

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____ центр післядипломної освіти _____
(повна назва)

Кафедра _____ програмної інженерії _____
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти _____ другий (магістерський) _____

_____ Дослідження методів штучного інтелекту _____
_____ для розробки рекомендаційної системи в сфері аніме _____
(тема)

Виконала:

студентка 2 курсу, групи ІПЗЗдм-22-1

_____ Семенова Н.В. _____

(прізвище, ініціали)

Спеціальність 121 – Інженерія програмного
забезпечення

(код і повна назва спеціальності)

Тип програми освітньо-наукова

Керівник проф. Смеляков К.С.

(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри

_____ (підпис)

_____ З.В.Дудар _____

(прізвище, ініціали)

2024 р.

Харківський національний університет радіоелектроніки

Факультет _____ центр післядипломної освіти
 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)
 «___» _____ 2024 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові _____ Семеновій Наталії Валеріївни _____
 (прізвище, ім'я, по батькові)

1. Тема роботи «Дослідження методів штучного інтелекту для розробки рекомендаційної системи в сфері аніме»
 Затверджена наказом по університету від 22.04. 2024р. № 60Стз
2. Термін подання студентом роботи до екзаменаційної комісії 10.06.2024
3. Вихідні дані до роботи дослідити методи визначення персоналізованих рекомендацій аніме та оцінки їх ефективності шляхом поєднання різних методів штучного інтелекту
4. Перелік питань, що потрібно опрацювати в роботі
вступ, аналіз предметної галузі, методи вирішення проблеми, проведення експерименту та аналіз отриманих результатів, опис прототипу додатку, висновки.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз предметної галузі та постановка задачі	20.04 – 27.04.24	<i>виконано</i>
2	Аналіз методів пошуку рекомендацій	28.04 – 14.05.24	<i>виконано</i>
3	Планування експериментів	10.05 – 16.05.24	<i>виконано</i>
4	Проектування архітектури та розробка прототипу додатку	15.05 – 25.05.24	<i>виконано</i>
5	Експериментальні дослідження	15.05 – 20.05.24	<i>виконано</i>
6	Аналіз результатів експериментальних досліджень	20.05 – 23.05.24	<i>виконано</i>
7	Підготовка пояснювальної записки	05.05 – 01.06.24	<i>виконано</i>
8	Підготовка презентації та доповіді	01.06 – 10.06.24	<i>виконано</i>
9	Плагіат та нормоконтроль	10.06 – 11.06.24	<i>виконано</i>
10	Рецензування	11.06 – 15.06.24	<i>виконано</i>
11	Попередній захист	16.06.2024	<i>виконано</i>
12	Занесення диплома в електронний архів	17.06.2024	<i>виконано</i>
13	Допуск до захисту у зав. кафедри	18.06.2024	<i>виконано</i>

Дата видачі завдання 20 квітня 2024р.

Студентка _____
(підпис)

_____ Семенова Н.В.

Керівник роботи _____
(підпис)

_____ проф. Смеляков К.С.
(посада, прізвище, ініціали)

РЕФЕРАТ / ABSTRACT

Пояснювальна записка містить: 56 с., 14 рис., 1 табл., 15 джерел.

АНИМЕ, ВАГОВЕ СЕРЕДНЄ, ГІБРИДНИЙ ПІДХІД, ДАТАСЕТ, КОЛАБОРАТИВНА ФІЛЬТРАЦІЯ, КОРИСТУВАЧ, ПРОГНОЗУВАННЯ, РЕКОМЕНДАЦІЙНА СИСТЕМА, ШТУЧНИЙ ІНТЕЛЕКТ.

Об'єктом дослідження є формування рекомендацій в сфері аніме.

Метою дослідження є розробка та апробація рекомендаційної системи, заснованої на алгоритмах штучного інтелекту, для персоналізації вибору контенту в сфері аніме.

У дослідженні буде застосовано такі методи, як математичне моделювання для розробки алгоритмів рекомендаційної системи, емпіричний аналіз для збору та обробки даних про вподобання користувачів.

У результаті виконання роботи буде отримано удосконалений метод, що підвищуватиме точність надання аніме-рекомендацій для користувачів.

ANIME, WEIGHT AVERAGE, HYBRID APPROACH, DATASET, COLLABORATIVE FILTERING, USER, PREDICTION, RECOMMENDATION SYSTEM, ARTIFICIAL INTELLIGENCE.

The object of the research is forming recommendations in the field of anime.

The purpose of the work is to develop and test a recommendation system based on artificial intelligence algorithms to personalize the choice of content in the field of anime.

The study will apply methods such as mathematical modeling to develop recommendation system algorithms, empirical analysis to collect and process data on user preferences.

As a result of the work, an improved method will be obtained, which will increase the accuracy of providing anime recommendations to users.

Я, Семенова Наталія Валеріївна, студентка гр. ІІЗЗдм-22-1, здобувач вищої освіти на другому (магістерському) рівні кафедри «Програмна інженерія», заявляю: моя кваліфікаційна робота на тему «Дослідження методів штучного інтелекту для розробки рекомендаційної системи в сфері аніме», 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 EIArKhNURE. Всі запозичення з друкованих та електронних джерел мають відповідні посилання.

Я ознайомена 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.

ЗМІСТ

Вступ.....	7
1 Аналіз предметної галузі.....	9
1.1 Аналіз предметної галузі.....	9
1.2 Постановка задачі.....	12
2 Аналіз методів штучного інтелекту.....	14
2.1 Колаборативна фільтрація.....	14
2.2 Фільтрація на основі вмісту.....	19
2.3 Розробка алгоритму та поєднання результатів.....	20
3 Опис прийнятих програмних рішень.....	23
3.1 UML-проекування програмного забезпечення.....	23
3.2 Опис бази даних та датасету.....	25
3.3 Реалізація гібридного методу.....	26
3.4 Опис мобільного додатку.....	29
4 Оцінка ефективності гібридного методу.....	33
4.1 Вибір підходу для проведення експерименту.....	33
4.2 Проведення експерименту та аналіз результатів.....	34
Висновки.....	39
Перелік джерел посилань.....	40
Додаток А Перелік джерел посилання за науковими напрямками керівника та науковців кафедри програмної інженерії.....	42
Додаток Б Звіт результатів перевірки на унікальність тексту в базі ХНУРЕ.....	43
Додаток В Апробація роботи.....	44
Додаток Г Слайди презентації.....	47
Додаток Д Експертний висновок результатів перевірки кваліфікаційної роботи на відповідність оформлення вимогам ДСТУ 3008:2015.....	56

ВСТУП

У сучасному світі аніме стало не просто розвагою, але й культурним явищем, яке поширюється по всьому світові, залучаючи десятки мільйонів прихильників різного віку та культурного багажу. Розвиток та популяризація аніме призводить до потреби у вдосконаленні методів рекомендації контенту, щоб задовольнити різноманітні запити глядачів. Значення штучного інтелекту у сфері рекомендаційних систем постійно зростає, оскільки ці системи забезпечують персоналізований підхід до вибору контенту.

Рекомендаційні системи знайшли широке застосування у багатьох галузях, зокрема у стрімінгових сервісах, інтернет-магазинах, соціальних мережах тощо. У сфері аніме дані системи також мають велике значення, оскільки дозволяють користувачам відкривати для себе нові фільми та серіали, що відповідають їхнім потребам.

Як показує статистика, кількість нових аніме-серіалів та повнометражних аніме, що виходять кожен рік, продовжує збільшуватися, ускладнюючи вибір для користувачів. На кінець 2023 року було випущено понад 300 нових екранізацій, що на 20% більше, ніж у попередньому році [1]. Це створює потребу в розробці більш адаптивних та інтелектуальних систем, які можуть ефективно аналізувати та рекомендувати контент, заснований на індивідуальних перевагах різних користувачів.

Актуальність даної роботи обумовлена зростаючою популярністю аніме на світовому ринку, кількістю випуску нових аніме-серіалів та фільмів та необхідністю удосконалення методів вибору контенту для забезпечення кращого користувацького досвіду. Це дослідження спрямоване на виявлення та реалізацію найбільш ефективних методів штучного інтелекту для створення рекомендаційної системи, яка зможе задовольнити унікальні запити глядачів.

Метою магістерської роботи є дослідження існуючих методів штучного інтелекту при розробці рекомендаційних систем у сфері аніме. Основними завданнями роботи є дослідження сучасних алгоритмів рекомендаційних систем та їх застосування у сфері аніме, розробка та реалізація методів для покращення

точності рекомендацій, оцінка ефективності запропонованих методів на реальних даних.

Об'єктом дослідження є процеси та явища, пов'язані з рекомендаційними системами у сфері аніме. Предметом дослідження є методи штучного інтелекту, які використовуються для створення рекомендаційних систем та їх застосування до аніме-контенту.

Методи дослідження включають аналіз існуючих алгоритмів, розробку нових підходів та їх тестування на основі реальних даних. Зокрема, буде розглянуто колаборативну фільтрацію, фільтрацію на основі вмісту, гібридні системи.

Практичне значення одержаних результатів полягає у можливості їх застосування для покращення рекомендаційних систем у сервісах, що спеціалізуються на аніме. Це дозволить забезпечити користувачів більш точними та персоналізованими рекомендаціями, що підвищить їх задоволеність від користування сервісами.

За результатами даної роботи було підготовлено та опубліковано тези на 28-ому міжнародному молодіжному форумі «Радіоелектроніка та молодь у XXI столітті».

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Аналіз предметної галузі

Аніме – це унікальне явище, яке вийшло далеко за межі Японії та здобуло світову популярність і визнання. Його різноманітні жанри та стилі приваблюють широку аудиторію, включаючи не лише молодь, але й дорослих глядачів. Відмінні художні особливості, глибокі сюжетні лінії та комплексні персонажі роблять аніме універсальною формою розваги і культурного вираження.

На рисунку 1.1 зображено графік, який наглядно ілюструє концепцію «довгого хвоста», згідно з якою загальна сума всіх менш відомих продуктів дорівнює сумі найпопулярніших [2].

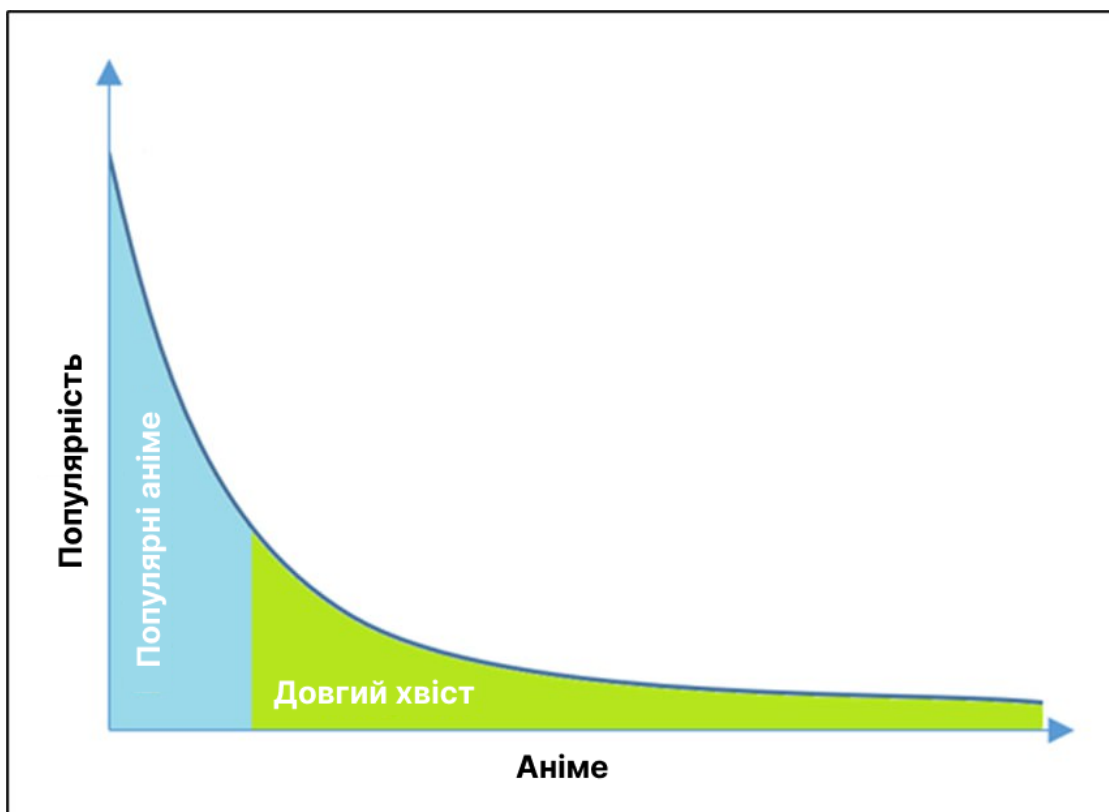


Рисунок 1.1 – Графік співвідношення популярності та аніме (рисунок виконаний самостійно)

Поява інтернету дозволила авторам контенту безпосередньо взаємодіяти з аудиторією, що значно спростило процес публікації та розповсюдження. Водночас медіа почало активно розширюватися, що привело до зростання популярності різних видів контенту. Аніме, як частина медіапростору, стало

набагато доступнішим для глобальної аудиторії, що сприяло його значному поширенню та популярності.

Згідно з останніми звітами, аніме-індустрія продемонструвала стійке зростання, особливо у сегментах стрімінгових платформ і цифрового дистрибуції. Це призвело до збільшення кількості доступного контенту та різноманітності жанрів. Стрімінгові сервіси, такі як Crunchyroll і Netflix, активно інвестують у виробництво та ліцензування аніме, розширюючи доступ до цього жанру для міжнародної аудиторії [3].

Таке розширення вимагає більш розвинутих методів рекомендації, щоб глядачі могли ефективно орієнтуватися в величезному обсязі контенту. Ефективність рекомендаційної системи зазвичай оцінюється на основі таких факторів, як точність, різноманітність і випадковість наданих рекомендацій.

Рекомендаційна система – це тип програмного забезпечення або алгоритму, призначений для надання персоналізованих пропозицій або рекомендацій користувачам. Основна мета рекомендаційних систем – допомогти користувачам знайти релевантні та цікаві предмети з великої кількості варіантів, таких як продукти, фільми, музика, статті чи інші типи контенту.

Традиційні методи, такі як проста класифікація за жанрами, вже не є достатньо ефективними. Сучасні вимоги до рекомендаційних систем змушують розробників звертатися до більш складних методів штучного інтелекту, що забезпечують персоналізацію та врахування унікальних інтересів кожного глядача.

У цьому контексті, розробка ефективних рекомендаційних систем, спеціалізованих на аніме, є критично важливою. Ці системи можуть допомогти користувачам знайти твори, що відповідають їхнім уподобанням серед величезної кількості доступного контенту. З огляду на швидке зростання обсягу аніме-контенту, вибір відповідного твору може бути непростим завданням. Тому, методи штучного інтелекту, що забезпечують персоналізовані рекомендації, є значущими для підвищення користувацького досвіду.

Рекомендаційні системи використовують різноманітні методи та алгоритми, такі як машинне навчання, інтелектуальний аналіз даних і штучний інтелект, для аналізу вподобань і закономірностей користувачів. Вони часто використовують такі дані, як оцінки користувачів, історія перегляду, пошукові запити, щоб створювати персоналізовані рекомендації. Ефективність рекомендаційної системи зазвичай оцінюється на основі таких факторів, як точність, різноманітність і випадковість наданих рекомендацій.

Процес рекомендацій складається з двох основних компонентів: користувачів та елементів. Його мета – знайти відповідні збіги між користувачами та об'єктами. У цьому процесі використовується метод прогнозування вартості продукту для користувачів. Мета полягає в тому, щоб вибрати елемент для кожного користувача, який максимізує його задоволення або користь.

При розробці алгоритму дуже важливо враховувати різні важливі компоненти рекомендаційної системи. Було виділено наступні аспекти, які відіграють ключову роль у процесі розробки:

- аналіз предметної області;
- збір та аналіз даних;
- вибір та оптимізація алгоритму;
- формулювання завдання або проблеми рекомендації;
- рівень персоналізації та кастомізації.

Під аналізом предметної області мається на увазі глибоке розуміння конкретної сфери або галузі, в якій буде впроваджена система рекомендацій. Розуміння предметної області дозволяє адаптувати рекомендації до унікальних потреб і вподобань цього конкретного контексту.

Якість та кількість наявних даних суттєво впливає на ефективність рекомендаційної системи. Ретельний збір, попередня обробка та аналіз даних є важливими кроками для забезпечення того, щоб система могла надавати точні та релевантні рекомендації.

Вибір правильних алгоритмів рекомендацій має вирішальне значення. Різні алгоритми, такі як колаборативна фільтрація, фільтрація на основі вмісту або гібридні підходи, мають певні сильні та слабкі сторони.

Також, важливо чітко визначити мету рекомендації. Це включає визначення того, що саме рекомендує система, а також розгляд будь-яких додаткових обмежень або вимог, які необхідно враховувати.

Визначення відповідного ступеня персоналізації є вирішальним рішенням. Деякі системи виграють від високого рівня персоналізації, адаптуючи рекомендації до індивідуальних уподобань, тоді як інші можуть віддавати перевагу більш широкому та узагальненому підходу для охоплення ширшої аудиторії.

Ретельно розглянувши всі ключові аспекти, можна створити систему рекомендацій, яка не тільки надає точні та релевантні аніме-пропозиції, але й відповідає конкретним цілям та вимогам галузі. Такий цілісний підхід гарантує, що система добре оснащена для задоволення потреб користувачів.

1.2 Постановка задачі

Головною метою кваліфікаційної роботи є дослідження існуючих методів штучного інтелекту при розробці рекомендаційної системи у сфері аніме. Основними завданнями роботи є дослідження сучасних алгоритмів рекомендаційних систем та їх застосування у сфері аніме, розробка та реалізація методів для покращення точності рекомендацій, оцінка ефективності запропонованих методів на реальних даних.

В результаті роботи буде проведено аналіз та порівняння різних підходів до створення систем рекомендацій аніме, включаючи колаборативну фільтрацію на основі користувачів і аніме, а також фільтрацію на основі вмісту. Буде виявлено сильні та слабкі сторони кожного методу.

Основною проблемою цих підходів є відсутність комплексного підходу, який би об'єднував переваги кожного методу для надання більш точних і персоналізованих рекомендацій користувачам.

У роботі необхідно дослідити методи оцінки вподобань користувачів на підставі даних про перегляди, оцінки та характеристики аніме, а також експериментально довести можливість формування загальної системи рекомендацій на основі різних типів даних.

В процесі виконання роботи необхідно розглянути наступні питання:

- аналіз методу колаборативної фільтрації на основі користувачів;
- аналіз методу колаборативної фільтрації на основі об'єктів;
- аналіз фільтрації на основі вмісту;
- нормалізація даних, приведення до єдиного формату;
- поєднання результатів, методом зваженого середнього;
- оцінка ефективності методу.

Після розглядання даних питань, буде створено прототип додатку, який буде надавати аніме-рекомендації для користувачів, використовуючи комплексний підхід, що поєднує різні методи для досягнення найвищої точності та релевантності рекомендацій.

2 АНАЛІЗ МЕТОДІВ ШТУЧНОГО ІНТЕЛЕКТУ

2.1 Колаборативна фільтрація

Колаборативна фільтрація ґрунтується на ідеї, що користувачі, у яких збігався вибір контенту в минулому, як правило, погоджуються з вибором і в майбутньому [4]. Для надання рекомендацій даний метод спирається на взаємодію користувача безпосередньо з контентом, наприклад, поставленою оцінкою переглянутого аніме.

Цей метод використовує історичні взаємодії між користувачами та елементами, що зберігаються в так званій «матриці взаємодії між користувачем і елементом». У цій матриці кожен рядок представляє користувача, а кожен стовпець – елемент (див. табл. 2.1).

Таблиця 2.1 – Матриця взаємодії між користувачем і елементом (таблиця виконана самостійно)

Глядач \ Аніме	«Берсерк»	«Боку но Піко»	«Євангеліон»	«Ковбой Бібоп»
Яна	2	5	?	2
Андрій	?	5	5	2
Настя	4	?	5	?

Алгоритми колаборативної фільтрації працюють, аналізуючи ці моделі взаємодії, щоб виявити користувачів зі схожими смаками та інтересами. Вони можуть генерувати аніме-рекомендації, які можуть здатися привабливими для користувача, навіть якщо він раніше не взаємодіяв із цими творами.

Відгуки користувачів про те, що їм сподобався товар, діляться на дві категорії: явні та неявні оцінки.

Явні оцінки – це прямі та чіткі вказівки на вподобання користувача щодо певного товару. Зазвичай вони встановлюються за числовою шкалою (наприклад, від 1 до 5 зірок) або якимось іншим заздалегідь визначеним набором параметрів.

Неявні оцінки явно не надаються користувачами як прямий зворотний зв'язок. Натомість вони визначаються на основі поведінки користувачів, як від часу перегляду, історії перегляду, пошукових запитів тощо. Ці дії використовуються для визначення інтересу або переваг користувача до певного контенту.

Існують два методи колаборативної фільтрації: на основі користувачів та на основі елементів (див. рис. 2.1).

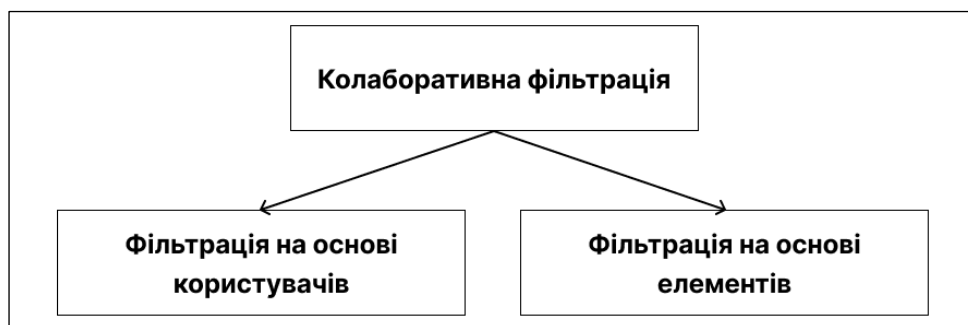


Рисунок 2.1 – Методи колаборативної фільтрації (рисунок виконаний самостійно)

Колаборативна фільтрація на основі користувачів передбачає виявлення користувачів, які демонструють схожість у своїх уподобаннях з іншими, які оцінили або виявили інтерес до подібних продуктів (див. рис. 2.2).

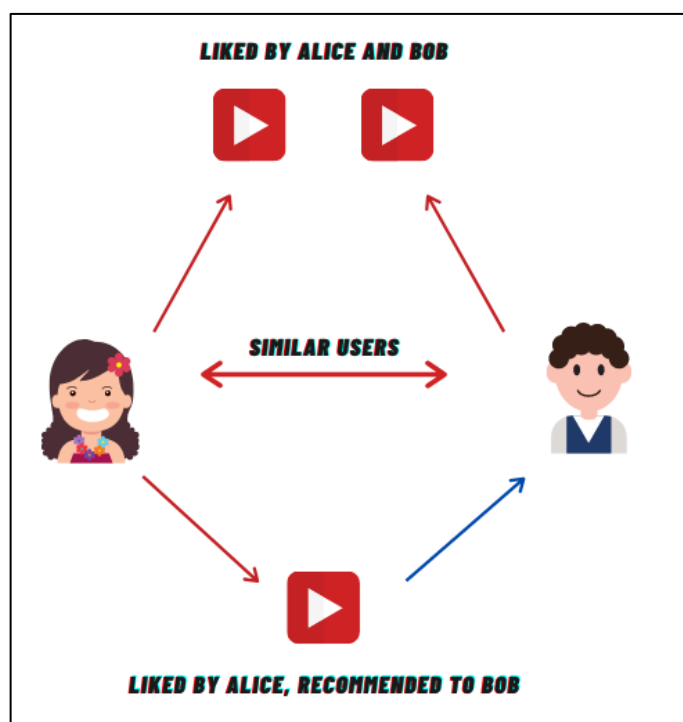


Рисунок 2.2 – Фільтрація на основі користувачів (за даними [5])

Колаборативна фільтрація на основі користувачів є досить розповсюдженим методом, який визначає користувачів зі схожими вподобаннями на основі їх оцінок і запитів. Цей підхід включає в себе кілька ключових етапів.

Спочатку треба визначити подібність між будь-якими двома користувачами. У контексті рекомендаційної системи під подібністю мається на увазі, наскільки схожі оцінки переглянутих аніме для двох користувачів. Вона може бути обчислена за допомогою таких показників, як подібність косинусів або коефіцієнт кореляції Пірсона [6]. В даному дослідженні буде розглянуто саме коефіцієнт кореляції Пірсона, який можна розрахувати за формулою 2.1.

$$\text{sim}(A, B) = \frac{\sum_{i=1}^n (A_i - \bar{A})(B_i - \bar{B})}{\sqrt{\sum_{i=1}^n (A_i - \bar{A})^2} \sqrt{\sum_{i=1}^n (B_i - \bar{B})^2}} \quad (2.1)$$

де A_i і B_i – елементи векторів A і B ,

\bar{A} і \bar{B} – середні значення векторів A і B .

Після визначення схожості між користувачами треба спрогнозувати оцінку для непереглянутих аніме за формулою 2.2.

$$p_{a,i} = \bar{r}_a + \frac{\sum_{u=1}^n (r_{u,i} - \bar{r}_u) w_{a,u}}{\sum_{u=1}^n w_{a,u}} \quad (2.2)$$

де \bar{r}_a – середня оцінка користувача a ,

$w_{a,u}$ – ступінь схожості між користувачем a та користувачем u ,

$r_{u,i}$ – реальний рейтинг користувача u для елемента i ,

\bar{r}_u – середній рейтинг користувача u .

Цей етап має вирішальне значення для надання користувачам персоналізованих рекомендацій. По суті, він заповнює прогалини в матриці взаємодії з користувачем.

Колаборативна фільтрація на основі елементів базується на виявленні елементів, які мають схожість з іншими елементами, що визначається оцінками користувачів. Наприклад, якщо користувач надає перевагу певному жанру аніме, цей підхід пропонує схожі екранізації на основі відомих уподобань.

На рисунку 2.3 можна побачити приклад, в якому кожне аніме виступає як окремий елемент.

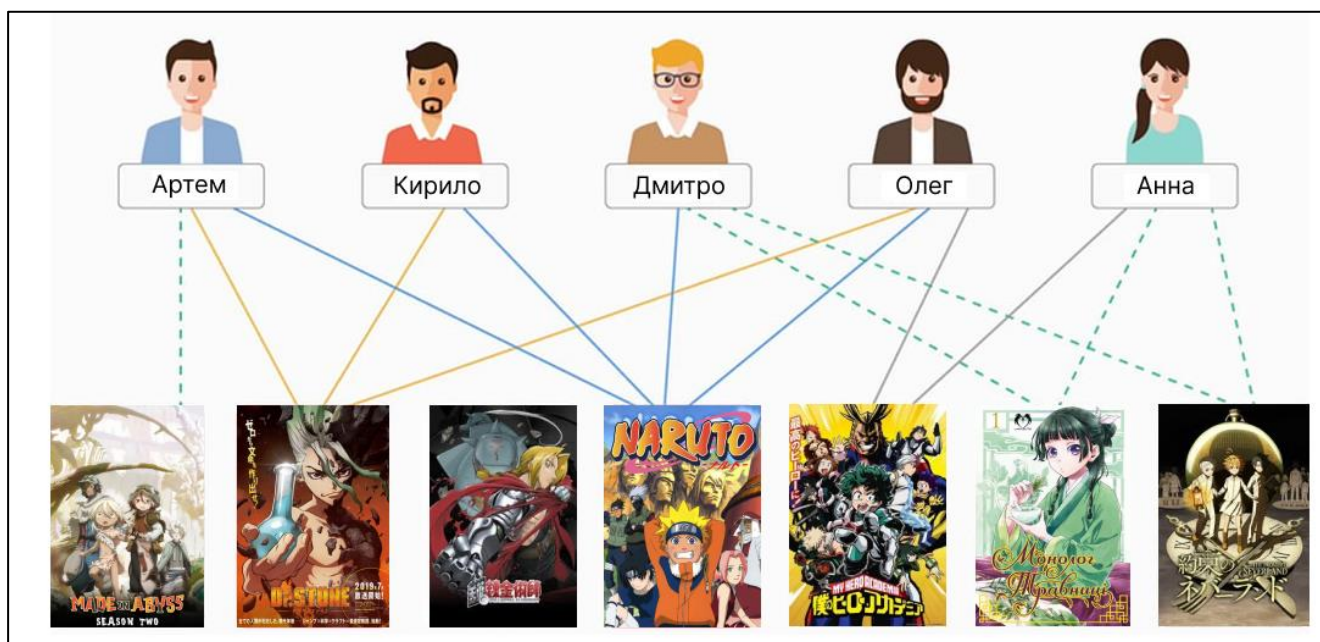


Рисунок 2.3 – Фільтрація на основі елементів (рисунок виконаний самостійно)

На відміну від колаборативної фільтрації на основі користувачів, яка зосереджена на пошуку схожих оцінок, фільтрація на основі елементів використовує інший підхід. Вона зіставляє кожне з оцінених глядачем аніме із подібними непереглянутими творами, а потім включає ці схожі аніме до списку рекомендацій [7].

Процес фільтрації на основі елементів складається з декількох етапів для прогнозування рейтингу, який користувач «u» може присвоїти аніме «m».

Спочатку треба визначити набір елементів, схожих на аніме m . Міра подібності розраховується за допомогою таких показників, як коефіцієнт Пірсона, який було розглянуто раніше.

Вибір показника подібності може вплинути на якість рекомендацій. Різні показники можуть бути більш прийнятними для певних типів продуктів або жанрів.

Далі необхідно розрахувати очікувану оцінку аніме «m» користувачем «u». Це досягається шляхом взяття середньозваженої оцінки оцінок, які користувач дав найближчим до нього фільмам. Чим більше елемент схожий на «m», тим більший вплив матиме його рейтинг на розрахунок. Середньозважене значення гарантує, що рейтинги найбільш схожих товарів матимуть більшу вагу в прогнозі, пропонуючи більш персоналізовані рекомендації.

Після цього порекомендувати користувачеві n кращих фільмів. На основі розрахованих оцінок формується список з n кращих рекомендованих аніме, який представляється користувачеві.

Далі необхідно розрахувати очікувану оцінку аніме «m» користувачем «u». Це досягається шляхом об'єднання матриці подібності з минулою історією користувачів, щоб згенерувати рекомендацію. Цього легко досягти, застосовуючи рівняння колаборативної фільтрації на основі елементів за формулою 2.3.

$$score(u, i) = \frac{\sum_j^i sim(i, j)(r_{u,i} - \bar{r}_j)}{\sum_j^i sim(i, j)} + \bar{r}_i \quad (2.3)$$

де u – користувач, для якого генерується рекомендація,

i – елемент, про який йде мова, тобто чи слід рекомендувати аніме чи ні,

j – об'єкти, подібні до головного елементу i ,

\bar{r}_i – середня оцінка для елементу i .

З наведеної формули видно, що для розрахунку рейтингу рекомендації елемента «i» для користувача «u» необхідно підсумувати подібність елементів «i» і «j» з різницею оцінок, присвоєних користувачем «u» елемента «j», і середньої оцінки елемента «j». Потім розділити результат на суму подібностей елементів «i» і «j» і додати результат до середньої оцінки користувача.

Це створює матрицю рейтингу для користувачів і доступних аніме. Твори з найвищим балом можуть бути рекомендовані користувачеві.

Не зважаючи на те, що колаборативна фільтрація на основі користувачів або елементів є простими моделями, вони мають свої обмеження [8]. Наприклад, дані методи базуються на пам'яті, тобто значною мірою покладаються на останні оцінки користувачів і можуть бути інтенсивним з обчислювальної точки зору, оскільки вимагають безперервних обчислень спорідненості з користувачами або елементами.

Крім того, масштабованість може стати проблемою, особливо коли є велика база користувачів, оскільки кількість попарних порівнянь між користувачами зростає в геометричній прогресії. Для вирішення цієї проблеми, перед застосуванням алгоритму, треба очистити дані, щоб підвищити продуктивність і точність алгоритму. Основні параметри, за якими можна фільтрувати користувачів:

- мінімальна кількість переглянутих аніме: відсіяти користувачів, які подивилися менше 10 аніме;
- середня оцінка: відсіяти користувачів з дуже низькою або дуже високою середньою оцінкою, яка може свідчити про те, що дані нерелевантні;
- кількість днів, витрачених на перегляд: відсіяти користувачів, які витратили дуже мало часу на перегляд.

Незважаючи на ці недоліки, варто зазначити, що колаборативна фільтрація відіграла ключову роль у закладенні фундаменту для удосконаленої концепції надання рекомендацій. Цей підхід вплинув на розробку більш складних рекомендаційних систем, які усувають деякі з цих обмежень.

2.2 Фільтрація на основі вмісту

Фільтрація на основі вмісту базується на аналізі характеристик об'єктів, наприклад, жанру, студії або акторів у аніме, та рекомендує контент, схожий на те, що користувач переглядав раніше.

Системи рекомендацій на основі вмісту покладаються на вилучення та представлення різних ознак, пов'язаних з елементами. Ці функції охоплюють різні медіаформати, такі як текст, зображення та відео.

Основне завдання полягає в тому, щоб розшифрувати конкретні ознаки, які визначають симпатію користувача до предмета. Користувачі створюють великий контент в Інтернеті, надаючи велику кількість інформації, яку можна використовувати для розуміння їхніх уподобань. Аналізуючи цей вміст, рекомендаційні системи прагнуть визначити явні вподобання користувачів і зіставити елементи, які відповідають цим уподобанням.

Рекомендаційні системи на основі контенту найкраще працюють у сферах, що характеризуються динамічним контентом [9, 10]. У випадках, коли колаборативна фільтрація ускладнена, найкраще підходять методи, засновані на вмісті. Здатність підлаштовуватися під змінні уподобання користувача та динамічний контент робить такий підхід дуже ефективним.

Окрім того, методи, засновані на вмісті, частково вирішують проблему з новими користувачами, у яких ще немає оцінених аніме, і тому застосувати колаборативну фільтрацію, яка базується на історії перегляду та оцінках, просто неможливо. Нові користувачі або об'єкти можуть бути описані на основі притаманних їм характеристик, що дозволяє вносити пропозиції навіть за відсутності великих даних. Ця гнучкість є помітною перевагою, особливо для аніме сфери, де новий контент з'являється кожен день.

2.3 Розробка алгоритму та поєднання результатів

Для об'єднання результатів трьох розглянутих методів у гібридний підхід, можна використати метод зваженого середнього.

Зважене середнє – метод обчислення середнього значення, в якому кожному елементу, в даному випадку, прогнозованій оцінці, надається певна коефіцієнт, який відображає його важливість [11]. На відміну від звичайного середнього, де всі елементи мають однаковий коефіцієнт, у ваговому середньому враховується, що деякі елементи можуть бути важливішими за інші.

Для обчислення вагового середнього треба скористатися загальною формулою 2.4.

$$\bar{x} = \sum_{i=1}^n w_i x_i \quad (2.4)$$

де x_i – значення кожного елемента,

w_i – вага, надана кожному елементу.

Кожен з методів має свої унікальні переваги: фільтрація на основі користувачів враховує схожість між користувачами, фільтрація на основі елементів аналізує схожість між аніме, а фільтрація на основі вмісту базується на характеристиках самого контенту. З метою забезпечення максимально точних і персоналізованих рекомендацій, комбінування цих методів дозволяє врахувати як вподобання користувачів, так і характеристики самих аніме [12].

Прогнозовані оцінки кожного з методів вже знаходяться в одній шкалі – від 1 до 5, тому нормалізація не потрібна. Це спрощує процес об'єднання результатів.

Ваги для кожного методу визначаються наступним чином: 0.6 – для результату фільтрації на основі користувачів, 0.3 – для фільтрації на основі елементів, 0.1 – для фільтрації на основі вмісту. Вибір таких коефіцієнтів обґрунтований важливістю кожного методу в контексті рекомендацій аніме.

Фільтрація на основі користувачів отримує найбільшу вагу, оскільки цей метод забезпечує високий рівень персоналізації, враховуючи вподобання схожих користувачів.

Фільтрація на основі елементів отримує вагу, оскільки цей метод ефективно визначає схожість між аніме, що дозволяє рекомендувати нові аніме на основі схожості з уже оціненими.

Фільтрація на основі вмісту отримує найменшу вагу, оскільки цей метод базується на середніх оцінках аніме, що не забезпечує високого рівня персоналізації, але все ж додає цінність завдяки аналізу аніме характеристик.

В результаті отримуємо формулу 2.5 для обчислення зваженої оцінки непереглянутого аніме.

$$\hat{r}_{ui} = 0.6 \cdot \hat{r}_i^{user-based} + 0.3 \cdot \hat{r}_i^{item-based} + 0.1 \cdot \hat{r}_i^{content-based} \quad (2.5)$$

де $\hat{r}_i^{user-based}$ – прогнозована оцінка за фільтрацією на основі користувачів,

$\hat{r}_i^{item-based}$ – прогнозована оцінка за фільтрацією на основі елементів,

$\hat{r}_i^{content-based}$ – прогнозована оцінка за фільтрацією на основі вмісту.

Таким чином, користувачеві буде рекомендовано аніме на основі зважених середніх оцінок всіх потенційних рекомендацій. Це забезпечує комплексний підхід до рекомендацій, враховуючи як історію взаємодій користувача, так і схожість контенту, та дозволяє надавати більш точні та релевантні рекомендації.

3 ОПИС ПРИЙНЯТИХ ПРОГРАМНИХ РІШЕНЬ

3.1 UML-проектування програмного забезпечення

Програмна система для рекомендацій аніме має одного актора: користувач. На рисунку 3.1 наведено Use-Case діаграму, яка описує функціонал програмної системи, доступний користувачеві.

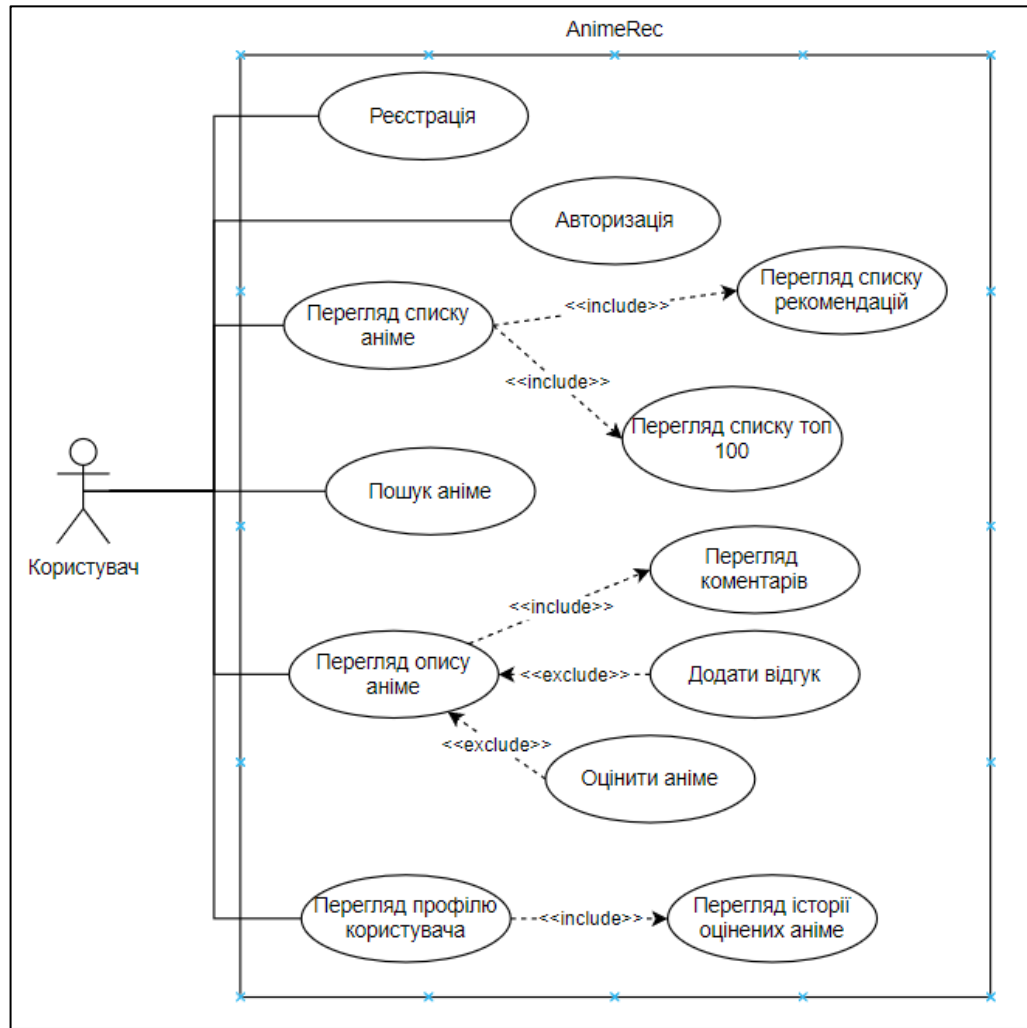


Рисунок 3.1 – Use-Case діаграма (рисунок виконаний самостійно)

Перед початком використання мобільного додатку користувач повинен авторизуватися в системі або зареєструватися, якщо ще не має облікового запису. Після авторизації користувач може переглядати аніме-рекомендації, засновані на його вподобаннях та історії переглядів.

Для користувачів передбачено наступний функціонал:

- реєстрація;

- авторизація;
- пошук аніме;
- перегляд списку аніме;
- перегляд списку рекомендованих аніме;
- перегляд списку топ 100 аніме;
- перегляд опису аніме;
- перегляд коментарів;
- додавання відгуку;
- оцінка аніме;
- перегляд профілю користувача;
- перегляд історії оцінених аніме.

Прототип складається з двох частин: iOS-додаток та веб-сервер.

На рисунку 3.2 можна побачити архітектуру проекту, представлену у вигляді діаграми розгортання.

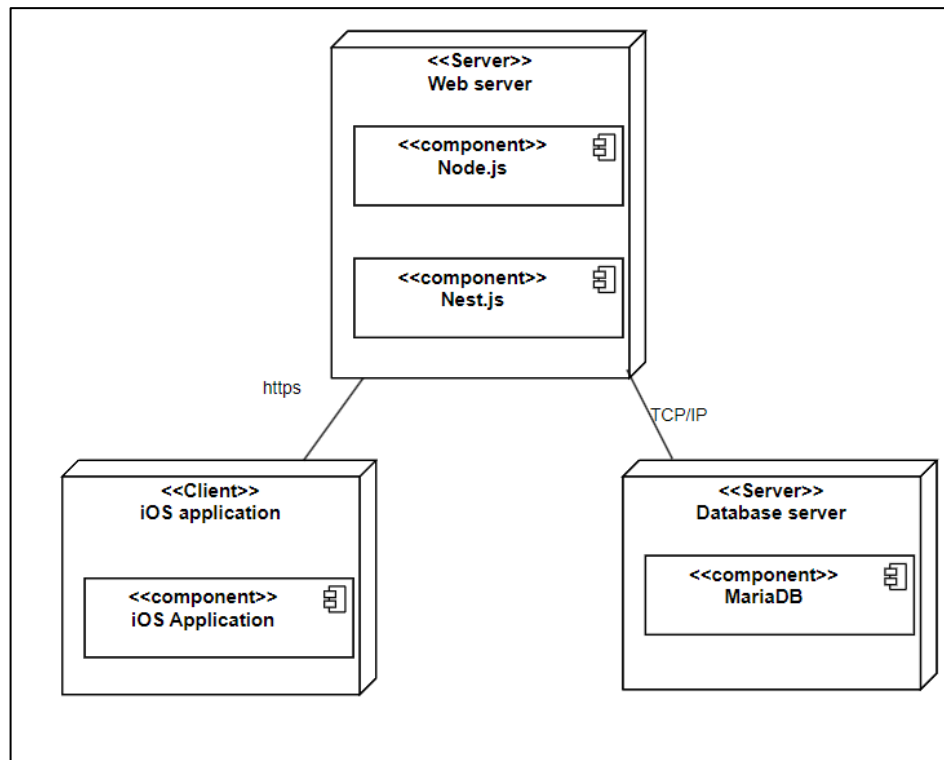


Рисунок 3.2 – Діаграма розгортання (рисунок виконаний самостійно)

На діаграмі розгортання показано взаємодію клієнта, тобто iOS-додатку, та веб-серверу, який складається безпосередньо з серверу та бази даних.

3.2 Опис бази даних та датасету

Для зберігання та обробки даних у прототипі використовується MariaDB.

Схема бази даних наведена на рисунку 3.3.

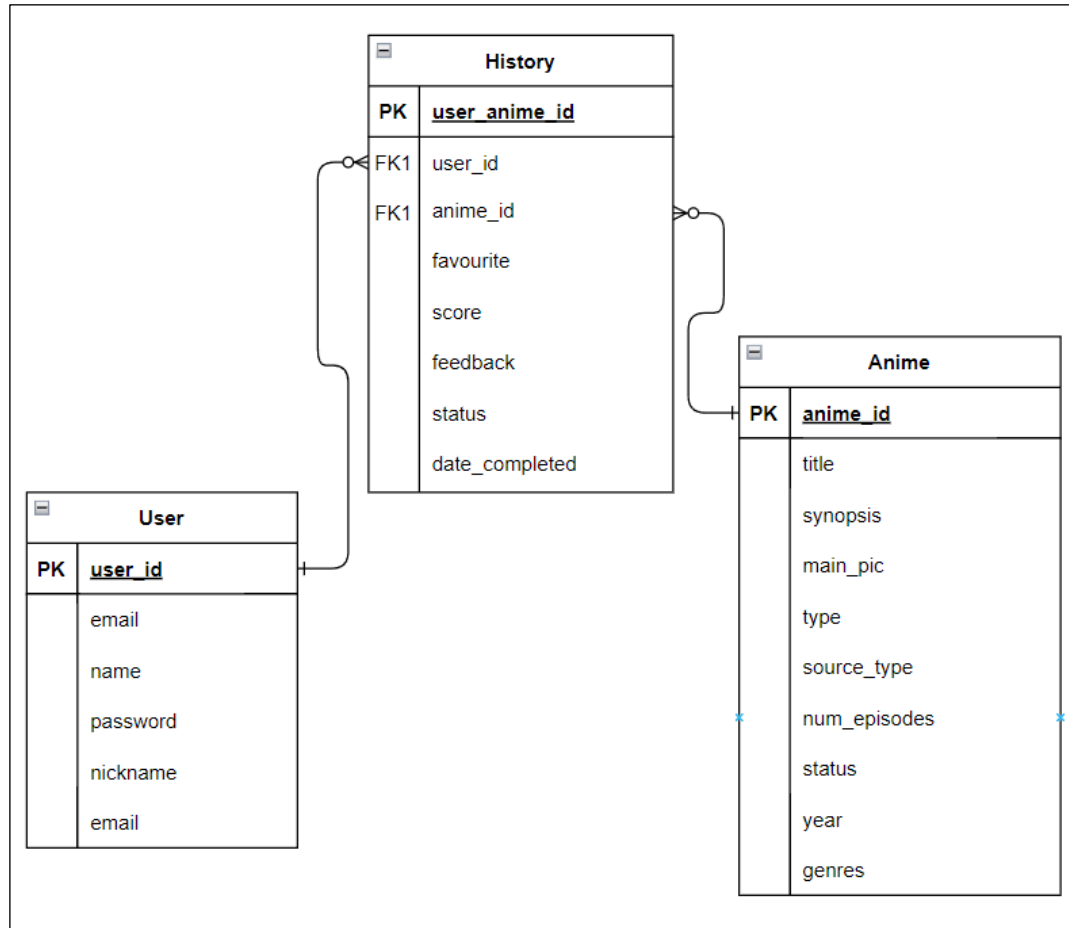


Рисунок 3.3 – Схема бази даних (рисунок виконаний самостійно)

Згідно зі схемою бази даних було створено три сутності: User, Anime, History.

Таблиця User відображає інформацію про користувачів системи. Сутність History відображає дані про взаємодію користувачів та контенту, тобто оцінки, фідбеки та статус перегляду.

Дані про аніме для тестування та розробки було взято з відкритого датасету на Kaggle від MyAnimeList – найбільшого онлайн-сервісу для аніме- та манга-спільноти [13].

Датасет містить наступну інформацію про 10 000+ аніме:

- anime_id;

- anime_url: посилання на сторінку аніме в myanimelist;
- title: назва аніме;
- synopsis: короткий опис аніме;
- main_pic: посилання на обкладинку;
- type: тип аніме, наприклад, повнометражний фільм, серіал тощо;
- source_type: джерело створення, наприклад манга;
- num_episodes: кількість епізодів;
- status: статус відображає, чи завершено аніме;
- studios: анімаційна студія, яка створила аніме;
- genres;
- score: середня оцінка.

Даний датасет та розроблена база даних дозволяє створити ефективну рекомендаційну систему, яка враховує як характеристики аніме, так і вподобання та поведінку користувачів. Використовуючи дані про оцінки, перегляди та відгуки, система може генерувати персоналізовані рекомендації, що відповідають індивідуальним вподобанням кожного користувача.

3.3 Реалізація гібридного методу

Розроблений гібридний метод пошуку аніме-рекомендацій було реалізовано на стороні серверу. Веб-сервер було реалізовано з використанням ядра Node.js та його фреймворку Nest.js. Далі наведено детальний опис реалізації алгоритму на JavaScript.

Перший крок – очищення даних. Для цього відфільтровуються користувачі з дуже малою кількістю переглядів аніме, щоб покращити продуктивність алгоритму. Приклад коду наведено нижче:

```
const filteredUsers = users.filter(user => user.num_completed >=
minAnimeWatched && user.mean_score >= 3 && user.mean_score <= 8);
const filteredUserIds = new Set(filteredUsers.map(user =>
user.user_id));
const filteredUserAnime = userAnimeData.filter(entry =>
filteredUserIds.has(entry.user_id));
```

Наступний крок включає реалізацію коефіцієнта кореляції Пірсона, який використовується в обох методах колаборативної фільтрації. Ця функція обчислює кореляцію між двома наборами даних (наприклад, оцінками двох користувачів або двох аніме). Приклад кода наведено нижче:

```
const pearsonCorrelation = (ratings1, ratings2) => {
  const commonItems = Object.keys(ratings1).filter(item => item in
ratings2);
  if (commonItems.length === 0) return 0;

  const mean1 = commonItems.reduce((sum, item) => sum +
ratings1[item], 0) / commonItems.length;
  const mean2 = commonItems.reduce((sum, item) => sum +
ratings2[item], 0) / commonItems.length;

  const numerator = commonItems.reduce((sum, item) => sum +
(ratings1[item] - mean1) * (ratings2[item] - mean2), 0);
  const denominatorA = Math.sqrt(commonItems.reduce((sum, item) =>
sum + Math.pow(ratings1[item] - mean1, 2), 0));
  const denominatorB = Math.sqrt(commonItems.reduce((sum, item) =>
sum + Math.pow(ratings2[item] - mean2, 2), 0));

  return denominatorA * denominatorB === 0 ? 0 : numerator /
(denominatorA * denominatorB);
};
```

Далі було реалізовано фільтрацію на основі користувачів. Першим кроком є обчислення схожості між користувачами за допомогою коефіцієнта кореляції Пірсона. Приклад коду наведено нижче:

```
const computeUserSimilarity = (ratings) => {
  const similarity = {};
  const userIds = Object.keys(ratings);

  userIds.forEach(user1 => {
    similarity[user1] = {};
    userIds.forEach(user2 => {
      if (user1 !== user2) {
        similarity[user1][user2] =
pearsonCorrelation(ratings[user1], ratings[user2]);
      }
    });
  });

  return similarity;
};
```

Наступний крок – знаходження топ-20 схожих користувачів для цільового користувача. Приклад коду наведено нижче:

```
const getTopSimilarUsers = (userId, userSimilarity, topN = 20) => {
  const similarities = Object.entries(userSimilarity[userId]);
  similarities.sort((a, b) => b[1] - a[1]);
  return similarities.slice(0, topN).map(entry => entry[0]);
};
```

Останній крок – прогнозування оцінок для цільового аніме на основі оцінок схожих користувачів. Приклад коду наведено нижче:

```
const predictRatingCF = (userRatings, similarityMatrix, targetUser,
targetItem) => {
  const targetUserRatings = userRatings[targetUser];
  const targetUserMean =
Object.values(targetUserRatings).reduce((sum, rating) => sum + rating, 0) /
Object.values(targetUserRatings).length;

  let numerator = 0;
  let denominator = 0;

  for (const user in userRatings) {
    if (user !== targetUser && userRatings[user][targetItem] !==
undefined) {
      const userRatingsMean =
Object.values(userRatings[user]).reduce((sum, rating) => sum + rating, 0) /
Object.values(userRatings[user]).length;
      const similarity = similarityMatrix[targetUser][user];
      numerator += similarity * (userRatings[user][targetItem]
- userRatingsMean);
      denominator += Math.abs(similarity);
    }
  }

  return denominator === 0 ? targetUserMean : targetUserMean +
(numerator / denominator);
};
```

Таким же чином було реалізовано колаборативну фільтрацію на основі елементів.

Наступний крок – реалізація фільтрації на основі вмісту. Прогнозування оцінок базується на середніх оцінках аніме.

Останній крок включає об'єднання результатів з різних методів за допомогою зваженого середнього, що дозволяє врахувати переваги кожного

підходу та досягти найвищої точності рекомендацій. Приклад коду наведено нижче:

```
const combineRecs = (userBased, itemBased, contentBased) => {
  const combined = {};
  const weights = { user: 0.6, item: 0.3, content: 0.1 };

  userBased.forEach(rec => {
    if (!combined[rec.anime]) combined[rec.anime] = 0;
    combined[rec.anime] += rec.score * weights.user;
  });

  itemBased.forEach(rec => {
    if (!combined[rec.anime]) combined[rec.anime] = 0;
    combined[rec.anime] += rec.score * weights.item;
  });

  contentBased.forEach(rec => {
    if (!combined[rec.anime]) combined[rec.anime] = 0;
    combined[rec.anime] += rec.score * weights.content;
  });

  const finalRecs = Object.keys(combined).map(anime => ({
    anime,
    score: combined[anime]
  }));

  finalRecs.sort((a, b) => b.score - a.score);
  return finalRecs.slice(0, 20);
};
```

Таким чином, було поєднано результати колаборативних фільтрацій на основі користувачів та на основі елементів для створення гібридної рекомендаційної системи, додавши фільтрацію на основі вмісту. Це дозволяє враховувати переваги кожного з підходів і покращити точність та релевантність рекомендацій для користувачів системи.

3.4 Опис мобільного додатку

Мобільний додаток було розроблено для iOS з використання мови програмування Swift та фреймворку SwiftUI.

В мобільному додатку було реалізовано наступні сторінки:

- сторінка реєстрації;
- сторінка авторизації;

- головна сторінка з переліком рекомендацій;
- сторінка пошуку аніме;
- сторінка профілю користувача з переліком переглянутих аніме;
- сторінка інформації про аніме;
- сторінка з описом аніме.

Спочатку користувачеві необхідно авторизуватися в додатку (див. рис. 3.4).

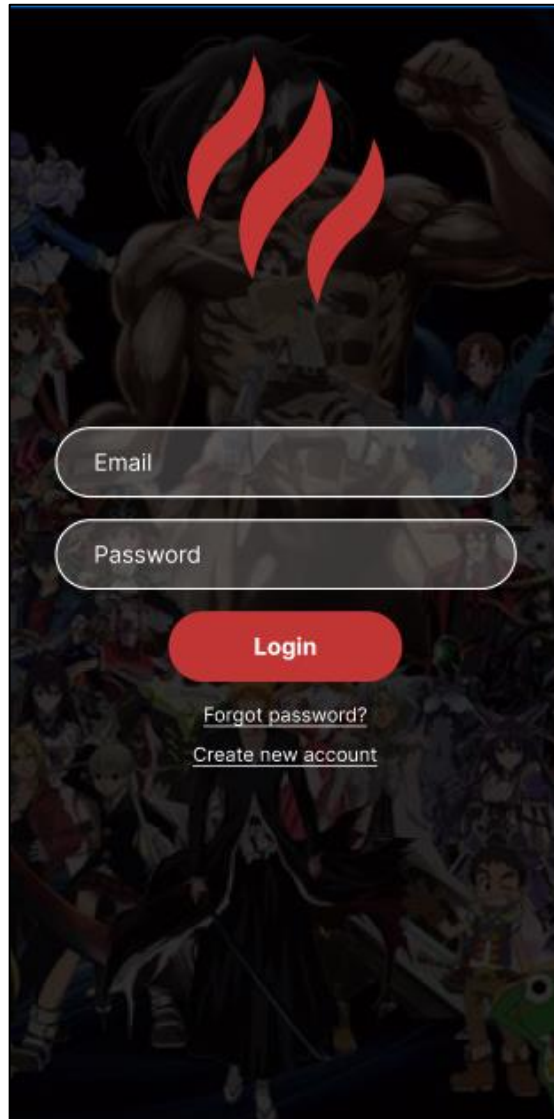


Рисунок 3.4 – Сторінка авторизації (рисунок виконаний самостійно)

Після авторизації користувач попадає на головну сторінку з переліком рекомендацій, топом аніме та можливістю пошуку будь-якого аніме (див. рис. 3.5).

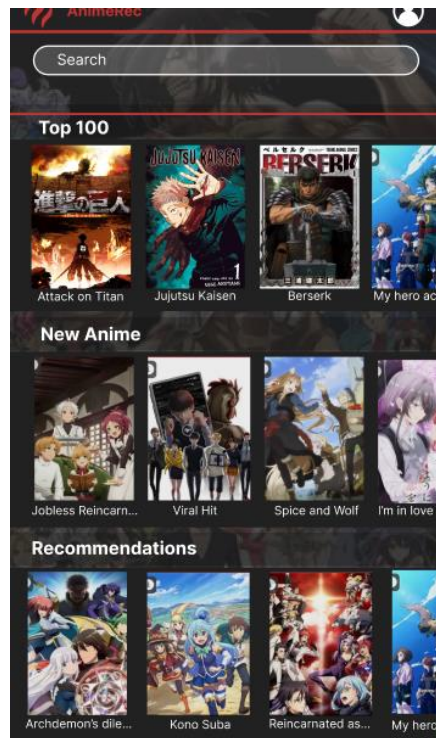


Рисунок 3.5 – Головна сторінка з аніме-рекомендаціями (рисунок виконаний самостійно)

З головної сторінки можна перейти в профіль користувача або обрати будь-яке аніме та подивитися детальний опис.

На рисунку 3.6 наведено профіль користувача.

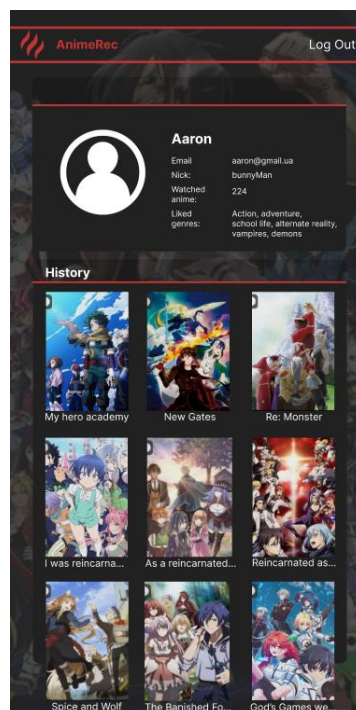


Рисунок 3.6 – Сторінка профілю користувача (рисунок виконаний самостійно)

На сторінці профілю користувач може подивитися історію переглянутих аніме, найбільш популярні жанри, кількість переглянутих аніме.

На рисунку 3.7 наведено сторінку з детальним описом обраного аніме.

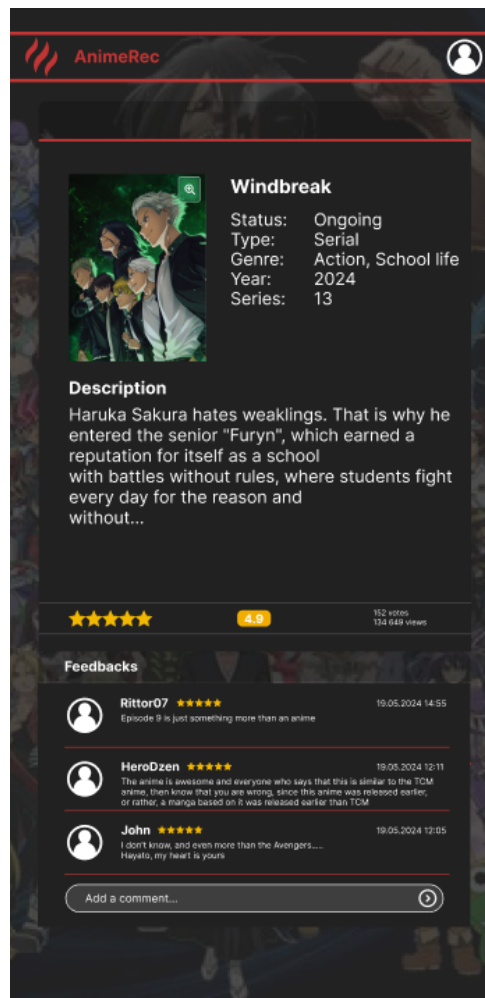


Рисунок 3.7 – Сторінка інформації про аніме (рисунок виконаний самостійно)

На даній сторінці наведено інформацію про аніме: назву, статус, тип, рік випуску, опис сюжету, а також оцінку користувача та середню оцінку серед усіх користувачів.

Нижче опису знаходяться фідбеки від інших користувачів та поле вводу, де можна залишити свій коментар.

4 ОЦІНКА ЕФЕКТИВНОСТІ ГІБРИДНОГО МЕТОДУ

4.1 Вибір підходу для проведення експерименту

Проведення експерименту є дуже важливим етапом для розуміння того, наскільки ефективно працює метод та наскільки точні результати пропонує. Чим більше використовується різних підходів при побудові рекомендаційної системи, тим важливішою стає їх оцінка.

На даний час не існує системи рекомендацій, яка б могла виконувати та задовольняти запити усіх користувачів. Тому, необхідно враховувати, які метрики мають вирішальне значення для цілей системи.

У сфері рекомендаційних систем метрики точності відіграють вирішальну роль в оцінці продуктивності алгоритмів, особливо в сфері аніме, де різноманітність жанрів та напрямків є надзвичайно великою [14].

Двома найбільш часто використовуваними показниками є середня абсолютна похибка (MAE) і середня квадратична помилка (RMSE).

Метрика MAE являє собою середню відстань прогнозованих точок від лінії найкращої відповідності. Результати обчислення даної метрики є інтуїтивно-зрозумілими, оскільки вони показують середню величину помилки у тій самій шкалі, що і оцінки.

Метрика RMSE вимірює середній квадрат помилки між прогнозованими та фактичними оцінками, потім бере квадратний корінь з цього середнього значення. Як і MAE, RMSE оцінює точність прогнозованих оцінок, але робить це з підвищеною чутливістю до великих помилок.

Існують також метрики ранжування необхідні для оцінки якості списку рекомендованих аніме, що надаються користувачам. Однією з таких метрик є показник корисності періоду напіврозпаду (HLU).

Даний показник використовується для оцінки ефективності рекомендаційних систем, враховуючи позицію рекомендованих елементів у списку рекомендацій.

Основна ідея полягає в тому, що користувачі більше звертають увагу на верхні елементи списку, а їх інтерес поступово зменшується зі збільшенням позиції елемента.

Таким чином, чим вище в списку рекомендацій знаходиться релевантний елемент, тим більшу корисність він має для користувача. Для аніме-сфери даний підхід пошуку рекомендацій досить важливим, тому використання даної метрики є доцільним.

Дані метрики дозволяють отримати чітке уявлення про те, наскільки точно алгоритм передбачає вподобання користувачів, і забезпечити високий рівень персоналізації рекомендацій.

4.2 Проведення експерименту та аналіз результатів

Для оцінки ефективності розробленого гібридного методу аніме-рекомендацій було проведено декілька експериментів з використанням метрик MAE, RMSE та HLU.

Для першого експерименту було обрано метрику MAE, яка є однією з найпоширеніших для вимірювання середньої величини помилки між фактичними та прогнозованими значеннями. Обчислення даної метрики виконується за формулою 4.1.

$$MAE = \frac{1}{n} \sum_{i=1}^n |\hat{r}_i - r_i| \quad (4.1)$$

де n – загальна кількість прогнозів,

\hat{r}_i – прогнозована оцінка для i -го елемента,

r_i – фактична оцінка для i -го елемента.

Для порівняння різних підходів було обчислено MAE для трьох розглянутих методів: колаборативна фільтрація на основі користувачів, колаборативна фільтрація на основі елементів, фільтрація на основі вмісту.

Також було обчислено MAE для гібридного методу, який комбінує результати трьох підходів за допомогою методу зваженого середнього.

За результатами обчислення для фільтрації на основі користувачів MAE дорівнює 0.11, для фільтрації на основі елементів – 0.16, для фільтрації на основі вмісту – 0.25, для гібридного методу – 0.08.

На рисунку 4.1 наведено графік абсолютних помилок та MAE для різних прогнозів.

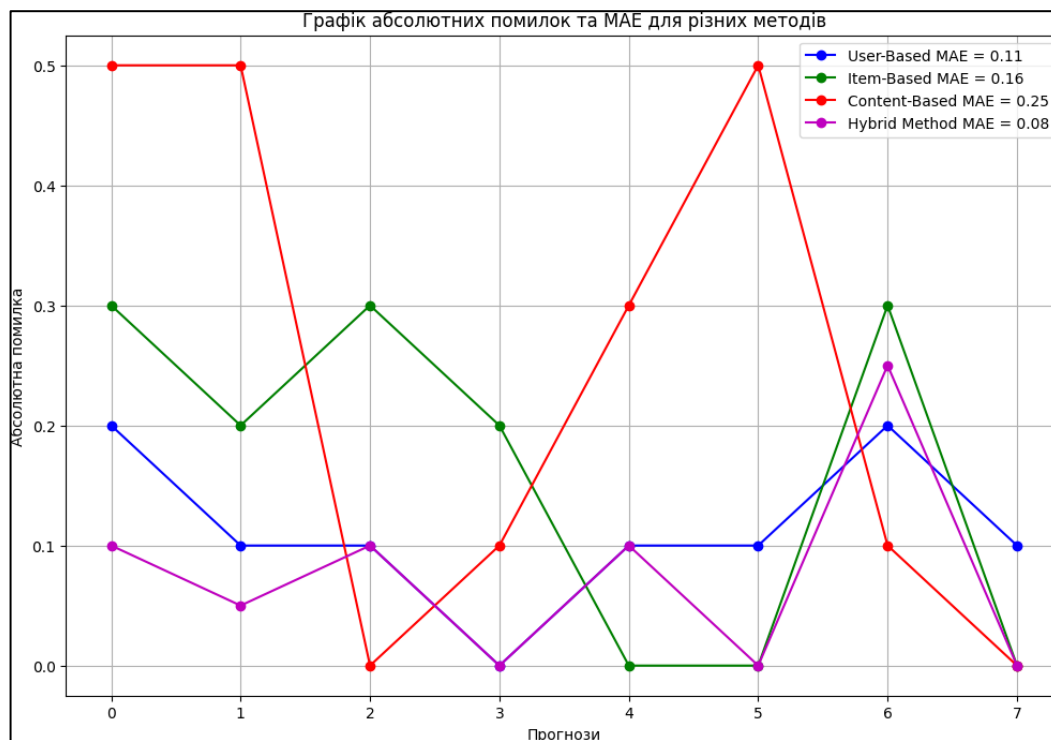


Рисунок 4.1 – Графік середньої абсолютної похибки (рисунок виконаний самостійно)

Результати експерименту показали, що гібридний метод має найнижче значення MAE, що свідчить про його високу точність у порівнянні з окремими методами. Якщо порівнювати три окремих методи, то кращим є фільтрація на основі користувачів, далі – фільтрація на основі елементів, і найгіршою є фільтрація на основі вмісту.

Другий експеримент було проведено з використанням метрики RMSE, яка оцінює різницю між прогнозованими та фактичними оцінками, надаючи більше значення великим похибкам.

Обчислення значення RMSE виконується за формулою 4.2.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{r}_i - r_i)^2} \quad (4.2)$$

де n – загальна кількість прогнозів,

\hat{r}_i – прогнозована оцінка для i -го елемента,

r_i – фактична оцінка для i -го елемента.

Як і в минулому випадку, було обчислено значення RMSE для чотирьох методів.

За результатами обчислення для фільтрації на основі користувачів RMSE дорівнює 0.13, для фільтрації на основі елементів – 0.21, для фільтрації на основі вмісту – 0.33, для гібридного методу – 0.11.

На рисунку 4.2 наведено графік квадратичних помилок та RMSE для різних прогнозів.

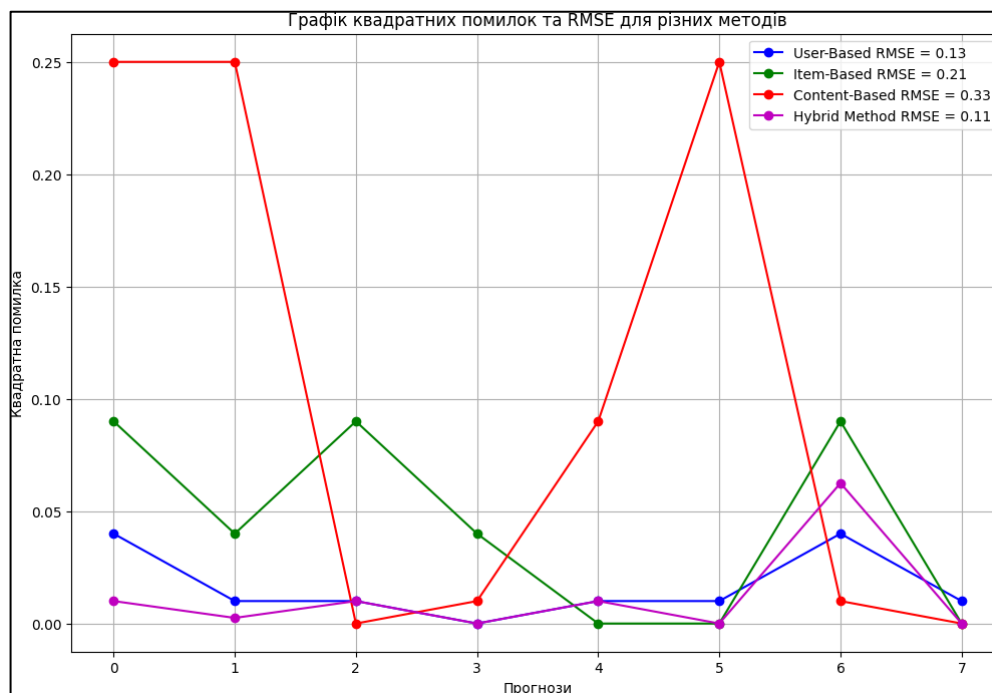


Рисунок 4.2 – Графік середньої квадратичної похибки (рисунок виконаний самостійно)

Результати експерименту показали, що гібридний метод має найнижче значення RMSE, що свідчить про його здатність мінімізувати великі похибки у прогнозуванні.

Третій експеримент було проведено з використанням метрики HLU для оцінки корисності рекомендаційних списків з різними позиціями в рекомендованому списку аніме.

Обчислення значення HLU виконується за формулою 4.3.

$$HLU = \sum_{i=1}^n \frac{rel(i)}{2^{(rank(i)-1)/(\lambda-1)}} \quad (4.3)$$

де n – загальна кількість рекомендованих елементів,

$rel(i)$ – релевантність елемента i ,

$rank(i)$ – позиція елемента i в списку рекомендацій,

λ – параметр напіврозпаду.

Було розраховано HLU для різних діапазонів позицій в списку аніме-рекомендацій 1-5, 6-10, 11-15, 16-20 та 21-25 (див. рис. 4.3).

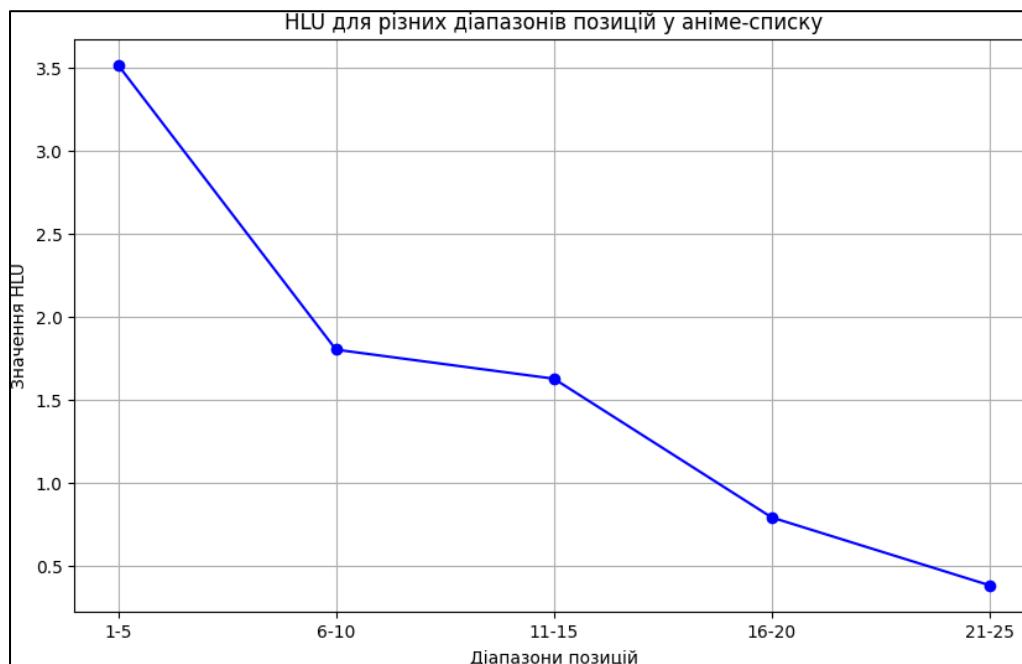


Рисунок 4.3 – Графік показника корисності періоду напіврозпаду (рисунок виконаний самостійно)

Результати показали, що зі зміною діапазону позицій в списку рекомендованих аніме, значення NLU падає. Це свідчить про те, що на перших місцях рекомендації є більш корисними для користувача, як було задумано.

Проведені експерименти підтвердили ефективність гібридного методу рекомендацій для аніме-контенту. Гібридний метод забезпечив найнижчі значення MAE та RMSE, що вказує на його високу точність у прогнозуванні оцінок користувачів.

Таким чином, використання гібридного підходу, який поєднує колаборативну фільтрацію на основі користувачів, колаборативну фільтрацію на основі елементів та фільтрацію на основі вмісту, дозволяє досягти найвищої точності та релевантності рекомендацій. Це підтверджує доцільність його застосування у системах рекомендацій для аніме-контенту.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи було розглянуто методи розробки та реалізації рекомендаційної системи для аніме-контенту на основі гібридного підходу, що поєднує колаборативну фільтрацію на основі користувачів, колаборативну фільтрацію на основі елементів та фільтрацію на основі вмісту.

Було проведено детальний аналіз сучасних методів колаборативної фільтрації та фільтрації на основі вмісту, що дозволило виявити їх сильні та слабкі сторони. Зокрема, фільтрація на основі користувачів враховує вподобання схожих користувачів, що забезпечує високий рівень персоналізації рекомендацій. Фільтрація на основі елементів дозволяє визначати схожість між аніме, що також є важливим для рекомендацій нових аніме на основі вже оцінених. Фільтрація на основі вмісту базується на аналізі характеристик аніме, що дозволяє надавати рекомендації навіть для нових користувачів або нових аніме.

Результати досліджень включають розробку алгоритму на основі гібридного підходу, який поєднує результати трьох методів за допомогою зваженого середнього. Вибір вагових коефіцієнтів для кожного методу обґрунтований важливістю їх внеску в точність рекомендацій: 0.6 – для фільтрації на основі користувачів, 0.3 – для фільтрації на основі елементів, 0.1 – для фільтрації на основі вмісту.

Було розроблено мобільний додаток, в який інтегровано та протестовано гібридний метод.

Проведені експерименти з використанням метрик MAE, RMSE та NLU підтвердили ефективність розробленого гібридного методу. Зокрема, гібридний метод продемонстрував найнижчі значення MAE та RMSE, що вказує на його високу точність у прогнозуванні оцінок користувачів. Використання метрики NLU показало, що рекомендації на верхніх позиціях списку є найбільш корисними для користувачів.

Під час виконання даної роботи було підготовлено та опубліковано тези на 28-ому міжнародному молодіжному форумі «Радіоелектроніка та молодь у XXI столітті» (див. додаток Б) [15].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Anime Market Size & Trends. Grand View Research. URL: <https://www.grandviewresearch.com/industry-analysis/anime-market> (дата звернення: 23.04.2024).
2. The Difference between High-mix Low-volume Wiring Harness Business and High-volume Wiring Harness Business. URL: <https://hilinkteq.com/blog/the-difference-between-high-mix-low-volume-wiring-harness-business-and-high-volume-wiring-harness-business> (дата звернення: 24.04.2024).
3. An Anime Insider's Take on the Netflix Engagement Report. Anime News Network. URL: <https://www.animenewsnetwork.com/feature/2024-01-03/an-anime-insider-take-on-the-netflix-engagement-report/.206122> (дата звернення: 24.04.2024).
4. Saleh I., Chartron G., Kembellec G. Recommender Systems. Wiley & Sons, Incorporated, John, 2021. P. 4-6.
5. Introduction to Collaborative Filtering. Analytics Vidhya. URL: <https://www.analyticsvidhya.com/blog/2022/02/introduction-to-collaborative-filtering> (дата звернення: 05.05.2024).
6. Berkovsky S., Cantador I., Tikk D. Collaborative Recommendations: Algorithms, Practical Challenges and Applications. World Scientific Publishing Co Pte Ltd, 2018. P. 4-5.
7. Comprehensive Guide on Item Based Recommendation Systems. URL: <https://towardsdatascience.com/comprehensive-guide-on-item-based-recommendation-systems-d67e40e2b75d> (дата звернення: 11.05.2024).
8. Chalyi S., Leshchynskyi V. Method of constructing explanations for recommender systems based on the temporal dynamics of user preferences. EUREKA: Physics and Engineering. 2020. Vol. 3. P. 43–50. DOI: <https://doi.org/10.21303/2461-4262.2020.001228> (дата звернення: 25.05.2024).
9. 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. Vol. 4. P. 34–40. DOI: <https://doi.org/10.21303/2461-4262.2019.00952> (дата звернення: 25.05.2024).

10. Chalyi S., Leshchynskyi V Temporal modeling of user preferences in recommender system Chalyi S., Leshchynskyi V. CEUR Workshop Proceedings, 2020, 2711, с. 518-528.

11. Omega C. Z., Hendry. Movie Recommendation System using Weighted Average Approach. 2021 2nd International Conference on Innovative and Creative Information Technology (ICITech), Salatiga, Indonesia, 23–25 September 2021. 2021. DOI: <https://doi.org/10.1109/icitech50181.2021.9590147> (дата звернення: 26.05.2024).

12. Rajasekaran R. K., Iliev A. I. Hybrid Recommendation System. Lecture Notes in Networks and Systems. Cham, 2023. P. 1–11. DOI: https://doi.org/10.1007/978-3-031-28076-4_1 (дата звернення: 27.05.2024).

13. MyAnimeList Dataset. Kaggle: Your Machine Learning and Data Science Community. URL: <https://www.kaggle.com/datasets/svanoo/myanimelist-dataset/data> (дата звернення: 28.05.2024).

14. Shani G., Gunawardana A. Evaluating Recommendation Systems. Recommender Systems Handbook. Boston, MA, 2010. P. 257–297. DOI: https://doi.org/10.1007/978-0-387-85820-3_8 (дата звернення: 27.05.2024).

15. Семенова Н., Смеляков К. Дослідження методів штучного інтелекту для розробки рекомендаційної системи в сфері аніме // 28-й Міжнародний молодіжний форум «Радіоелектроніка та молодь у ХХІ столітті», Харків, ХНУРЕ, 2024. С.804-805.