об’єкти дуже схожі. Якщо кут більший – схожість менша.

Косинусна подібність вимірюється за допомогою формули, яка порівнює напрямки двох векторів. Результат такого розрахунку – число від 0 до 1. Чим ближче це число до 1, тим більш схожими вважаються об’єкти. Формула враховує не розміри самих векторів, а саме напрямок, тому вона добре працює навіть тоді, коли дані дуже різноманітні за масштабом (див. формулу 5.1):

(5.1)

де $a \cdot b$ позначає скалярний добуток векторів,

$\|a\|$ і $\|b\|$ є нормами (довжинами) цих векторів.

Особливо корисною ця метрика є у випадках, коли матриця даних містить багато нулів – тобто коли користувачі оцінили лише невелику частину всіх можливих об’єктів. У таких умовах, наприклад, у великих рекомендаційних системах, косинусна подібність дозволяє робити точні порівняння, навіть коли інформації небагато.

5.2.2 Кореляція Пірсона

Кореляція Пірсона – це спосіб дізнатись, чи існує зв’язок між двома наборами чисел і наскільки він сильний. Вона показує, як зміни в одному наборі значень пов’язані зі змінами в іншому. Якщо обидва зростають або зменшуються одночасно – зв’язок позитивний, якщо один зростає, а інший падає – зв’язок негативний.

Коефіцієнт кореляції Пірсона r розраховується за формулою 5.2:

(5.2)

де та є значеннями двох змінних,

та є середніми значеннями змінних X та Y відповідно.

Щоб порахувати цей коефіцієнт, спочатку дивляться, наскільки кожне значення відхиляється від середнього. Потім множать ці відхилення для кожної пари значень. Ці множення вказують на напрямок зв’язку: додатні значення – це прямий зв’язок, від’ємні – зворотний. Сума цих множень показує загальний характер зв’язку. Щоб результат був зручним для порівняння, його нормалізують: ділять на спеціальний коефіцієнт, що враховує масштаби обох наборів даних. У підсумку значення кореляції Пірсона завжди лежить у межах від -1 до +1.

Якщо результат близький до +1 – зв’язок сильний і прямий. Якщо близький до -1 – зв’язок сильний, але зворотний. Якщо значення близьке до 0 – ніякого чіткого зв’язку немає.

Цей показник дуже корисний, коли потрібно з’ясувати, як пов’язані між собою різні характеристики, наприклад, у системах рекомендацій. Він особливо

добре працює, коли важливо враховувати не абсолютні значення, а саме відхилення від середнього рівня – наприклад, у випадку, коли один користувач ставить високі оцінки всьому, а інший навпаки – низькі. Кореляція Пірсона дозволяє вирівнювати такі відмінності та краще оцінювати подібність у вподобаннях.

5.2.3 Міра Жаккара

Міра Жаккара, або як її ще називають – індекс Жаккара, використовується для того, щоб оцінити, наскільки дві множини (тобто набори даних) схожі між собою. Вона показує, яка частина елементів є спільною для обох наборів, порівняно з усіма елементами, що зустрічаються хоча б в одному з них.

Щоб це визначити, спочатку рахується, скільки елементів є одночасно в обох множинах – це так званий перетин. Потім обчислюється об’єднання – тобто всі унікальні елементи, що є хоча б в одній множині. Після цього перетин ділять на об’єднання, і виходить значення від 0 до 1. Якщо результат близький до 1 – множини майже однакові. Якщо ж значення ближче до 0 – вони майже не мають спільного [19].

Індекс Жаккара J між двома множинами A та B визначається як (формула 5.3):

(5.3)

де $|A \cap B|$ – це кількість елементів у перетині множин A та B, тобто кількість елементів, які присутні в обох множинах,

$|A \cup B|$ – це кількість елементів в об'єднанні множин A та B, тобто всі унікальні елементи, які присутні в одній або обох множинах.

Наприклад, якщо у двох користувачів є схожі оцінки або вподобання – міра Жаккара це покаже. Її зручно застосовувати там, де важливо враховувати не просто значення оцінок, а сам факт наявності чи відсутності інтересу до певного

об’єкта – наприклад, у випадках, коли дані представлені як бінарні: "переглянуто / не переглянуто", "куплено / не куплено", "оцінено / не оцінено".

Індекс Жаккара корисний для завдань, де потрібно зрозуміти ступінь схожості в умовах обмеженої або розрідженої інформації, зокрема в системах рекомендацій чи класифікації.

5.2.4 Кореляція Спірмана

Кореляція Спірмена – це спосіб перевірити, наскільки два набори даних змінюються разом, але не через самі значення, а через їх порядок (тобто ранги). Іншими словами, вона показує, чи є зв’язок між тим, як змінюються місця елементів у двох списках. Цей метод добре працює з даними, які не обов’язково мають нормальний розподіл або містять викиди.

Щоб обчислити кореляцію Спірмена, кожному значенню в обох наборах спочатку призначають ранг – наприклад, перше місце, друге, третє і так далі. Якщо два значення однакові, їм присвоюється середнє місце. Потім для кожної пари елементів рахується різниця між їхніми рангами. Усі ці відхилення зводяться до квадратів, підсумовуються, і з цього розраховується остаточне значення за спеціальною формулою. Результат буде від -1 до +1: +1 означає повний позитивний зв’язок, -1 – повний негативний, а 0 – відсутність зв’язку.

Коефіцієнт рангової кореляції Спірмена ρ між двома змінними X і Y можна визначити за формулою 5.4:

(5.4)

де – різниця рангів кожного спостереження в двох наборах даних,

n – кількість спостережень.

У рекомендаційних системах цей метод корисний, коли потрібно порівнювати не самі оцінки, а порядок, у якому користувачі розставили вподобані об’єкти. Наприклад, навіть якщо двоє людей ставлять різні числа, але фільми їм подобаються в однаковій послідовності – кореляція Спірмена покаже, що вони

схожі. Вона також зручна, коли дані нестандартні, з пропусками або мають викиди, бо менше чутлива до таких особливостей.

5.3 Обрання показників оцінки моделей прогнозування

Щоб реально оцінити, наскільки добре працює той чи інший метод прогнозування, потрібно проаналізувати, наскільки точними є його результати. Для цього використовують спеціальні показники, які вимірюють помилки моделі – тобто наскільки передбачене значення відрізняється від того, що є насправді. Такі показники допомагають кількісно порівняти різні моделі між собою й об’єктивно оцінити, яка з них справляється з завданням краще.

У рекомендаційних системах найчастіше застосовують дві метрики – MAE (середню абсолютну помилку) та RMSE (середньоквадратичну помилку). Обидві вони дають змогу оцінити, наскільки великі в середньому відхилення між прогнозами й реальними значеннями. Ці метрики добре відомі й активно використовуються як у практиці, так і в наукових дослідженнях, бо дають зрозумілу та точну оцінку якості роботи моделей.

У цьому дослідженні ми будемо використовувати саме MAE та RMSE для порівняння результатів усіх протестованих методів. Це дозволить нам глибше зрозуміти, як моделі поводяться як в середньому, так і в умовах, коли прогноз сильно відрізняється від реального значення.

Вибір цих метрик також важливий з практичної точки зору – адже в реальних системах, особливо в комерційних, потрібно, щоб модель була не лише точною в загальному, а й здатною правильно працювати навіть у складних або нестандартних ситуаціях.

5.3.1 Середня абсолютна помилка

MAE – це показник, який дозволяє зрозуміти, наскільки в середньому прогноз відрізняється від реального значення. Іншими словами, ми просто беремо

різницю між тим, що передбачила модель, і тим, що насправді сталося, переводимо цю різницю в абсолютне значення (тобто без «-» або «+») і рахуємо середнє для всіх випадків у наборі даних.

Формула виглядає так (формула 5.5):

(5.5)

де n – кількість оцінюваних елементів у датасеті,

– фактичне значення і-го елемента,

у – прогнозоване значення для і-го елемента,

- абсолютна помилка прогнозу для і-го елемента.

Чим менше значення MAE, тим точніше працює модель. Якщо значення дорівнює нулю – це означає, що помилок взагалі не було, тобто всі прогнози були ідеальними. Тому, коли ми порівнюємо декілька моделей, найкращою вважається та, у якої MAE найнижче.

Ця метрика дуже зручна, бо її легко зрозуміти: вона прямо показує, наскільки сильно в середньому «промахується» модель у тих самих одиницях, у яких вимірюється результат (наприклад, у балах або кількості). Вона особливо добре підходить для задач, де важливо знати точний розмір помилки.

MAE також менш чутлива до одиничних дуже великих помилок (так званих «викидів»), ніж інші метрики, наприклад RMSE. Але водночас вона не розрізняє, наскільки сильно помилки відрізняються одна від одної. Тому її часто використовують разом з іншими показниками, щоб отримати повнішу картину.

5.3.2 Середньоквадратична помилка

Середньоквадратична помилка (MSE) – це спосіб оцінити, наскільки далеко в середньому прогнози моделі відрізняються від справжніх значень. Для цього всі помилки підносять до квадрата (щоб не було від’ємних значень), потім

обчислюють середнє – і це і є MSE. Така оцінка враховує як те, наскільки прогнозовані значення розкидані між собою, так і наскільки вони відхиляються від реальних результатів.

RMSE – це вдосконалена версія MSE. Щоб отримати RMSE, потрібно просто взяти квадратний корінь із середньоквадратичної помилки. Формула виглядає так (формула 5.6):

(5.6)

де n – кількість оцінюваних елементів у датасеті,

– фактичне значення і-го елемента,

у – прогнозоване значення для і-го елемента,

- квадрат відстані між фактичним та прогнозованим значеннями для і-го елемента.

Ця метрика особливо чутлива до великих помилок. Якщо десь модель сильно «промахнулась» – RMSE це підкреслить сильніше, ніж MAE. Саме тому її часто використовують у тих випадках, коли важливо уникати великих відхилень – наприклад, у фінансових чи критичних бізнес-задачах.

RMSE зручно використовувати разом з MAE. Якщо значення RMSE значно вище за MAE – це може означати, що в моделі є кілька великих помилок. Якщо ж обидва показники близькі – помилки моделі, швидше за все, невеликі й рівномірно розподілені. Це дозволяє краще розуміти якість прогнозування і контролювати точність моделі.

6 ПРАКТИЧНЕ ДОСЛІДЖЕННЯ МЕТОДІВ ПРОГНОЗУВАННЯ РЕКОМЕНДАЦІЙНИХ СИСТЕМ

6.1 Підготовка до проведення експерименту

Для проведення експерименту було обрано датасет MovieLens100k – це відомий набір даних, який часто використовують у дослідженнях, пов’язаних із рекомендаційними системами. Ми будемо оцінювати якість моделей за допомогою метрик похибки, але також важливо врахувати метрики подібності, які застосовуються в двох з чотирьох обраних моделей – user-KNN та item-KNN.

Ці метрики подібності напряму впливають на те, які рекомендації формує модель, тому результати можуть змінюватися в залежності від того, яку саме метрику ми використаємо. У моделях PMF і NCF такі метрики не застосовуються, тому для зручності дослідження було поділено на два етапи.

На першому етапі ми протестуємо різні метрики подібності для моделей user-KNN і item-KNN та визначимо, які з них дають найменші похибки. Оберемо по одній найкращій реалізації для кожної з цих моделей. Після цього, на другому етапі, вже будемо порівнювати обрані реалізації user-KNN та item-KNN з результатами моделей PMF і NCF, щоб оцінити, який метод працює найточніше.

6.2 Дослідження метрик подібності

У цьому дослідженні ми будемо тестувати кілька метрик подібності, які обрали для порівняння: косинусну схожість, індекс Жаккара, кореляцію Пірсона та Спірмена. Ці метрики застосовуються в трьох із чотирьох моделей, які ми аналізуємо, тож для кожної з них ми будемо перевіряти, як змінюється результат залежно від обраної метрики.

Окрім цього, ще один важливий параметр, який впливає на якість прогнозування в усіх моделях – це кількість «сусідів», тобто схожих користувачів або об’єктів, яких враховує модель. Для дослідження ми обрали перевірку з різною кількістю сусідів: від 10 до 50, з кроком 10.

6.2.1 Дослідження метрики подібності для user-KNN моделі

Результат дослідження метрик подібності для моделі user-KNN (див. рис. 6.1):

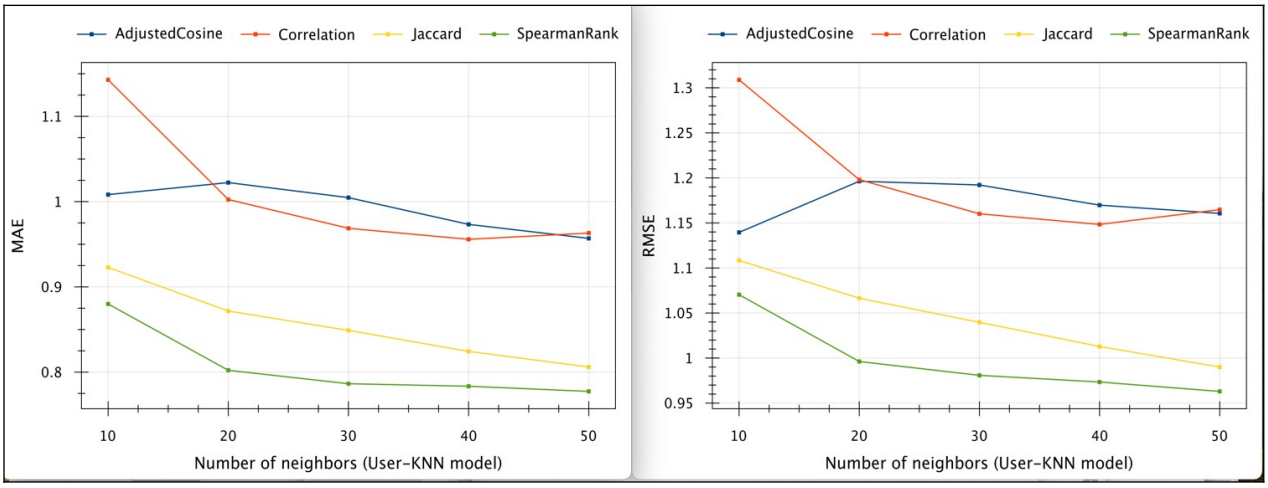

Рисунок 6.1 – Значення помилок MAE та RMSE для User-KNN моделі в залежності від обраної метрики подібності (рисунок виконано самостійно)

Згідно з результатами на графіках, найкращі результати показала кореляція Спірмена. Вона стабільно має найменші значення обох помилок у всьому діапазоні – від 10 до 50 сусідів. Це означає, що саме ця метрика дає найточніші прогнози. Метрика Жаккара теж показала непогані результати, але трохи гірші, ніж у кореляції Спірмена. Натомість косинусна подібність і кореляція Пірсона мають вищі значення MAE і RMSE, особливо коли кількість сусідів невелика, тому вони менш ефективні. Отже, кореляцію Спірмена можна вважати найкращим варіантом для другого етапу дослідження.

6.2.2 Дослідження метрики подібності для item-KNN моделі

Дослідження метрик подібності для item-KNN моделі представлено на рисунку 6.2.

Аналіз результатів демонструє, що найкращі показники точності має індекс Жаккара. Він стабільно дає найнижчі значення помилок як за MAE, так і за RMSE на всіх етапах експерименту. Різниця особливо помітна при 50 сусідах, де індекс

Жаккара суттєво випереджає інші метрики. Кореляція Спірмена також показує хороші результати, але трохи поступається.

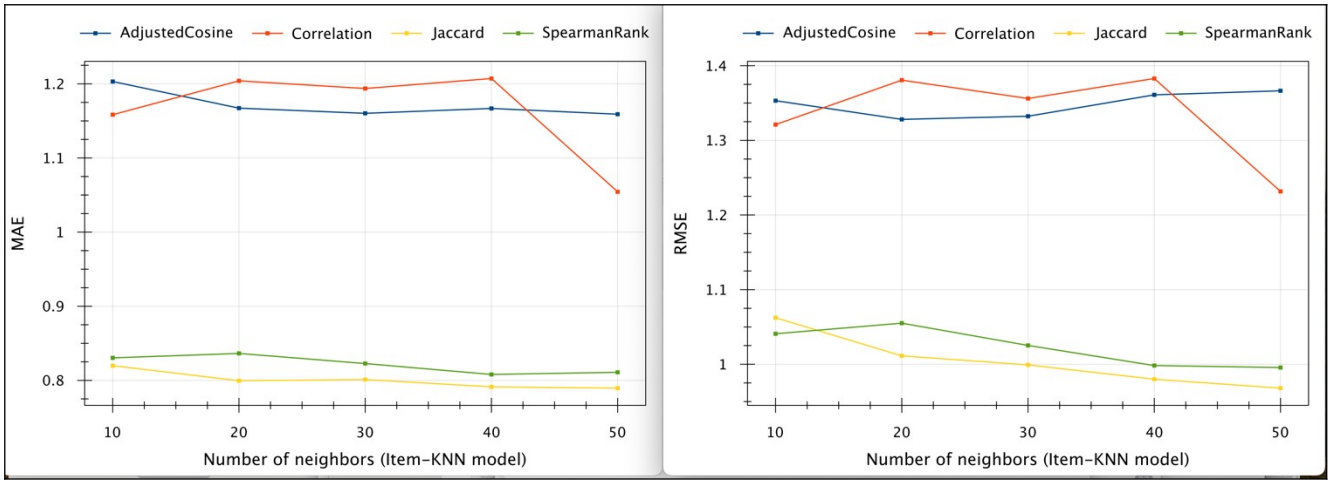

Рисунок 6.2 – Значення помилок MAE та RMSE для item-KNN моделі в залежності від обраної метрики подібності (рисунок виконано самостійно)

Метрики кореляції та косинусної подібності, навпаки, дають вищі значення помилок і демонструють менш стабільну динаміку. Отже, у рамках дослідження можна зробити висновок, що саме індекс Жаккара є найточнішою метрикою подібності для моделі item-KNN.

6.2.3 Оцінка дослідження метрик подібності

Після того як було обрано метрики подібності з найменшими похибками для моделей KNN, можна зробити проміжні висновки щодо результатів дослідження. Найперше, що помітно – це загальна тенденція до зменшення похибок із збільшенням кількості сусідів. Це цілком логічно, адже що більше сусідів враховується, то більше інформації має система, а значить – точність прогнозів покращується. Цікаво, що ця тенденція однакова як для MAE, так і для RMSE, бо обидві метрики мають схожий математичний підхід.

Щодо кореляції Пірсона, можна сказати, що вона не дає добрих результатів на нашому діапазоні сусідів. У більшості випадків її похибки вищі за 1, особливо в item-KNN, де значення майже не змінюється протягом усього діапазону. Для user-KNN навіть помітне зростання помилки наприкінці, що вважається

негативним результатом. Проте для моделей item-KNN ситуація трохи краща – там значення повільно зменшується, тож, можливо, на більших діапазонах кореляція Пірсона покаже себе краще.

Косинусна подібність виявилася малоефективною для моделей KNN. Це пояснює, чому в реальних системах її часто використовують саме в таких випадках. Індекс Жаккара, навпаки, дуже добре працює з моделями KNN, особливо з item-KNN, де він показав найкращі результати серед усіх метрик. Кореляція Спірмена показала досить стабільні й хороші результати майже у всіх випадках. Якщо обирати одну універсальну метрику, яка підходить для більшості моделей – то це, скоріше за все, була б саме вона.

6.3 Проведення загального дослідження

Після того як ми дослідили різні метрики подібності, вдалося визначити найкращі варіанти для кожної з моделей. Для user-KNN найточніші результати показала кореляція Спірмена, для item-KNN – індекс Жаккара. Саме ці варіанти ми будемо порівнювати з моделями PMF та NCF за допомогою показників MAE та RMSE. У PMF та NCF немає такого параметра, як кількість сусідів, але в них є схожа характеристика – кількість прихованих факторів. Тому для них також обрано діапазон значень від 10 до 50 з кроком 10, як і для інших моделей. Усі отримані значення похибок для кожної моделі можна побачити на рисунку 6.4.

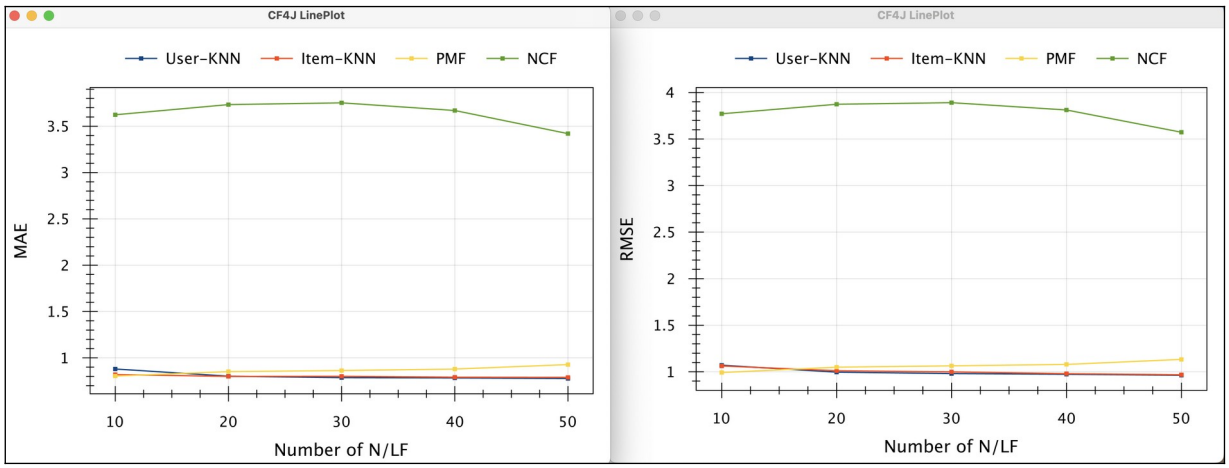

Рисунок 6.4 – Значення похибок MAE та RMSE для усіх моделей (рисунок виконано самостійно)

На графіку чітко видно, що модель NCF показує значно більші значення похибок порівняно з іншими моделями. Це дозволяє зробити висновок, що нейронна колаборативна фільтрація не дуже підходить для невеликих рекомендаційних систем, у яких кількість прихованих факторів обмежена до 50. У таких умовах її точність суттєво поступається іншим підходам. Інші моделі демонструють набагато кращі результати, і значення помилок у них приблизно на одному рівні. Для моделі PMF простежується інша тенденція – хоча зміни не різкі, але з кожним кроком похибка поступово зростає. Це може свідчити про те, що при ще більшій кількості факторів точність моделі буде погіршуватись. Щоб детальніше розглянути поведінку решти моделей, далі наведено збільшений фрагмент графіка (див. рис. 6.5).

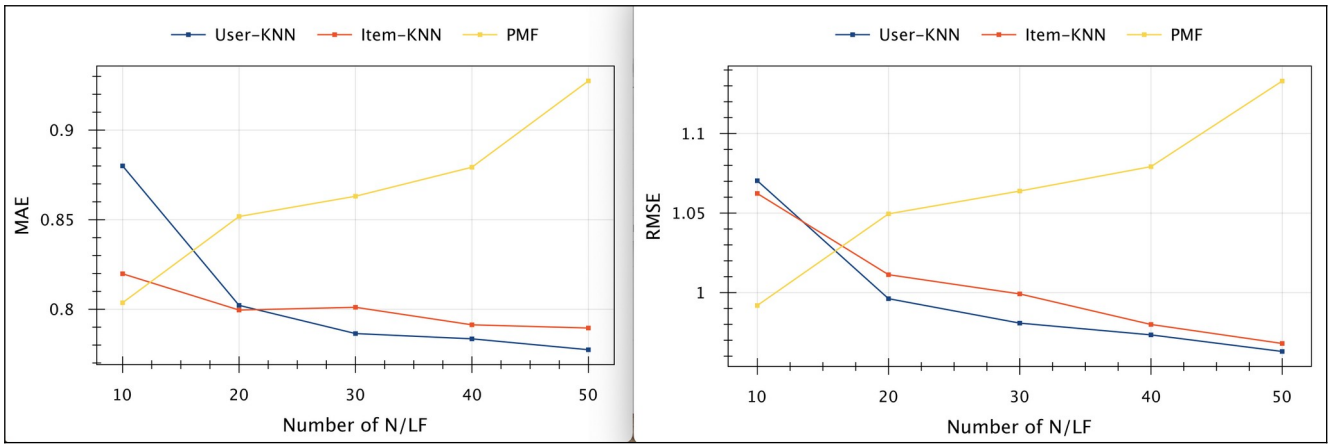

Рисунок 6.5 – Значення похибок MAE та RMSE для усіх моделей, крім NCF (рисунок виконано самостійно)

На збільшеному графіку чіткіше помітно, що зі зростанням кількості прихованих факторів похибка в моделі PMF все більше відрізняється від інших моделей. Наприкінці діапазону ця різниця стає вже досить значною, що може негативно вплинути на точність рекомендацій. У той час як значення помилок для user-KNN та item-KNN залишаються майже однаковими між собою. Щоб краще оцінити різницю, далі розглянемо точні числові значення похибок на рисунках 6.6 і 6.7.

Number of N/LF	User-KNN	Item-KNN	PMF	NCF
10	0.8800	0.8198	0.8037	3.1453
20	0.8022	0.7996	0.8518	3.6075
30	0.7864	0.8011	0.8631	3.4275
40	0.7835	0.7913	0.8793	3.4654
50	0.7774	0.7895	0.9274	3.5744

Рисунок 6.6 – Абсолютні значення похибки MAE для усіх моделей (рисунок виконано самостійно)

Number of N/LF	User-KNN	Item-KNN	PMF	NCF
10	1.0704	1.0623	0.9919	3.3168
20	0.9962	1.0113	1.0495	3.7533
30	0.9808	0.9992	1.0639	3.5802
40	0.9734	0.9800	1.0792	3.6153
50	0.9630	0.9680	1.1330	3.7199

Рисунок 6.7 – Абсолютні значення похибки RMSE для усіх моделей (рисунок виконано самостійно)

З графіків видно, що при збільшенні кількості сусідів точність у моделей user-KNN та item-KNN стає майже однаковою – різниця в похибках у кінці

діапазону складає менше ніж 0,01. Це означає, що між цими моделями складно обрати найкращу лише за точністю. У такому випадку головне – орієнтуватися на те, яка саме система рекомендацій буде впроваджуватись. Наприклад, якщо система базується на взаємодії користувачів між собою, як у соцмережах, тоді краще підійде user-KNN. Якщо це онлайн-магазин, де важливо відстежувати, які товари переглядає або купує користувач, то логічніше використовувати item-KNN. Іншими словами, кожна модель покаже кращий результат у своїй сфері, якщо система може забезпечити їй достатньо даних. У нашому дослідженні, де не враховувався конкретний контекст застосування, ці моделі показали дуже схожу точність. Найголовніше – це правильно підібрана метрика подібності для кожної моделі, що ми якраз і зробили на початковому етапі аналізу.

ВИСНОВКИ

У межах магістерської роботи було проведено теоретичне й практичне дослідження методів колаборативної фільтрації, зокрема user-based, item-based, матричної факторизації (PMF) та нейронної колаборативної фільтрації (NCF). Метою дослідження було порівняння точності цих методів прогнозування в контексті рекомендаційних систем та виявлення оптимальних рішень для різних сценаріїв використання.

У процесі експериментів було визначено, що ефективність моделей напряму залежить як від обраної метрики подібності (у випадку user-KNN та item-KNN), так і від параметрів моделі (наприклад, кількості сусідів або прихованих факторів). Для моделей user-KNN найкращі результати було отримано при використанні кореляції Спірмена, а для item-KNN – індексу Жаккара. Ці метрики демонстрували найменші значення MAE та RMSE, що свідчить про вищу якість прогнозування.

Результати порівняння показали, що з традиційних моделей найкращі показники точності продемонструвала item-KNN з індексом Жаккара. Модель user-KNN з кореляцією Спірмена також показала хорошу стабільність. Модель PMF виявилась менш чутливою до параметрів, але зі збільшенням кількості прихованих факторів спостерігалось зростання похибок. Найменш точним методом виявилась модель NCF, що продемонструвала найвищі значення похибок у порівнянні з іншими моделями, особливо при невеликій кількості факторів.

Окремо варто зазначити, що без урахування предметної області застосування традиційні підходи – user-KNN, item-KNN – виявились приблизно однаковими за точністю на обраному діапазоні параметрів. Це означає, що в практичному використанні вибір між цими методами має базуватись не лише на точності, а й на специфіці даних та архітектурі самої рекомендаційної системи.

Загалом результати роботи підтверджують важливість правильного вибору як типу колаборативної моделі, так і метрики подібності, оскільки саме це забезпечує найвищу точність у прогнозуванні рекомендацій.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Лобур М. В., Шварц М. Є., Стех Ю. В. Моделі і методи прогнозування рекомендацій для колаборативних рекомендаційних систем // Вісник Національного університету «Львівська політехніка». Серія: Інформаційні системи та мережі. – 2018. – Вип. 901. – С. 68–75.

2. Bobadilla J., Ortega F., Hernando A., Gutiérrez A. Recommender systems survey // Knowledge-Based Systems. – 2013. – Vol. 46. – P. 109–132.

3. Koren Y., Bell R., Volinsky C. Matrix factorization techniques for recommender systems // Computer. – 2009. – Vol. 42, No. 8. – P. 30–37.

4. He X., Liao L., Zhang H., Nie L., Hu X., Chua T.S. Neural collaborative filtering // Proceedings of the 26th International Conference on World Wide Web. – 2017. – P. 173–182.

5. Sarwar B., Karypis G., Konstan J., Riedl J. Item-based collaborative filtering recommendation algorithms // Proceedings of the 10th International Conference on World Wide Web. – 2001. – P. 285–295.

6. Ricci F., Rokach L., Shapira B. Recommender Systems Handbook. – 2nd ed. – New York: Springer, 2015. – 1003 p.

7. Adomavicius G., Tuzhilin A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions // IEEE Transactions on Knowledge and Data Engineering. – 2005. – Vol. 17, No. 6. – P. 734–749.

8. Ekstrand M. D., Riedl J. T., Konstan J. A. Collaborative Filtering Recommender Systems. – Now Publishers Inc, 2011. – 131 p.

9. Resnick P., Varian H. R. Recommender systems // Communications of the ACM. – 1997. – Vol. 40, No. 3. – P. 56–58.

10. Zhang S., Yao L., Sun A., Tay Y. Deep learning based recommender system: A survey and new perspectives // ACM Computing Surveys. – 2019. – Vol. 52, No. 1. – P. 1–38.

11. Linden G., Smith B., York J. Amazon.com recommendations: Item-to-item collaborative filtering // IEEE Internet Computing. – 2003. – Vol. 7, No. 1. – P. 76–80.

12. Schafer J. B., Konstan J., Riedl J. Recommender systems in e-commerce // Proceedings of the 1st ACM Conference on Electronic Commerce. – 1999. – P. 158–166.

13. Su X., Khoshgoftaar T. M. A survey of collaborative filtering techniques // Advances in Artificial Intelligence. – 2009. – Vol. 2009. – Article ID 421425. – 19 p.

14. Breese J. S., Heckerman D., Kadie C. Empirical analysis of predictive algorithms for collaborative filtering // Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence. – 1998. – P. 43–52.

15. Herlocker J. L., Konstan J. A., Borchers A., Riedl J. An algorithmic framework for performing collaborative filtering // Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. – 1999. – P. 230–237.

16. Gantner Z., Drumond L., Freudenthaler C., Rendle S., Schmidt-Thieme L. Learning attribute-to-feature mappings for cold-start recommendations // Data Mining and Knowledge Discovery. – 2010. – Vol. 21, No. 3. – P. 345–381.

17. Zhou Y., Wilkinson D., Schreiber R., Pan R. Large-scale parallel collaborative filtering for the Netflix prize // Algorithmic Aspects in Information and Management. – 2008. – P. 337–348.

18. Burke R. Hybrid recommender systems: Survey and experiments // User Modeling and User-Adapted Interaction. – 2002. – Vol. 12, No. 4. – P. 331–370.

19. Bell R. M., Koren Y. Lessons from the Netflix Prize Challenge // ACM SIGKDD Explorations Newsletter. – 2007. – Vol. 9, No. 2. – P. 75–79.

20. Shani G., Gunawardana A. Evaluating recommendation systems // Recommender Systems Handbook. – New York: Springer, 2015. – P. 265–308.

21. Байдак В.Є., Мазурова О.О., “Розробка комбінованого методу побудови рекомендаційної системи для онлайн-магазину електронних ігор”, Біоніка інтелекту, 2021, с. 56–63

22. Лещинський В.О., “Удосконалення методу колаборативної фільтрації з неявною зворотньою реакцією на основі ранжування негативних результатів у матриці вхідних даних”, Системи управління, навігації та зв'язку, 2018, с. 73–77

23. Чалий С.Ф., Лещинський В.О., “Метод формування рекомендацій із використанням темпоральних обмежень у ситуації циклічного "холодного старту" рекомендаційної системи”, EUREKA, Physics and Engineering, 2019, с. 34–40

24. Chalyi, S., Leshchynskyi, V., Leshchynska, I. "Method of forming recommendations using temporal constraints in a situation of cyclic cold start of the recommender system", EUREKA, Physics and Engineering, 2019, 2019(4), с. 34-40

25. S. Chalyi, V. Leshchynskyi, I. Leshchynska, “Modeling explanations for the recommended list of items based on the temporal dimension of user choice” - Control, navigation and communication systems, 2019, p. 97-101

26. Кириченко І. В., Терещенко Г. Ю., Шанідзе Н. О., Шубін І. Ю. Ідентифікація і трансформація контенту в системах електронного навчання : монографія. – Харків : ТОВ "В справі", 2021. – 136 с. – ISBN 978-617-7305-71-1. – DOI: 10.30837/978-617-7305-71-1. – Укр. мовою.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ ЗА НАУКОВИМИ НАПРЯМАМИ КЕРІВНИКА ТА НАУКОВЦІВ КАФЕДРИ ПРОГРАМНОЇ ІНЖЕНЕРІЇ

21. Байдак В.Є., Мазурова О.О., “Розробка комбінованого методу побудови рекомендаційної системи для онлайн-магазину електронних ігор”, Біоніка інтелекту, 2021, с. 56–63

22. Лещинський В.О., “Удосконалення методу колаборативної фільтрації з неявною зворотньою реакцією на основі ранжування негативних результатів у матриці вхідних даних”, Системи управління, навігації та зв'язку, 2018, с. 73–77

23. Чалий С.Ф., Лещинський В.О., “Метод формування рекомендацій із використанням темпоральних обмежень у ситуації циклічного "холодного старту" рекомендаційної системи”, EUREKA, Physics and Engineering, 2019, с. 34–40

24. Chalyi, S., Leshchynskyi, V., Leshchynska, I. "Method of forming recommendations using temporal constraints in a situation of cyclic cold start of the recommender system", EUREKA, Physics and Engineering, 2019, 2019(4), с. 34-40

25. S. Chalyi, V. Leshchynskyi, I. Leshchynska, “Modeling explanations for the recommended list of items based on the temporal dimension of user choice” - Control, navigation and communication systems, 2019, p. 97-101

26. Кириченко І. В., Терещенко Г. Ю., Шанідзе Н. О., Шубін І. Ю. Ідентифікація і трансформація контенту в системах електронного навчання : монографія. – Харків : ТОВ "В справі", 2021. – 136 с. – ISBN 978-617-7305-71-1. – DOI: 10.30837/978-617-7305-71-1. – Укр. мовою.