

Міністерство освіти і науки України
Харківський національний університет імені В. Н. Каразіна
Навчально-науковий інститут комп'ютерних наук та штучного інтелекту
Кафедра комп'ютерних систем та робототехніки

«Затверджую»
в.о. завідуючого кафедри
комп'ютерних систем та робототехніки
_____ к. ф.-м. н., доцент Максим Хруслов
«___» червня 2025 р.

Пояснювальна записка

до кваліфікаційної роботи
бакалавра

на тему: «**МОДЕЛЬ ІНФОРМАЦІЙНОЇ СИСТЕМИ РЕКОМЕНДАЦІЙ
ФІЛЬМІВ ДЛЯ ПОКРАЩЕННЯ КОРИСТУВАЦЬКОГО ДОСВІДУ**».

Спеціальність 123 – Комп'ютерна інженерія.
Галузь знань: 12 – Інформаційні технології.
Освітня програма «Комп'ютерна інженерія».

Захищено на засіданні
Екзаменаційної комісії № 44
протокол № __ від __.06.2025 р.
Оцінка ____ / ____
Голова Екзаменаційної комісії
_____ **ЧУГАЙ А.М.**

Виконав:
студент 4 курсу, групи КІ– 41
СУХИНА Василь Васильович 
Керівник: доцент зво кафедри
комп'ютерних систем та робототехніки,
PhD з інформаційних технологій
МОРОЗ Ольга Юріївна 

Рецензент:
Професор кафедри математичного
моделювання та штучного інтелекту
Національного аерокосмічного
університету «Харківський авіаційний
інститут»,
доктор технічних наук, професор
СКОБ Юрій Олексійович 

Харків – 2025

АНОТАЦІЯ

Пояснювальна записка до кваліфікаційної роботи бакалавра складається зі вступу, трьох розділів, висновків, списку використаних джерел і чотирьох додатків. Загальний обсяг роботи складає 61 сторінку, із яких 43 сторінки основної частини з 19 рисунками, 2 таблицями, 11 найменуваннями списку використаних джерел та чотирма додатками.

У кваліфікаційній роботі розглядається процес розробки моделі інформаційної системи персоналізованих рекомендацій фільмів, орієнтованої на покращення користувацького досвіду. Основна увага приділена застосуванню сучасних методів аналізу даних та алгоритмів машинного навчання для прогнозування вподобань користувачів і формування релевантних рекомендацій кіноконенту.

Метою дослідження є побудова концептуальної та алгоритмічної моделі інформаційної системи рекомендацій фільмів, що забезпечує релевантність персоналізованих рекомендацій і сприяє покращенню користувацького досвіду на основі сучасних підходів до обробки даних та методів машинного навчання.

Об'єкт дослідження – процеси інформаційної підтримки користувачів цифрових платформ розважального контенту засобами рекомендаційних систем персоналізованого типу.

Предмет дослідження – алгоритми та методи побудови систем персоналізованих рекомендацій фільмів на основі аналізу поведінкових даних користувачів.

Ключові слова: інформаційна система, рекомендаційна система, фільтрація, контентна фільтрація, косинусна подібність, датасет, Python, TMDB.

ABSTRACT

An explanatory note to the bachelor's qualification work consists of an introduction, three chapters, conclusions, a list of references, and four appendices. The total volume of the work is 61 pages, of which 43 pages constitute the main part, including 19 figures, 2 tables, 11 references, and two appendices.

The qualification thesis explores the development process of a movie recommendation information system model aimed at enhancing user experience. The focus is placed on the application of modern data analysis methods and machine learning algorithms to predict user preferences and generate relevant film content recommendations.

The goal of the study is to build a conceptual and algorithmic model of a movie recommendation information system that ensures the relevance of personalized recommendations and improves user experience through contemporary data processing approaches and machine learning techniques.

The object of the study is the processes of information support for users of digital entertainment content platforms by means of personalized recommendation systems.

The subject of the study is the algorithms and methods for building personalized movie recommendation systems based on the analysis of user behavioral data.

Keywords: *information system, recommendation system, filtering, content-based filtering, cosine similarity, dataset, Python, TMDB.*

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	5
ВСТУП	6
РОЗДІЛ 1 АНАЛІЗ ТА ОСНОВИ ПОБУДОВИ РЕКОМЕНДАЦІЙНИХ СИСТЕМ	10
1.1 Методи побудови рекомендаційних систем	10
1.2 Аналіз проблем та викликів у створенні рекомендаційних систем.	16
1.3 Побудова персоналізованих систем рекомендацій на онлайн-платформах для фільмів	22
Висновки за розділом 1	26
РОЗДІЛ 2 ПРОЄКТУВАННЯ МОДЕЛІ ІНФОРМАЦІЙНОЇ СИСТЕМИ РЕКОМЕНДАЦІЙ ФІЛЬМІВ	28
2.1 Вибір і підготовка даних для проєктування моделі інформаційної системи рекомендацій фільмів	28
2.2 Фільтрація на основі контенту та алгоритм зваженого рейтингу	29
2.3 Проєктування бази даних	33
Висновки за розділом 2	34
РОЗДІЛ 3 РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ РЕКОМЕНДАЦІЙ ФІЛЬМІВ	36
3.1 Обґрунтування обраних технологій та засобів розробки інформаційної системи рекомендацій фільмів	36
3.2 Опис моделі інформаційної системи рекомендацій фільмів	38
3.3 Забезпечення якості користувацького досвіду	44
Висновки за розділом 3	45
ВИСНОВКИ.....	47
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	50
ДОДАТКИ	52

ПЕРЕЛІК СКОРОЧЕНЬ І УМОВНИХ ПОЗНАЧЕНЬ

ML	–	Machine Learning
AI	–	Artificial Intelligence
IC	–	Інформаційна система
CF	–	Collaborative Filtering
CBF	–	Content-Based Filtering
ГРС	–	Гібридна рекомендаційна система
CSP	–	Cold-start Problem
CoCoS	–	Continuous Cold-start Problem
VoD	–	Video on Demand
OTT	–	Over-The-Top
TMDB	–	The Movie Database
SQL	–	Structured Query Language
API	–	Application Programming Interface
DB	–	Database

ВСТУП

У сучасному світі інформаційних технологій обсяг цифрового контенту зростає експоненційно, що створює проблему вибору для користувачів. Інформаційні системи відіграють ключову роль у забезпеченні ефективної обробки, зберігання та аналізу даних, допомагаючи користувачам отримувати релевантну інформацію. З тисячами фільмів, доступних у різних жанрах, користувачам часто важко знайти вміст, який відповідає їхнім смакам. Отже, системи рекомендацій стали важливим інструментом для покращення взаємодії з користувачем, допомагаючи у відкритті фільмів на основі їхніх інтересів.

Рекомендаційні системи є частиною інформаційних систем, що працюють на основі аналізу великих обсягів даних і застосування штучного інтелекту. Вони відіграють важливу роль у покращенні користувацького досвіду, оскільки дозволяють мінімізувати час на пошук потрібного контенту та підвищити задоволеність від перегляду. Такі системи базуються на різних алгоритмах, зокрема колаборативній фільтрації, контентному підході та гібридних моделях, що поєднують кілька методів для досягнення максимальної точності рекомендацій.

Більшість сучасних рекомендаційних систем фільмів орієнтується на соціальні аспекти. Алгоритми аналізують популярність фільмів і серіалів, враховуючи жанр та акторський склад. Хоча такий підхід часто дає корисні результати, він не завжди дозволяє сформувати список дійсно схожих фільмів.

Актуальність роботи. У сучасних умовах стрімкого зростання обсягів аудіовізуального контенту на платформах потокового передавання виникає проблема ефективної навігації користувачів у великому масиві доступних фільмів. Однією з ключових складностей є пошук контенту, який відповідає індивідуальним вподобанням користувача. Традиційні методи сортування, зокрема фільтрація за жанром, рейтингом або алфавітним порядком, не

забезпечують достатнього рівня релевантності та персоналізації результатів, що негативно впливає на якість користувацького досвіду.

У зв'язку зі зростанням попиту на персоналізовані сервіси в сфері цифрових розваг, особливо актуальним є впровадження систем рекомендацій, заснованих на аналізі користувацьких даних із використанням методів машинного навчання. Такі системи дозволяють адаптувати пропозиції контенту до індивідуальних уподобань, підвищуючи задоволеність користувачів та ефективність взаємодії з платформою.

Незважаючи на значний прогрес у застосуванні алгоритмів машинного навчання для передбачення вподобань, все ще існує низка викликів, пов'язаних з обмеженістю початкових даних, холодним стартом для нових користувачів та потребою у збалансуванні точності й швидкості рекомендацій. У цьому контексті особливої ваги набуває дослідження, спрямоване на побудову алгоритмічних моделей, здатних формувати релевантні рекомендації навіть за умов обмеженої інформації про користувача.

Таким чином, створення ефективної моделі рекомендаційної системи, що використовує сучасні методи аналізу даних і машинного навчання, є актуальним завданням як з наукової, так і з практичної точки зору. Результати такого дослідження можуть бути застосовані у розробці адаптивних інформаційних сервісів у сфері цифрових медіа та розваг, сприяючи підвищенню якості користувацької взаємодії та персоналізації цифрового середовища.

Основна задача, що вирішується в кваліфікаційній роботі полягає в тому, щоб підвищити точність та ефективність рекомендацій фільмів шляхом розробки та впровадження моделі інформаційної системи, що використовує сучасні методи аналізу даних та машинного навчання.

Метою дослідження є побудова концептуальної та алгоритмічної моделі інформаційної системи рекомендацій фільмів, що забезпечує релевантність персоналізованих рекомендацій і сприяє покращенню

користувацького досвіду на основі сучасних підходів до обробки даних та методів машинного навчання.

Об’єкт дослідження – процеси інформаційної підтримки користувачів цифрових платформ розважального контенту засобами рекомендаційних систем персоналізованого типу.

Предмет дослідження – алгоритми та методи побудови систем персоналізованих рекомендацій фільмів на основі аналізу поведінкових даних користувачів.

Задля досягнення мети роботи прийнято рішення розробити концептуальну та алгоритмічну модель системи персоналізованих рекомендацій фільмів, яка включає в себе модулі збору даних, обробки інформації про користувачів, побудови моделей машинного навчання для генерації рекомендацій, а також інтерфейси взаємодії з користувачем.

Задачі дослідження:

1. Провести аналіз сучасних інформаційних систем рекомендацій фільмів та сучасні підходи до покращення користувацького досвіду.
2. Дослідити рекомендаційні алгоритми та їх застосування в подібних системах.
3. Розробити модель інформаційної системи рекомендацій фільмів із урахуванням уподобань користувача.
4. Реалізувати програмний прототип системи з базовим функціоналом:
 - реєстрація/авторизація користувачів;
 - введення/збір вподобань;
 - генерація персоналізованих рекомендацій;
 - виведення списку фільмів з коротким описом і рейтингом.
5. Провести проектування інтерфейсу користувача для зручної взаємодії із системою.
6. Розробити програмну реалізацію інформаційної системи рекомендацій фільмів для покращення користувацького досвіду.

7. Провести тестування інформаційної системи на основі вибраного набору фільмів.
8. Розробити пропозиції щодо покращення та масштабування системи

Розроблена система призначена для покращення користувацького досвіду під час вибору фільмів. Вона може бути інтегрована в онлайн-кінотеатри, стрімінгові платформи, медіасервіси та інші цифрові платформи, що надають контент. Система аналізує вподобання користувачів і надає персоналізовані рекомендації, підвищуючи зручність користування, залученість аудиторії та якість перегляду.

Особливу увагу приділено аналізу ефективності моделі: для оцінювання якості рекомендацій використано відповідні метрики точності та повноти. Результати дослідження демонструють потенціал використання інтегрованих методів штучного інтелекту для вдосконалення взаємодії користувача з цифровими медіасервісами. Розроблену систему можна адаптувати до потреб онлайн-кінотеатрів, стрімінгових платформ та інших сервісів, що надають мультимедійний контент.

РОЗДІЛ 1

АНАЛІЗ ТА ОСНОВИ ПОБУДОВИ РЕКОМЕНДАЦІЙНИХ СИСТЕМ

1.1 Методи побудови рекомендаційних систем

Основні способи формування прогнозів у рекомендаційних системах поділяються на чотири категорії: методи колаборативної фільтрації, підходи, що базуються на контентному аналізі, моделі, засновані на знаннях, а також комбіновані (гібридні) технології

Колаборативна фільтрація (рис. 1.1) є одним з основних методів, на якому ґрунтується робота деяких рекомендаційних сервісів. Вона має два значення: більш конкретне та загальне. В загальному розумінні, колаборативна фільтрація означає процес відбору інформації за допомогою взаємодії між кількома агентами, точками зору чи джерелами даних тощо. Цей метод найчастіше застосовується до великих обсягів даних. Колаборативні методи фільтрації використовуються в різних сферах, таких як моніторинг даних, пов'язаних з розвідкою корисних копалин на великих територіях, фінансові дані, що обробляються установами фінансових послуг, або в електронній торгівлі та веб-додатках, орієнтованих на дані користувачів. У даному контексті основна увага приділяється колаборативній фільтрації, яка орієнтована на дані користувачів, хоча деякі методи можуть бути застосовані й у інших сферах [1].



Рисунок 1.1 – Схема функціонування методу кореляційної фільтрації.

У сучасному трактуванні під колаборативною фільтрацією розуміють метод прогнозування в рамках рекомендаційних систем, що базується на аналізі вже наявних вподобань (оцінок), наданих спільнотою користувачів, з метою передбачення інтересів інших осіб.

Основне припущення колаборативної фільтрації полягає в тому, що користувачі, які давали схожі оцінки предметам у минулому, ймовірно, надаватимуть подібні оцінки іншим предметам у майбутньому. Наприклад, музичний додаток, застосовуючи колаборативну фільтрацію, може прогнозувати, яка музика може сподобатися користувачеві, навіть якщо його список уподобань (симпатій та антипатій) неповний. Прогнози створюються індивідуально для кожного користувача, хоча для цього використовуються дані, зібрані від багатьох інших учасників. Це відрізняє колаборативну фільтрацію від більш простих підходів, які дають середні оцінки для кожного об'єкта на основі загальної кількості голосів. Дослідження у цій галузі тривають і сьогодні, оскільки метод колаборативної фільтрації має ще низку нерозв'язаних проблем [2].

Контентно-орієнтовані підходи (фільтрація на основі контенту) становлять основу функціонування багатьох сучасних рекомендаційних систем. На відміну від колаборативного підходу, цей метод не потребує попередньої інформації про інтереси конкретного користувача. Замість цього, пропозиції формуються, спираючись на властивості самих об'єктів – наприклад, жанр, виробника або інші характерні риси товару чи послуги. До процесу можуть залучатися будь-які наявні дані, які можливо зібрати для подальшого аналізу.

Цей підхід часто застосовується платформами для того, щоб не втратити нових користувачів, дані про яких ще не зібрані. Однак є два основних недоліки: спочатку система може давати неточні прогнози, і її реалізація вимагає більше часу.

Фільтрація, заснована на знаннях, ґрунтується на знаннях про певну предметну область, такі як інформація про користувачів, товари та інші елементи, що допомагають у процесі ранжування. Як і в випадку з контентною фільтрацією, оцінки інших користувачів системи не беруться до уваги. Цей підхід має кілька різновидів, серед яких можна виділити: методи, що базуються на прикладах (case-based), демографічні підходи (demographic-based), моделі, орієнтовані на корисність (utility-based), критиковані рекомендації (critique-based), а також інші специфічні варіанти, такі як «whatever-you-want-based».

При розробці нового проєкту, залежно від сфери діяльності, до рекомендаційної системи можна інтегрувати будь-яку предметну область для подальшого ранжування за її критеріями.

Основний недолік фільтрації, заснованої на знаннях, полягає в тому, що для її впровадження потрібно значно більше часу та ресурсів. Однак отримані результати, як правило, виправдовують ці витрати. Комбінація кількох алгоритмів у рамках гібридної системи на одній платформі дозволяє зменшити недоліки кожного окремого методу. Великі сервіси та інтернет-магазини часто застосовують гібридні підходи для досягнення кращих результатів.

Одним із ефективних підходів до побудови таких систем є гібридні рекомендуючі системи (ГРС), які поєднують кілька методів рекомендацій з метою усунення обмежень окремих підходів і підвищення загальної ефективності системи.

Класифікація гібридних стратегій

Згідно з узагальненими підходами, що наведені у працях Cano & Morisio [1], а також у систематичному огляді Ricci et al [2], розрізняють такі основні типи гібридизації:

- Зважене поєднання – результати декількох моделей поєднуються на основі заданих або адаптивних ваг, що забезпечує баланс між точністю та узагальненістю.

- Каскадна гібридизація – вихід одного рекомендуючого методу передається як вхід до іншого, що дозволяє уточнювати попередньо сформовані рекомендації.
- Перемикання – система обирає найдоцільніший метод в залежності від поточного контексту (наприклад, наявності даних про користувача або об'єкт).
- Інтеграція ознак – об'єднання атрибутів, отриманих з різних джерел або методів, з подальшим використанням у загальній моделі машинного навчання.
- Моделювання ансамблю – побудова моделі вищого рівня (метамоделі), яка використовує вихідні дані інших рекомендуючих систем як вхідні ознаки.

Існує кілька найбільш поширених методів комбінування алгоритмів:

- окреме застосування колаборативних та контентних алгоритмів з подальшим поєднанням їх припущень;
- інтеграція контентних правил у колаборативний підхід;
- впровадження колаборативних правил у контентний підхід;
- створення єдиної моделі, що поєднує правила обох методів.

Проблема холодного старту – це потенційна складність, що виникає в комп'ютерних інформаційних системах, яка пов'язана з обмеженою здатністю автоматично моделювати дані. Зокрема, це стосується ситуації, коли система не може зробити прогнози або рекомендації для користувачів або елементів, для яких вона ще не зібрала достатньо інформації [3].

Під час розробки рекомендаційної системи важливо обрати найбільш доцільний метод, що відповідатиме завданням платформи. Це рішення має базуватись на доступності даних, особливостях їх джерел, напрямку роботи системи, а також враховувати ймовірність виникнення ефекту холодного старту або його повторного прояву при оновленні чи розширенні інформаційної бази.

Головною метою рекомендаційної системи є надання користувачеві інформації про той товар чи послугу, які мають найвищу ймовірність зацікавити його саме в даний момент часу. Клієнт отримує корисну інформацію, а сервіс заробляє, надаючи високоякісні послуги. Послуги можуть включати не лише прямий продаж товару, але й заробіток на комісіях або збільшення лояльності користувачів, що в подальшому може призвести до зростання доходів від реклами та інших джерел.

Таблиця 1.1

Стандартна модель рекомендаційної системи

Параметр	Опис
Об'єкт рекомендації	Те, що пропонується користувачу (фільми, товари, музика, контакти тощо).
Мета рекомендації	Причина або мета, з якою робиться рекомендація (покупка, ознайомлення, навчання, соціальна взаємодія тощо).
Ситуація використання	Дія, яку виконує користувач у момент отримання рекомендації (огляд товарів, прослуховування музики, спілкування тощо).
Джерело рекомендацій	Хто або що формує рекомендації (середній рейтинг, схожі користувачі, експертна думка).
Рівень персоналізації	Неперсоналізовані: однакові для всіх, можуть враховувати загальні чинники (регіон, час). Сесійні: базуються на поточній поведінці користувача. Персоналізовані: враховують історію взаємодії і додаткові дані користувача.
Відкритість системи	Впливає на довіру користувачів та знижує ризики маніпуляцій, наприклад, підкуп рейтингів чи накрутки.
Візуалізація	Форма подання рекомендацій: спливаючі вікна, списки, стрічки в нижній частині екрану тощо.
Використані алгоритми	Summary-based – неперсоналізовані рекомендації. Content-based – моделі, які спираються на характеристики контенту. Collaborative Filtering – колаборативна фільтрація. Matrix Factorization – методи, що базуються на розкладанні матриць.

В залежності від бізнес-моделі, рекомендації можуть бути ключовим елементом бізнесу, як це відбувається в TripAdvisor, або додатковою зручною

функцією, наприклад, в інтернет-магазинах одягу, покликаною покращити досвід користувача та полегшити навігацію по каталогу.

Персоналізація онлайн-маркетингу стала одним із основних трендів останнього десятиліття. За оцінками McKinsey, 35% доходу Amazon і 75% доходу Netflix генерується завдяки рекомендованим товарам, і цей показник, ймовірно, буде зростати. Рекомендаційні системи зосереджені на тому, щоб пропонувати клієнтам такі продукти чи послуги, які роблять їх задоволеними.

Для розробки рекомендаційних систем використовують ті ж архітектурні підходи, що й для інших сервісів: монолітну, мікросервісну та сервісно-орієнтовану архітектури. Монолітна архітектура характеризується єдиною структурою додатку, що спрощує процес розробки і розгортання, особливо для невеликих проєктів. Проте у випадку систем продажів із вбудованою рекомендаційною функціональністю монолітний підхід може призвести до зниження модульності та проблем із масштабуванням, що ускладнює подальше розширення і підтримку. Крім того, така архітектура ускладнює оперативну заміну або масштабну модернізацію компоненту рекомендацій.

Застосування сервісно-орієнтованої архітектури у програмних рішеннях для рекомендаційних систем забезпечує низку ключових переваг (рис. 1.2):

- Модульність – компоненти програми, зокрема рекомендаційна система, можуть бути замінені або масштабно модернізовані без впливу на інші частини системи;
- Масштабованість – окремі частини сервісу можна розширювати новим функціоналом без порушення роботи інших компонентів;
- Незалежність технологічного стеку – кожен компонент може використовувати свій власний стек технологій;
- Надійне з'єднання реалізується через протокол HTTP [4].

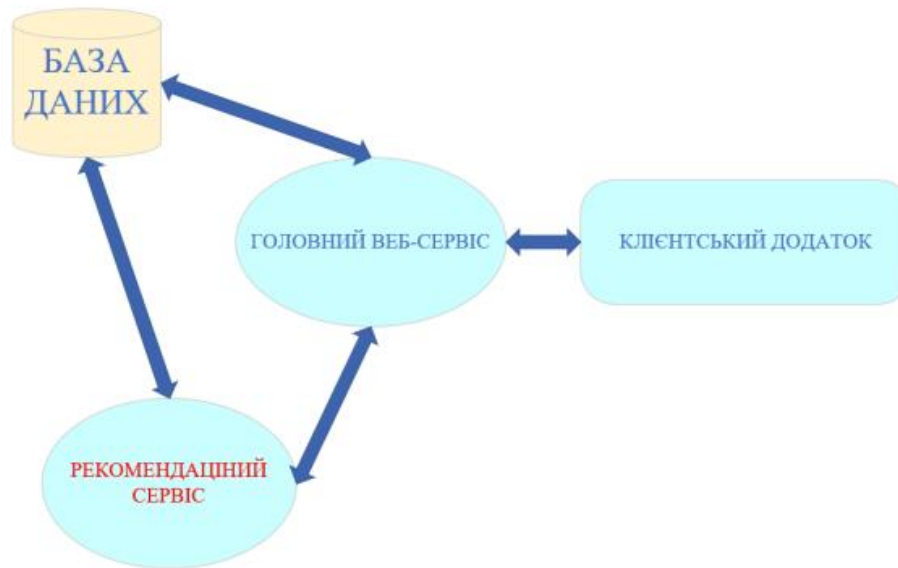


Рисунок 1.2 – Схематичне зображення застосування сервіс-орієнтованої архітектури при розробці додатку з вбудованою рекомендаційною системою.

1.2 Аналіз проблем та викликів у створенні рекомендаційних систем

Однією з ключових складностей у роботі рекомендаційних систем є проблема холодного старту (Cold-start Problem, CSP). Вона виникає, коли до системи додаються нові елементи – нові користувачі (User Cold-Start), які ще не мають історії вподобань, або нові об'єкти (Item Cold-Start), які не отримали жодних оцінок чи характеристик.

У практичних системах проблема холодного старту може набувати циклічного характеру, що впливає на вже існуючих користувачів та об'єкти. Наприклад, коли користувачі змінюють свої інтереси, ця ситуація отримала назву проблеми постійного холодного старту (Continuous Cold-start Problem, CoCoS). Така проблема може стосуватися як користувачів (User Continuous Cold-start Problem), так і об'єктів (Item Continuous Cold-start Problem) [5].

User Continuous Cold-start Problem з'являється у випадках, коли користувачі змінюють свої вподобання або нечасто взаємодіють із системою, залишаючи мало даних для оцінювання нових об'єктів. Item Continuous Cold-start Problem відноситься до об'єктів, характеристики яких змінюються з плином часу.

Для подолання проблеми холодного старту зазвичай застосовують такі підходи:

- гібридні рекомендаційні системи, які комбінують контентну та колаборативну фільтрацію;
- використання контекстної інформації, наприклад демографічних даних, часу або дати, під час генерації рекомендацій.

Такі методи не дають належного результату у випадках постійного холодного старту, адже вони базуються на гіпотезі про незмінність властивостей користувача чи об'єкта після того, як їх уже було ідентифіковано. Для успішного вирішення цієї задачі потрібно не лише прогнозувати вподобання користувачів, але й відстежувати та передбачати зміни їхніх інтересів, а також враховувати можливі трансформації властивостей рекомендованих об'єктів. Сучасні підходи для подолання цієї проблеми базуються на методах машинного навчання, які підвищують адаптивність систем до постійних змін.

Ще однією проблемою у роботі рекомендаційних систем є явище фільтрових бульбашок. Класичні системи рекомендацій формують пропозиції, спираючись виключно на минулі вподобання користувача. В результаті користувач потрапляє в інформаційне середовище, яке містить лише вузький набір схожих об'єктів. Наслідки, що виникають через так звану фільтрову бульбашку, включають:

- Користувач не отримує альтернативну інформацію, яка могла б бути для нього корисною (наприклад, об'єкти, про які він не знає, але які можуть ефективніше задовольнити його потреби).
- Виникає спотворене сприйняття інформаційного середовища, оскільки користувач не бачить повної картини (наприклад, у випадку з рекомендаціями новин).

- Користувач може втратити інтерес до рекомендацій, оскільки йому постійно пропонують однакові або схожі об'єкти (наприклад, втрати інтересу до онлайн-радіо, яке транслює одні й ті ж пісні).

Оскільки основною метрикою якості роботи більшості рекомендаційних систем є точність прогнозування вподобань користувачів, формулювання задачі прогнозування оцінок виглядає наступним чином:

$$d(R, V) \rightarrow \min, \quad (1.1)$$

де $R = (r_1, r_2, \dots, r_n)$ – вектор прогнозованих оцінок користувача, впорядкований за спаданням значень, $V = (v_1, v_2, \dots, v_n)$ – вектор реальних оцінок, які користувач поставив, але які системі невідомі під час формування рекомендацій. У такій ситуації проблема фільтрових бульбашок залишається актуальною для будь-яких рекомендаційних систем, адже якісна система повинна генерувати рекомендації, які максимально відображають справжні вподобання користувача.

Найчастіше використовуються різні метрики відстані, серед яких:

- Евклідова відстань (формула 1.2) – класична метрика, яка обчислює корінь із суми квадратів різниць між відповідними ознаками;

$$d(x_1, x_2) = \sqrt{\sum_{i=1}^m (x_{1i} - x_{2i})^2}, \quad (1.2)$$

- Зважена евклідова відстань (формула 1.3) – враховує важливість кожної змінної через вагові коефіцієнти;

$$d(x_1, x_2) = \sqrt{\sum_{i=1}^m w_i (x_{1i} - x_{2i})^2}, \quad (1.3)$$

- Манхеттенська (міська) відстань (формула 1.4) – базується на абсолютних різницях між значеннями ознак;

$$d(x_1, x_2) = \sum_{i=1}^m |x_{1i} - x_{2i}|, \quad (1.4)$$

- Мінкова відстань (формула 1.5) – узагальнення попередніх метрик з регульованим параметром p ;

$$d(x_1, x_2) = (\sum_{i=1}^m |x_{1i} - x_{2i}|^p)^{1/p}, \quad (1.5)$$

- Махаланобісова відстань (формула 1.6) – враховує кореляцію між ознаками та масштабує простір ознак з використанням коваріаційної матриці.

$$d(x_1, x_2) = \sqrt{(\overline{X}_1 - \overline{X}_2)^T \Sigma^{-1} (\overline{X}_1 - \overline{X}_2)} \quad (1.6)$$

де $d(x_1, x_2)$ – відстань між об'єктами x_1 та x_2 ; x_{1i} , x_{2i} – значення i -ї ознаки відповідно для першого та другого об'єкта; w_i – вага, що призначається цій ознаці; Σ^{-1} – обернена матриця до коваріаційної матриці, обчислена на основі всієї вибірки; $\overline{X}_1$, $\overline{X}_2$ – вектори значень ознак для першого та другого об'єктів відповідно.

Коефіцієнти подібності між парами об'єктів можуть бути подані у вигляді матриці подібності, що дає змогу здійснити оцінку різноманітності сформованого списку рекомендацій. Такий підхід особливо ефективний, коли ознаки мають числову природу, адже дозволяє безпосередньо застосовувати метрики відстані, розглянуті раніше.

Проте на практиці часто трапляються категоріальні, номінальні або рангові ознаки, які не підлягають безпосередньому числовому аналізу. У таких випадках застосовується перетворення ознак у двійкову форму, де кожна категорія або клас кодується окремим бітовим індикатором. У результаті

кожен об'єкт представлений вектором ознак, що описує його характеристики у стандартизованому вигляді:

$$\bar{X}_i = (x_{i1}, x_{i2}, \dots, x_{im}), \text{ де } i = 1, n \quad (1.7)$$

Для вирішення проблеми фільтрових бульбашок зазвичай застосовують додаткові вимоги до формування списку рекомендацій, наприклад, забезпечення таких властивостей:

Різноманітність (Diversity) – це показник, який характеризує ступінь відмінності елементів у списку рекомендацій. Елементи не повинні бути надто схожими, а навпаки – мати різні властивості, наприклад, у випадку фільмів це можуть бути стрічки різних жанрів, а не лише фільми одного жанру чи трилогії. Для вимірювання схожості між елементами використовують різні коефіцієнти подібності, що дозволяють порівнювати елементи попарно. На основі цих показників можна визначити рівень різноманітності всього списку рекомендацій. Якщо об'єкт, описаний m ознаками, відображається як точка в k -вимірному просторі, тоді подібність між об'єктами визначається відстанню між відповідними точками. Найпоширенішими метриками, які застосовують у такому контексті, є:

Неочікуваність (Serendipity) – це характеристика елементів, які є сюрпризом для користувача та не схожі на його попередні вподобання. Хоча точне визначення неочікуваності варіюється, більшість дослідників вважають, що елемент, що має цю властивість, повинен бути важливим, новим і несподіваним для користувача. Важливість для користувача виражається в його реакції на рекомендований елемент, а новизна – в тому, наскільки цей елемент є незнайомим для нього.

Елемент може вважатися незнайомим для користувача у таких випадках:

- 1) Користувач ніколи не чув про існування цього елемента;
- 2) Користувач знає про елемент, але ніколи ним не користувався;

- 3) Користувач раніше використовував цей елемент, але вже його забув.

Для користувача цей елемент є непрогнозованим у випадку, коли:

- 1) Користувач не припускає, що цей елемент може бути для нього актуальним;
- 2) Користувач не припускає, що цей елемент йому порекомендують;
- 3) Користувач не зміг би самостійно його знайти;
- 4) Цей елемент суттєво відрізняється від того, що користувач зазвичай обирає;
- 5) Користувач не очікує побачити цей елемент у рекомендаціях, оскільки в останній час він переглядав інші категорії об'єктів.

Новизна (Novelty) в контексті рекомендаційних систем – це характеристика елементів, яка визначає, наскільки цей елемент є новим і незнайомим для користувача. Ключова ідея полягає в тому, що якщо рекомендації складаються лише з популярних об'єктів, користувач, ймовірно, вже знайомий з ними і може вибрати їх без допомоги системи, що робить такі рекомендації малоефективними. Формула, що використовується для розрахунку новизни, має наступний вигляд:

$$novely(i) = -\log_2 p(i) \quad (1.8)$$

де $p(i)$ – ймовірність того, що елемент i увійде до списку рекомендацій (буде обраний).

Забезпечення виконання додаткових вимог при формуванні списку рекомендацій, таких як новизна, різноманітність та неочікуваність, дійсно може зменшити точність прогнозування в рекомендаційних системах, оскільки ці елементи не завжди відповідають безпосереднім вподобанням користувача. Однак такі вимоги є важливими для подолання проблеми бульбашки фільтрів, оскільки вони дозволяють користувачам отримувати рекомендації, які можуть бути новими та відрізнятися від того, що вони вже

бачили, тим самим знижуючи ефект «інформаційного заціпеніння», де система рекомендує тільки схожі елементи.

Таблиця 1.2

**Результати порівняльного аналізу відомих методів побудови
рекомендаційних систем [6]**

Метод рекомендацій	Складність реалізації	Проблема User Cold-Start	Проблема Item Cold-Start	Проблема User Continuous Cold-Start	Проблема Item Continuous Cold-Start	Проблема фільтрових бульбашок
Колаборативна фільтрація	–	+	+	+	+	+
Контентна фільтрація	–/+	+	–	+	+	+
Гібридні методи	+	+	–	+	+	+
Гібридні методи з контекстом	+	– /+	–	+	+	+
Гібридні методи з урахуванням Diversity, Serendipity, Novelty	++	+	–	+	+	– /+
Адаптивні гібридні методи з використанням штучного інтелекту	++	+	–	+/-	+/-	+

1.3 Побудова персоналізованих систем рекомендацій на онлайн-платформах для фільмів

Сучасні системи рекомендацій стали невід’ємною складовою провідних відеоплатформ, таких як Netflix, Megogo, YouTube, HBO,

Amazon Prime, Disney+, Hulu, Okko і Spotify. Ці сервіси широко використовують різноманітні методи персоналізації контенту з метою підвищення рівня залученості своїх користувачів.

Наприклад, онлайн-відеосервіс Netflix використовує технології штучного інтелекту та машинного навчання для демонстрації персоналізованих трейлерів фільмів і серіалів, враховуючи індивідуальні вподобання глядачів. Архітектура рекомендаційної системи Netflix базується на кількох ключових компонентах: зборі даних, їхній обробці, використанні моделей машинного навчання та генерації рекомендацій. Ці компоненти тісно взаємодіють між собою, оскільки ефективна архітектура неможлива без збору найважливішої інформації про користувачів і їх взаємодію з платформою.

На початковому етапі здійснюється збір користувацьких даних. Платформа Netflix акумулює різноманітну інформацію про поведінку користувачів, зокрема історію переглядів, виставлені оцінки фільмам і серіалам, пошукові запити, а також дані про взаємодію з інтерфейсом системи (наприклад, тривалість перебування на окремих екранах, перегляд трейлерів, відкриття описів тощо) (рис. 1.3).



Рисунок 1.3 – Метрики, за якими формуються рекомендації у Netflix

Важливою складовою архітектури системи є механізм персоналізації. Моделі постійно підлаштовуються під індивідуальні уподобання кожного користувача, враховуючи його унікальну історію взаємодії з платформою. Персоналізація відбувається в режимі реального часу, що дає змогу Netflix оперативно реагувати на зміни у поведінці користувача і пропонувати найбільш релевантний контент.

У 2006 році компанія Netflix вперше впровадила метод матричної факторизації для побудови персоналізованих рекомендацій у межах конкурсу з машинного навчання Netflix Prize. З того часу підходи до надання рекомендацій у Netflix зазнали значної еволюції: від простого прогнозування оцінок користувачів до динамічного формування сторінок з контентом і створення індивідуалізованого користувацького досвіду. Візуалізацію цієї еволюції наведено на рисунку 1.4.

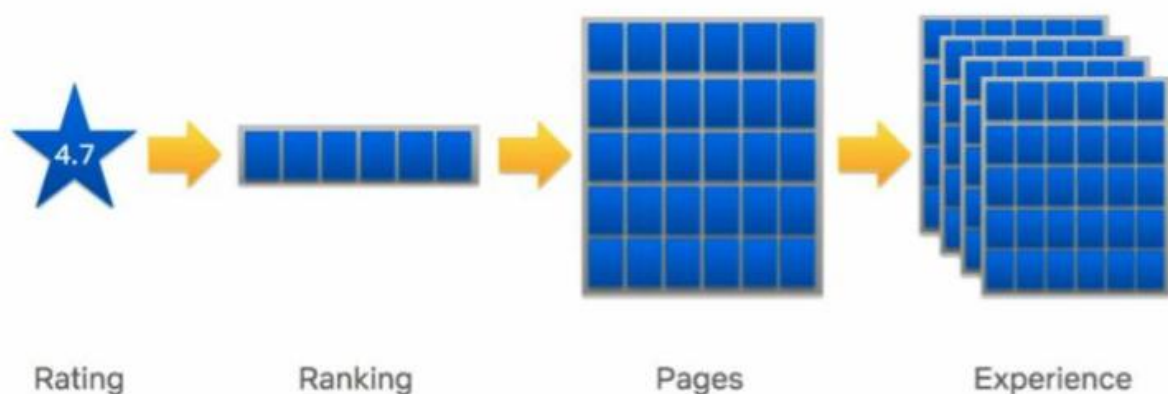


Рисунок 1.4 – Розвиток методів формування рекомендацій у системі Netflix.

Megogo є прикладом гібридної цифрової платформи, що об'єднує можливості OTT (over-the-top) та VoD (відео на вимогу) сервісів. Цей сервіс дає користувачам змогу дивитися телепередачі в режимі реального часу, а також отримувати доступ до відео- та аудіоконтенту за запитом. Наразі аудиторія Megogo перевищує 50 мільйонів унікальних користувачів із 15 різних країн.

Історично компанія розпочала діяльність у складний для галузі період, коли спостерігався високий рівень піратства, що зумовило необхідність використання безплатної моделі з монетизацією через рекламу. У подальшому Megogo трансформувався в підписний сервіс, досягнув фінансової стабільності та увійшов до числа провідних медіаплатформ Східної Європи.

У контексті персоналізованих рекомендацій Megogo використовує алгоритмічні підходи до аналізу поведінки користувачів, їхніх інтересів і переваг, що дозволяє формувати індивідуальні добірки контенту, адаптовані до уподобань конкретного глядача.

На платформі реалізовано інтелектуальну систему рекомендацій, яка ґрунтується на методі матричної факторизації та функціоналі запам'ятовування місця зупинки перегляду (рис. 1.5). Основним джерелом даних для побудови рекомендацій слугує внутрішня система трекінгу активності користувачів – WatchStat, яка фіксує фактичні перегляди контенту. Формування персоналізованих рекомендацій здійснюється для всіх користувачів, що переглянули щонайменше два відео.

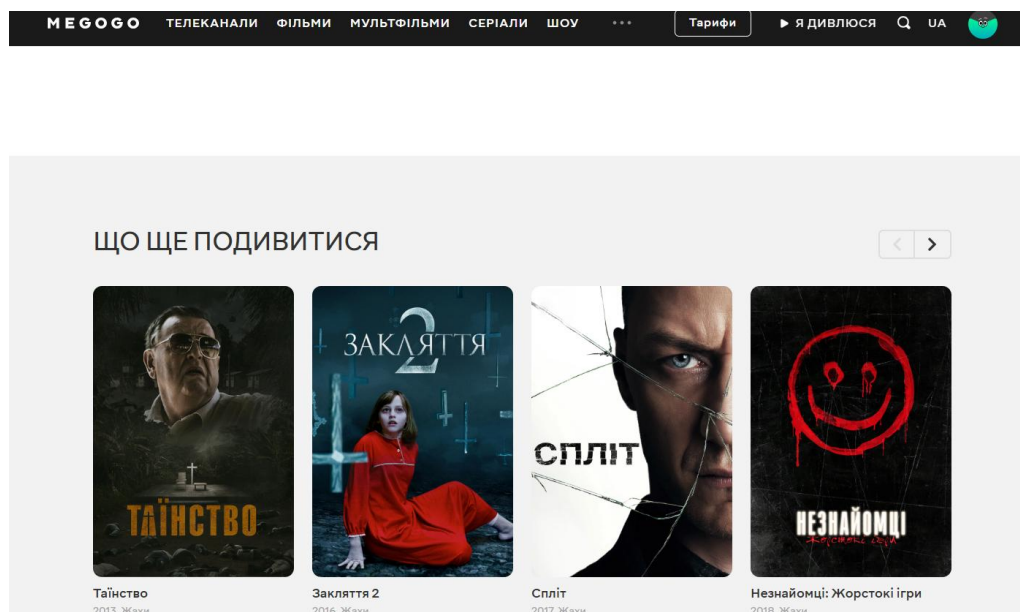


Рисунок 1.5 – Формування рекомендацій Megogo на основі раніше переглянутого контенту.

Як зазначають самі розробники системи рекомендацій, нещодавно їм вдалося реалізувати нову модель, яка сприяла значному зростанню клікабельності добірки «рекомендоване» – вона підвищилась з 14% до 28%. Це свідчить про те, що користувачі оцінюють цю добірку як значно більш корисну та релевантну в порівнянні з попередніми версіями системи рекомендацій. Однак реальну ефективність цього алгоритму можна точно оцінити лише в тому випадку, якщо визначити, чи дійсно користувач переглянув рекомендований контент після того, як він ознайомився з ним через систему рекомендацій.

Здійснений аналіз показав, що основні платформи для доступу до відеоконтенту орієнтовані на надання персоналізованих рекомендацій з метою вирішення комерційних завдань. Більшість таких сервісів базує свої рекомендації на платних підписках. Це створює потребу у розробці систем, які враховують інтереси та потреби конкретного користувача, забезпечуючи при цьому гнучкість у взаємодії з рекомендаційними алгоритмами.

Висновки за розділом 1

У першому розділі було розглянуто теоретичні та практичні основи побудови рекомендаційних систем, а також проаналізовано актуальні проблеми, які виникають під час їх проєктування та впровадження у цифрових сервісах.

Описано основні методи побудови рекомендацій, зокрема контентну фільтрацію, колаборативну фільтрацію та гібридні підходи. Визначено переваги та недоліки кожного з методів, а також умови, за яких їх доцільно використовувати.

Розглянуто специфіку формування персоналізованих рекомендацій у цифрових сервісах доступу до фільмів. Визначено, що успішна

рекомендаційна система повинна не лише точно прогнозувати вподобання користувача, але й враховувати контекст, змінність інтересів, та прагнути до балансу між релевантністю та новизною контенту.

РОЗДІЛ 2

ПРОЄКТУВАННЯ МОДЕЛІ ІНФОРМАЦІЙНОЇ СИСТЕМИ РЕКОМЕНДАЦІЙ ФІЛЬМІВ

2.1 Вибір і підготовка даних для проєктування моделі інформаційної системи рекомендацій фільмів

Для створення рекомендаційної системи було використано набір даних TMDb 5000 Movie Dataset, доступний на платформі Kaggle [7]. Цей датасет включає докладну інформацію про 5000 фільмів, зокрема такі характеристики, як назва, жанри, опис (overview), акторський склад, режисери, дата випуску, популярність, рейтинги користувачів та кількість голосів. Завдяки своїй актуальності та надійності, цей набір даних є чудовою основою для побудови точних і персоналізованих рекомендацій.

На етапі підготовки дані було попередньо оброблено з використанням бібліотек `pandas` і `pumpru`, а саме – здійснено злиття даних, видалення пропущених або зайвих значень, приведення форматів до однакової структури. Основні поля, використані в системі: `title`, `genres`, `overview`, `vote_average`, `popularity`, `cast`, `crew`.

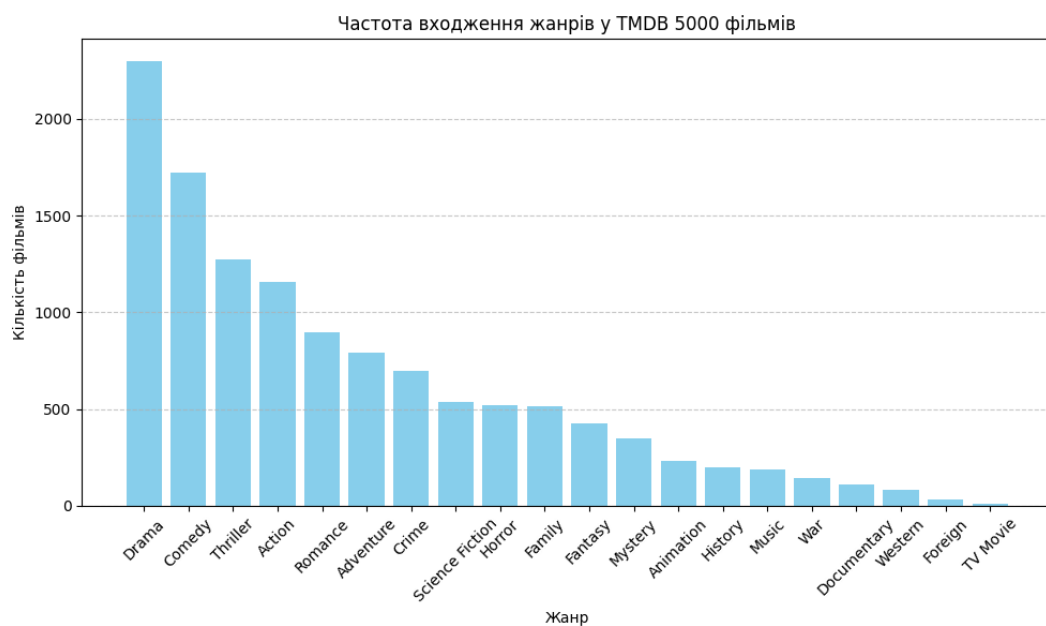


Рисунок 2.1 – Частота входження жанрів у TMDb.

Для базової візуалізації частоти входження жанрів у вибрану множину фільмів було побудовано графік за допомогою бібліотеки `matplotlib`, що дозволяє оцінити розподіл жанрового складу у вибірці.

На рис. 2.1 зображено частоту входження жанрів у TMBD.

На рис. 2.2 представлено лінійний графік, який ілюструє динаміку середнього доходу за різними жанрами кінофільмів залежно від року їхнього виходу. Горизонтальна вісь відображає рік випуску фільмів (від 1980 до 2016 року), а вертикальна вісь – середній дохід у доларах США.

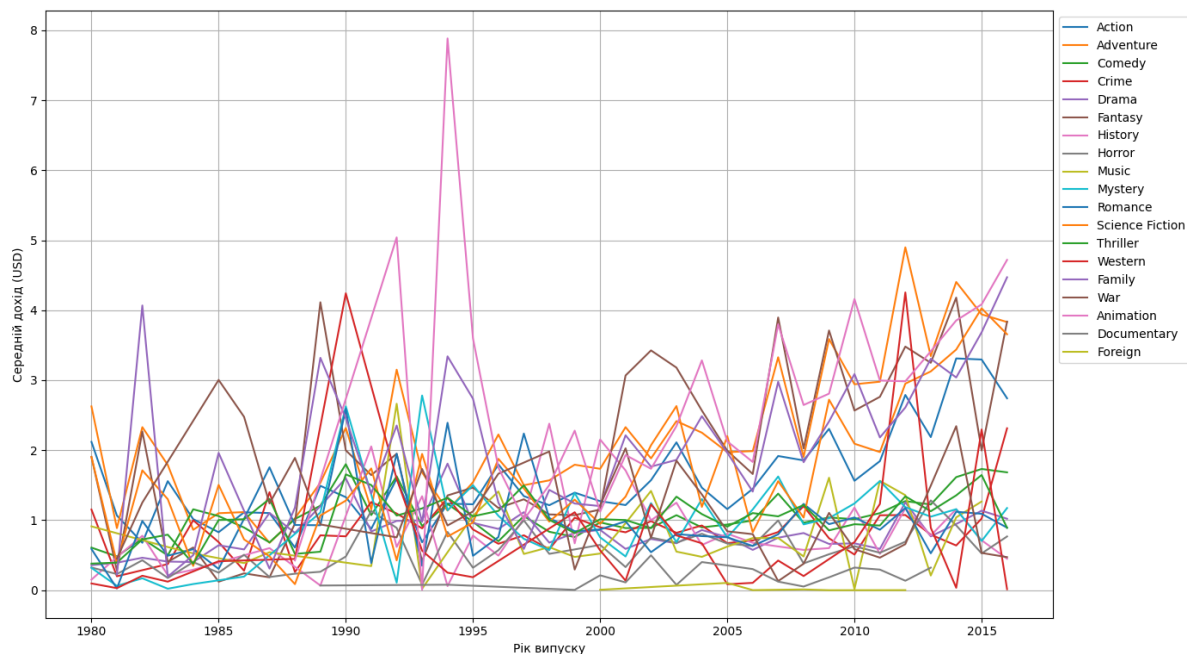


Рисунок 2.2 – Взаємозв'язок між роком випуску, прибутком і жанром фільмів.

Жанри «Fantasy», «Animation» та «Adventure» демонструють найбільші коливання і в окремі роки – зокрема у 1993 році – досягають пікових значень.

2.2 Фільтрація на основі контенту та алгоритм зваженого рейтингу

Фільтрація на основі контенту застосовує алгоритм, який вибирає елементи з подібним змістом, щоб рекомендувати фільми, виходячи з того, що вже було переглянуто або входить до кола уподобань користувача. Логіка роботи моделі, яка здійснює пошук фільмів за контентом, представлена на рисунку 2.3.



Рисунок 2.3 – Логічна структура моделі рекомендацій.

Першим етапом є фільтрація на основі рейтингу фільмів. Для цього ми використовуємо зважений рейтинг WR за наступною формулою:

$$WR = \frac{v}{v+m} R + \frac{m}{v+m} C, \quad (2.1)$$

де v – кількість голосів, які отримав фільм; m – мінімальна кількість голосів, необхідна для включення фільму до графіка; R – середня оцінка фільму; C – середній рейтинг усіх фільмів.

Середній рейтинг усіх фільмів приблизно дорівнює 6 за десятибальною шкалою. Наступним етапом є визначення мінімальної кількості голосів для відбору фільмів до графіку. Для цього застосовується 90-й процент –

враховуються лише ті фільми, кількість голосів у яких перевищує 90% інших. Результат сортування за рейтингом формується в список найпопулярніших фільмів, результат наведено на рисунку 2.4.

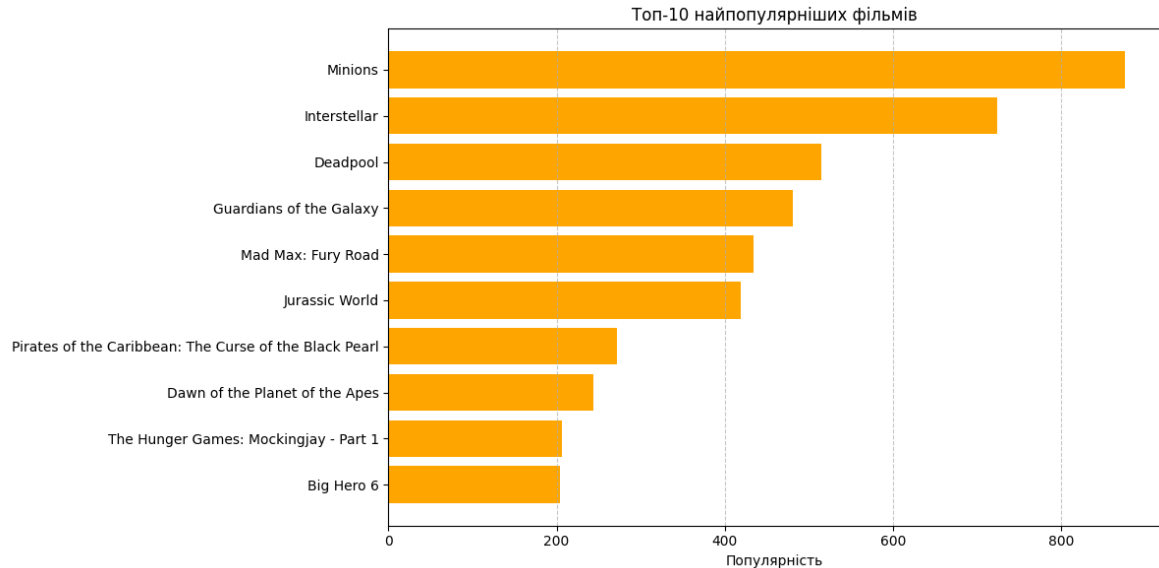


Рисунок 2.4 – Топ-10 найпопулярніших фільмів.

Наступним кроком є обчислення попарних значень схожості між усіма фільмами на основі їхніх описів (тегів) для формування рекомендацій схожого контенту. Для цього сюжети (теги) трансформуються у числові вектори за допомогою моделі Bag of Words із застосуванням CountVectorizer. Цей інструмент створює матрицю, де кожен рядок відповідає окремому фільму, а кожен стовпець – частоті появи конкретного слова (з обмеженням до 5000 найбільш уживаних слів та виключенням англійських стоп-слів).

Для покращення якості текстового представлення перед векторизацією застосовується стемінг (через PorterStemmer), який зводить слова до їх основних форм, що допомагає об'єднати схожі слова, наприклад, "loved" і "loving" у "love".

Для визначення подібності між двома фільмами застосовується косинусна міра схожості, яка оцінює кут між векторами, що відображають

фільми у багатовимірному просторі слів. Косинусна схожість ігнорує довжину векторів (масштаб тексту) і забезпечує швидкі обчислення.

Математично косинусна подібність між векторами A і B визначається формулою:

$$\text{similarity}(A,B) = \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}}, \quad (2.2)$$

де $A \cdot B$ – скалярний добуток векторів; $\|A\|$ та $\|B\|$ – їхні евклідові норми; n – розмірність векторів.

При проектуванні використовуємо функцію `cosine_similarity` з бібліотеки `sklearn.metrics.pairwise`, щоб отримати матрицю попарної подібності між фільмами.

Для реалізації рекомендацій створюється функція, яка на вході приймає назву фільму та повертає список із 5 найбільш подібних фільмів за змістом. Основні кроки роботи алгоритму такі:

1. Отримати індекс фільму за його назвою.
2. Взяти відповідний рядок із матриці косинусної подібності – він містить оцінки подібності цього фільму до всіх інших.
3. Перетворити отриманий результат на список кортежів у форматі: (індекс фільму, значення подібності)
4. Відсортувати список за значенням подібності у спадяючому порядку.
5. Обрати п'ять найбільш схожих фільмів, виключивши перший результат, який є самим фільмом-запитом.
6. Повернути список назв найбільш схожих фільмів.

Після побудови векторної моделі та обчислення косинусної подібності ця функція дозволяє здійснювати рекомендації фільмів на основі їхнього змісту.

2.3 Проектування бази даних

У рамках даної моделі було прийнято рішення використати вбудовану реляційну систему керування базами даних SQLite, яка є легкою, не вимагає окремого сервера для функціонування та добре підходить для десктопних додатків та проєктів малого й середнього рівня.

На рисунку 2.5 наведена модель бази даних.

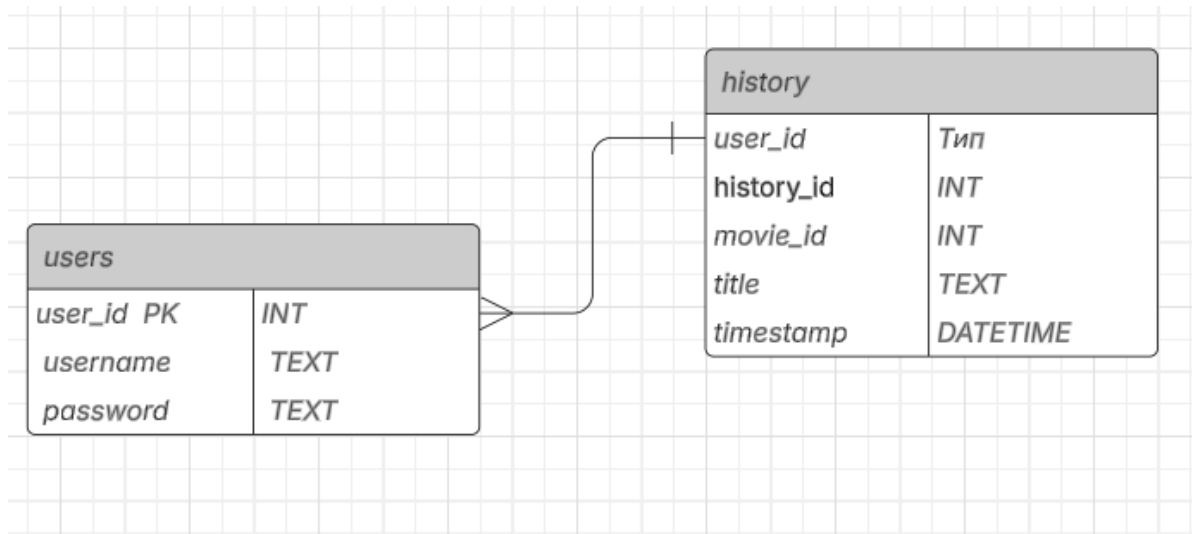


Рисунок 2.5 – Модель бази даних SQLite.

Таблиця **users** відповідає за зберігання облікових даних користувачів. Вона містить такі поля:

- `user_id` – ціле число, первинний ключ, що автоматично інкрементується;
- `username` – текстове поле, унікальне ім'я користувача;
- `password` – текстове поле, що зберігає пароль користувача.

Таблиця **history** фіксує історію рекомендацій для кожного користувача. Структура таблиці має такі атрибути:

- `history_id` – первинний ключ;
- `user_id` – зовнішній ключ, який посилається на `users.user_id`;
- `movie_id` – ідентифікатор фільму (пов'язаний з внутрішнім набором даних рекомендацій);

- `title` – назва фільму;
- `timestamp` – дата й час створення запису, що автоматично встановлюється на момент вставки запису.

Таблиці пов'язані між собою відношенням типу «один до багатьох»: один користувач може мати багато записів у таблиці *history*. Це забезпечується за рахунок зовнішнього ключа *user_id*, що зберігається в таблиці *history*.

Модель інформаційної системи рекомендацій фільмів:

1. Модуль введення та автентифікації користувача – відповідає за збір базової інформації про користувача, обробку логінів, паролів і створення персоналізованого сеансу взаємодії.

2. Модуль обробки вхідних даних – здійснює попередню обробку датасету фільмів, очищення та нормалізацію текстових описів, а також побудову векторних представлень.

3. Модуль векторизації та обчислення подібності – реалізує перетворення текстових характеристик фільмів (жанри, опис, ключові слова тощо) у числову форму за допомогою інструменту `CountVectorizer`. Далі розраховується ступінь схожості між фільмами за допомогою косинусної метрики (*cosine similarity*).

4. Модуль генерації рекомендацій – на основі обраного фільму або вподобань користувача формує список релевантних фільмів, відсортованих за спаданням коефіцієнта подібності.

5. Модуль візуалізації – відповідає за графічне подання результатів пошуку, рейтингів, жанрових розподілів, що сприяє кращому сприйняттю інформації.

6. Модуль збереження історії переглядів – інтегрований із базою даних `SQLite` та виконує функцію логування активності користувача з метою подальшого аналізу його вподобань.

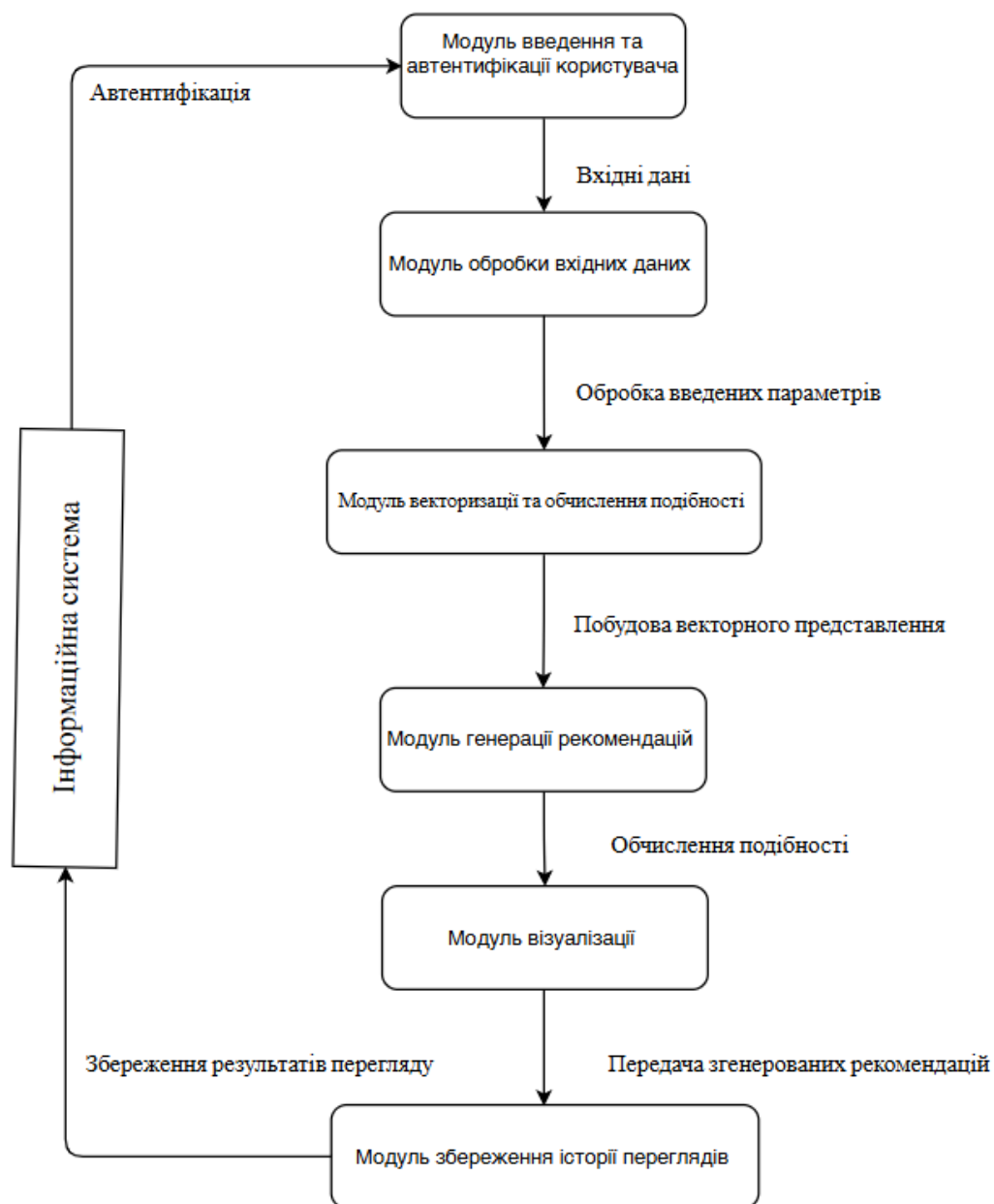


Рисунок 2.6 – Концептуальна схема моделі інформаційної системи рекомендацій фільмів.

Висновки за розділом 2

У другому розділі було розглянуто основні етапи проєктування інформаційної системи рекомендацій фільмів. Було здійснено вибір і підготовку даних на основі якісного датасету TMDb 5000, що містить релевантну інформацію про сучасні кінострічки. Проведено попередню

обробку даних, зокрема очищення, об'єднання та нормалізацію ключових полів, що забезпечило коректну роботу подальших алгоритмів.

Для побудови системи персоналізованих рекомендацій реалізовано фільтрацію на основі контенту, яка базується на зваженому рейтингу та обчисленні текстової подібності між описами фільмів. Використано модель Bag of Words із застосуванням стемінгу, що підвищило точність представлення контенту. Для визначення схожості фільмів застосовано косинусну метрику, яка дала змогу сформуванню релевантні рекомендації на основі змісту.

Окрему увагу приділено проєктуванню бази даних системи. Було створено дві основні таблиці – users та history – які забезпечують збереження облікових даних користувачів і реєстрацію їхньої історії взаємодії з рекомендаційною системою.

РОЗДІЛ 3

РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ РЕКОМЕНДАЦІЙ ФІЛЬМІВ

3.1 Обґрунтування обраних технологій та засобів розробки інформаційної системи рекомендацій фільмів

У процесі розробки інформаційної системи рекомендацій фільмів було прийнято обґрунтоване рішення щодо вибору мови програмування, середовища розробки, інструментів обробки даних, візуалізації та зберігання інформації. Ключовим чинником при виборі стало прагнення забезпечити максимальну гнучкість, модульність, масштабованість та легкість інтеграції компонентів системи. З огляду на це основною технологічною платформою було обрано мову програмування Python у поєднанні з інтерактивним середовищем Jupyter Notebook та вбудованою системою управління базами даних SQLite [8].

Мова Python є однією з найпопулярніших у сучасному програмному інжинірингу, особливо в галузях обробки даних, машинного навчання, статистичного аналізу, штучного інтелекту та побудови рекомендаційних систем. Python характеризується лаконічним синтаксисом, високою читабельністю коду, підтримкою великої кількості спеціалізованих бібліотек та активною спільнотою розробників. Серед основних переваг Python, які стали вирішальними при виборі:

- широкий набір готових рішень для реалізації контентно-орієнтованих фільтрацій;
- потужні інструменти для обробки великих обсягів структурованої та неструктурованої інформації;
- простота інтеграції з іншими мовами, базами даних, API та вебтехнологіями;

- ефективна підтримка роботи з векторами, матрицями та операціями лінійної алгебри.

У якості середовища розробки обрано Jupyter Notebook – інтерактивну оболонку для виконання коду Python, що забезпечує покрокову обробку даних з можливістю документування кожного етапу. Середовище підтримує відображення тексту, формул, таблиць і графіків у єдиному робочому просторі, що є особливо зручним для аналітичної роботи та прототипування моделей. Основні причини вибору Jupyter Notebook:

- підтримка інтерактивного виконання коду з миттєвим відображенням результатів;
- можливість вбудованої візуалізації даних;
- зручна структура для тестування і демонстрації окремих модулів системи;
- підтримка Markdown-розмітки та математичних формул у форматі LaTeX [9].

Для реалізації функціональності системи було використано низку спеціалізованих бібліотек Python:

- *pandas* – бібліотека для обробки табличних даних, що дозволяє зчитувати, фільтрувати, об'єднувати та трансформувати великі масиви інформації з високою швидкістю;
- *numpy* – бібліотека для виконання високопродуктивних обчислень із багатовимірними масивами, яка лежить в основі чисельної обробки;
- *scikit-learn* – набір інструментів для аналізу даних і машинного навчання, зокрема для обчислення косинусної подібності (cosine_similarity) між векторами ознак;
- *nltk* (Natural Language Toolkit) – бібліотека для лінгвістичної обробки текстів, що забезпечує виконання таких процедур, як токенізація, стемінг, видалення стоп-слів тощо;

- *CountVectorizer* – засіб перетворення текстових даних у числову форму за моделлю «мішка слів», що дозволяє векторизувати опис фільмів для подальшого аналізу;
- *matplotlib* і *seaborn* – бібліотеки для побудови статистичних та аналітичних графіків, необхідних для візуалізації жанрового розподілу, динаміки прибутків тощо.

3.2 Опис моделі інформаційної системи рекомендацій фільмів

Інформаційна система (див. рис. 2.6) побудована за принципами модульної архітектури й включає низку логічно взаємопов'язаних компонентів:

1. Модуль введення та автентифікації користувача – відповідає за збір базової інформації про користувача, обробку логінів, паролів і створення персоналізованого сеансу взаємодії.
2. Модуль обробки вхідних даних – здійснює попередню обробку датасету фільмів, очищення та нормалізацію текстових описів, а також побудову векторних представлень.
3. Модуль векторизації та обчислення подібності – реалізує перетворення текстових характеристик фільмів (жанри, опис, ключові слова тощо) у числову форму за допомогою інструменту *CountVectorizer*. Далі розраховується ступінь схожості між фільмами за допомогою косинусної метрики (*cosine similarity*).
4. Модуль генерації рекомендацій – на основі обраного фільму або вподобань користувача формує список релевантних фільмів, відсортованих за спаданням коефіцієнта подібності.
5. Модуль візуалізації – відповідає за графічне подання результатів пошуку, рейтингів, жанрових розподілів, що сприяє кращому сприйняттю інформації.

6. Модуль збереження історії переглядів – інтегрований із базою даних SQLite та виконує функцію логування активності користувача з метою подальшого аналізу його вподобань.

Основним принципом побудови моделі є обчислення векторної схожості між фільмами на основі ознак, таких як жанр, ключові слова, актори, режисери, опис сюжету тощо. Фільми, які мають найвищу подібність до тих, що були переглянуті користувачем або ним обрані, вважаються релевантними і пропонуються для перегляду.

На рисунку 3.1 представлено зовнішній вигляд сторінки входу до системи, реалізованої з використанням мови програмування Python у середовищі Streamlit, що дозволяє швидко створювати інтерактивні вебінтерфейси.

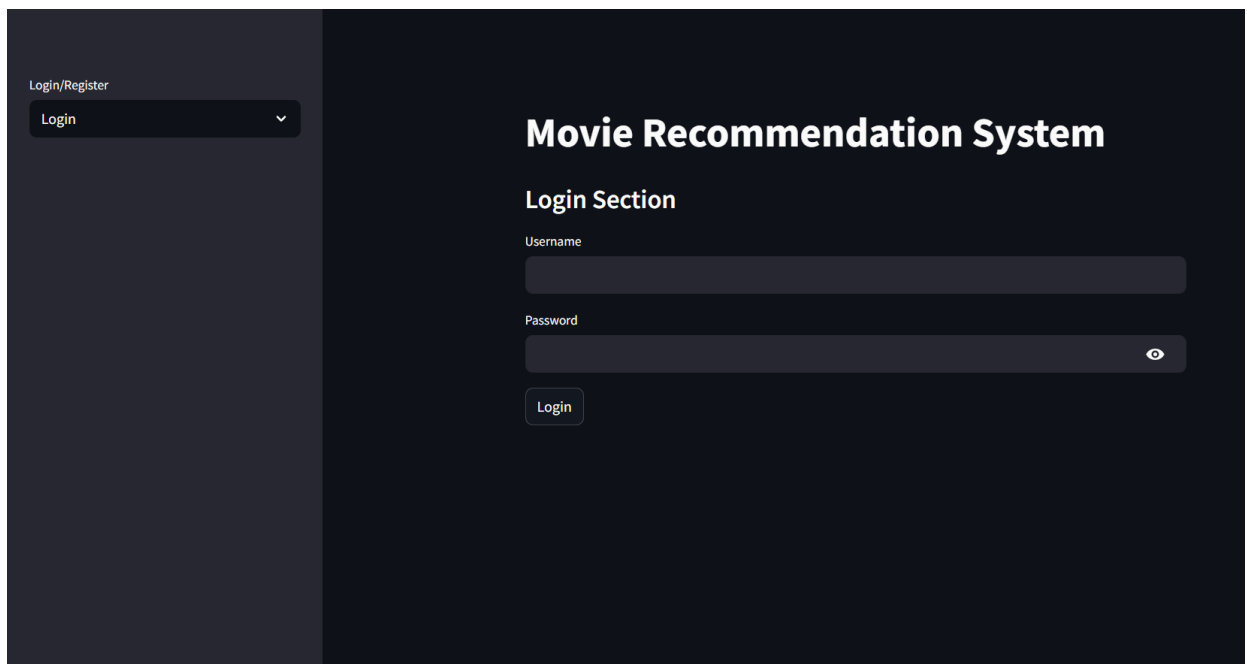


Рисунок 3.1 – Сторінка входу до ситеми.

Основні елементи сторінки входу:

- Поле введення логіна (username) – обов’язкове для заповнення текстове поле, в якому користувач вводить унікальне ім’я.

- Поле введення пароля (password) – текстове поле з прихованим введенням, яке забезпечує базову конфіденційність.
- Кнопка «Login» – підтверджує введені дані та викликає перевірку автентичності на основі збережених у базі даних облікових записів.
- Кнопка «Register» – перенаправляє на форму реєстрації нового користувача, якщо обліковий запис ще не створено.

Процес входу реалізується через виклик функції `login_user(username, password)`, яка взаємодіє з базою даних SQLite. При правильному введенні облікових даних відбувається вхід до системи; у разі помилки користувач отримує відповідне повідомлення (рис 3.2).

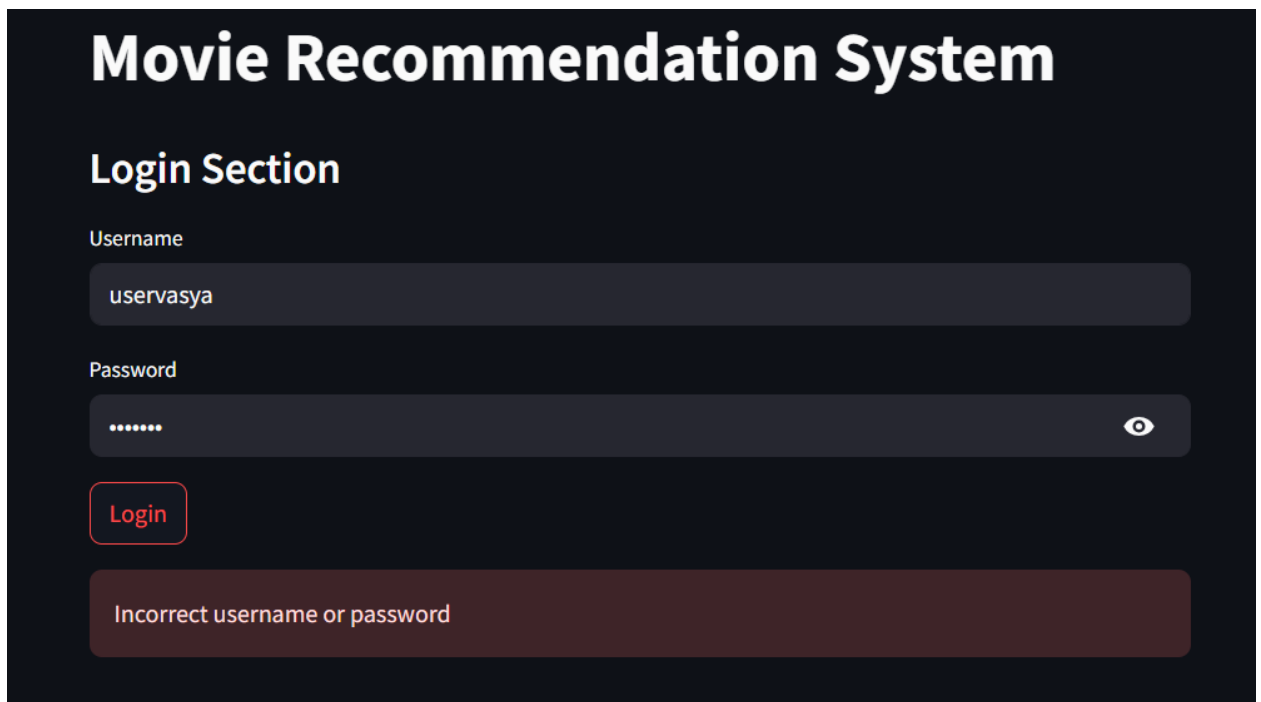


Рисунок 3.2 – Помилка входу у систему.

У разі успішної автентифікації система відображає привітальне повідомлення у вигляді динамічного текстового елементу «Welcome <username>» (рис. 3.3).

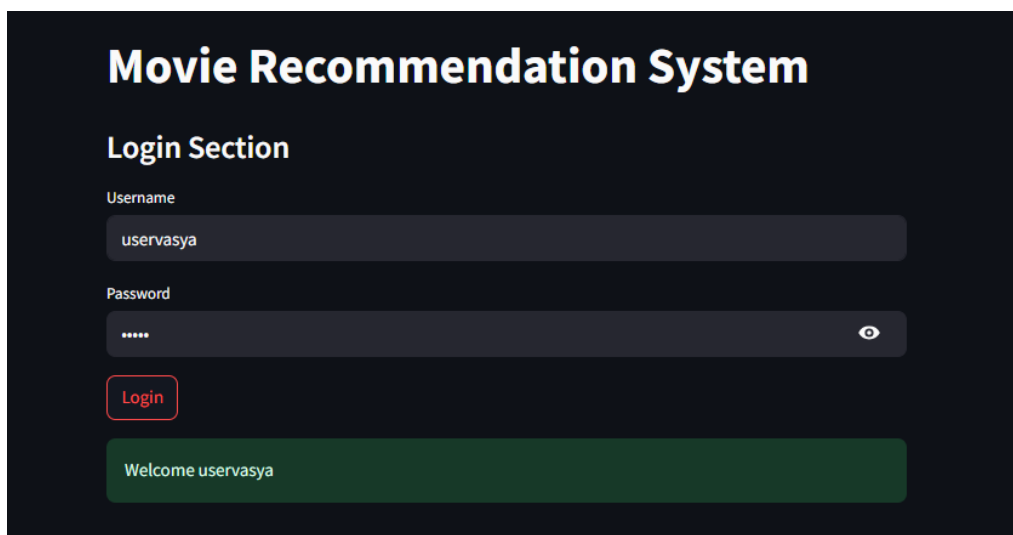


Рисунок 3.3 – Успішна автентифікація у системі.

Однією з ключових функцій є можливість швидкого пошуку фільму за його назвою (рис. 3.4). Така функція забезпечує зручний доступ до персоналізованих рекомендацій, дозволяючи користувачеві миттєво отримати результати, що ґрунтуються на змістовній подібності.



Рисунок 3.4 – Функція пошуку фільму.

На рисунку 3.5 а), б) зображено приклад інтерфейсу, в якому користувач здійснює пошук за назвою фільму «I Am Legend» (укр. «Я – легенда»). Після підтвердження введеної назви система обчислює ступінь подібності між описом цього фільму та всіма іншими фільмами у датасеті. Результатом є список із п'яти найбільш схожих фільмів, який відображається на сторінці рекомендацій.

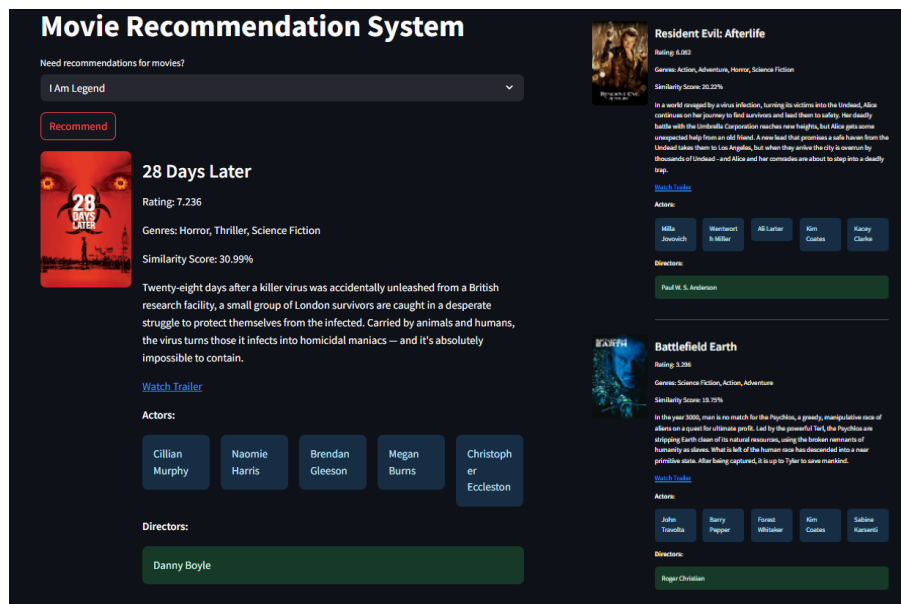


Рисунок 3.5 а)– Рекомендації на основі фільму «I Am Legend».

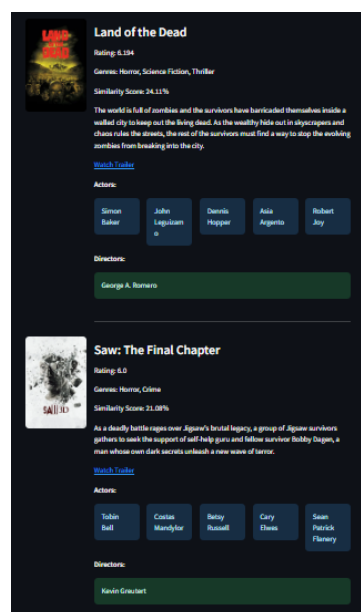


Рисунок 3.5 б)– Рекомендації на основі фільму «I Am Legend».

Після формування списку рекомендованих фільмів, поруч із кожним найменуванням відображається інтерактивне посилання, за допомогою якого користувач може перейти до офіційного трейлера фільму на платформі YouTube (рис. 3.6).

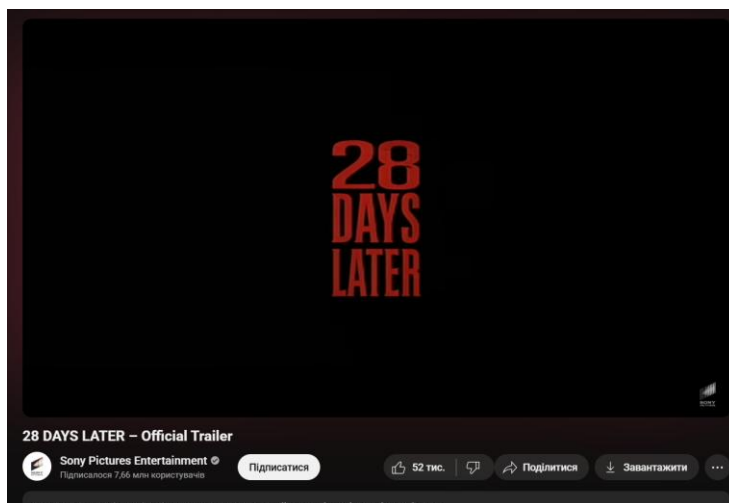


Рисунок 3.6 – Функція переходу до трейлера фільму.

Кожного разу, коли користувач звертається до функції отримання рекомендацій, система реєструє запит у відповідній таблиці бази даних (history). На рисунку 3.7 зображено інтерфейс відображення історії рекомендацій фільмів.



Рисунок 3.7 – Інтерфейс відображення історії рекомендацій.

3.3 Забезпечення якості користувацького досвіду

Інтерфейс користувача і рекомендацій

Ефективність інформаційної системи значною мірою залежить від зручності та інтуїтивної зрозумілості інтерфейсу користувача. Важливим є забезпечення швидкого доступу до релевантних рекомендацій без необхідності здійснення додаткових дій або переміщення між розділами системи. Застосування принципів адаптивного дизайну дає змогу створити уніфікований досвід на різних типах пристроїв – від смартфонів до настільних комп'ютерів. Крім того, важливою є візуальна ієрархія інформації: найрелевантніші пропозиції мають розміщуватись у зоні першочергової уваги користувача.

UX-фактори та зручність навігації

Ключовими факторами, що впливають на користувацький досвід (UX), є простота використання, зрозуміла навігація, швидкість реакції системи та релевантність рекомендацій. У процесі розробки інтерфейсу важливо враховувати когнітивне навантаження на користувача, а також зменшення кількості кліків для досягнення цілі.

Впровадження таких елементів, як **автоматичне сортування за інтересами, підказки, інтерактивні фільтри та особисті добірки**, дозволяє значно зменшити час на пошук контенту. Окрему увагу слід приділити **зворотному зв'язку**: можливість оцінити рекомендації чи повідомити про їхню нерелевантність допомагає покращити точність подальших прогнозів.

Візуалізація результатів та адаптація під користувача

Окрім точності моделей, важливим чинником є спосіб представлення результатів рекомендацій. Ефективна візуалізація забезпечує не лише зручність сприйняття, але й підвищує довіру користувача до системи. Для цього застосовуються:

- карткові інтерфейси (card-based UI);
- візуальні рейтинги та позначки;

- попередній перегляд опису, трейлерів або жанрових ознак;
- можливість сортування й перемикання між типами добірок.

Система має підтримувати **адаптацію** під стиль споживання інформації конкретного користувача: зміна способу відображення (таблиця/плитка), мови інтерфейсу, пріоритетів у видачі рекомендацій тощо.

Роль рекомендацій у покращенні взаємодії

Рекомендаційні системи відіграють вирішальну роль у підвищенні рівня залученості користувачів. Їх застосування дозволяє скоротити час на пошук контенту, стимулює повторні відвідування платформи, формує відчуття індивідуального підходу.

Особливе значення має **контекстна релевантність**: рекомендації, які враховують поточну активність, час доби, сезонність або минулі вподобання, демонструють вищі показники точності та клікабельності.

Таким чином, впровадження персоналізованих рекомендацій сприяє покращенню загального враження від взаємодії з платформою та підвищенню її конкурентоспроможності.

Висновки за розділом 3

У третьому розділі було розглянуто процес реалізації інформаційної системи рекомендацій фільмів, спрямованої на покращення користувацького досвіду за допомогою контентно-орієнтованого підходу.

Проведено аналіз та обґрунтовано вибір технологій, інструментів і засобів, використаних у процесі розробки. Описано архітектурну модель інформаційної системи. Визначено функціональне призначення основних модулів: автентифікації користувача, обробки вхідних даних, обчислення подібності, генерації рекомендацій, візуалізації результатів та ведення історії взаємодії.

Особливу увагу приділено реалізації інтерфейсу системи за допомогою фреймворку Streamlit. Послідовно продемонстровано ключові елементи графічного інтерфейсу, включаючи сторінку входу, форму

пошуку фільму, виведення рекомендацій на основі вибраного фільму, можливість перегляду трейлерів та перегляду історії запитів із фіксацією дати й часу.

ВИСНОВКИ

У межах кваліфікаційної роботи було успішно досліджено, спроектовано та реалізовано інформаційну систему рекомендацій фільмів на основі контентно-орієнтованого підходу. Робота охопила повний цикл створення інтелектуального програмного рішення – від теоретичного аналізу та вибору методів до побудови архітектури системи та реалізації її функціональності в інтерактивному середовищі.

Актуальність цієї роботи обумовлена потребою створення алгоритмічної моделі, здатної забезпечити високий рівень релевантності рекомендацій навіть за умов обмежених даних, що має як наукове значення – у контексті вдосконалення методів машинного навчання і гібридних підходів – так і практичну користь для цифрових платформ, які прагнуть підвищити залучення користувачів.

У першому розділі розглянуто теоретичні основи побудови рекомендаційних систем. Проаналізовано сучасні підходи до фільтрації контенту – контентну, колаборативну та гібридну. Визначено ключові проблеми, пов'язані з точністю, релевантністю, новизною рекомендацій і зміною інтересів користувачів. Окрему увагу приділено особливостям рекомендацій у сфері цифрового кіноконтенту, де важливу роль відіграє персоналізація.

У другому розділі описано процес проєктування системи рекомендацій, починаючи з вибору джерела даних та закінчуючи підготовкою текстової інформації для подальшого аналізу. Проведено очистку, нормалізацію, а також трансформацію даних у формат, придатний для обчислення схожості між фільмами. Для цього було використано модель "мішка слів" та косинусну міру подібності, яка дозволила побудувати ефективний алгоритм генерації рекомендацій.

У третьому розділі здійснено практичну реалізацію інформаційної системи за допомогою мови Python, інтерактивного середовища Jupyter Notebook, фреймворку Streamlit та бази даних SQLite. Обґрунтовано вибір технологій і бібліотек, які забезпечують ефективну обробку, аналіз і візуалізацію даних. Побудовано архітектуру системи, що складається з модулів автентифікації, обробки даних, векторизації, генерації рекомендацій, візуалізації результатів і збереження історії переглядів.

В роботі описано інформаційну взаємодію між користувачем і цифровою платформою розважального контенту, що реалізується через рекомендаційну систему з модулями збору даних, обробки користувацької інформації, побудови моделей машинного навчання та генерації персоналізованих рекомендацій.

Система має простий і зручний вебінтерфейс, що дозволяє користувачеві входити до системи, виконувати пошук за назвою фільму, отримувати релевантні рекомендації на основі змістовної подібності, переглядати трейлери рекомендованих фільмів і бачити історію своїх запитів. Усі запити логуються в базі даних із фіксацією часу, що забезпечує основу для подальшого аналізу користувацьких вподобань.

Отримані результати підтверджують ефективність застосованих методів і рішень у задачах формування персоналізованих рекомендацій у цифрових медіасервісах. Система може бути основою для подальшого розширення: впровадження колаборативної фільтрації, врахування оцінок користувачів, автоматичного навчання на основі збереженої історії, розробки мобільної версії або інтеграції з іншими платформами через API.

Проведене дослідження підтвердило доцільність і ефективність використання гібридних моделей рекомендацій для досягнення більшої точності у передбаченні вподобань користувача. Отримані результати можуть бути використані як основа для створення реального програмного

продукту у сфері онлайн-кінотеатрів, стримінгових сервісів та цифрових платформ розваг.

Основними напрямками застосування моделі інформаційної системи рекомендацій фільмів є:

1. **Платформи потокового відео (Streaming services):** покращення точності рекомендацій, персоналізація добору контенту для кожного користувача.
2. **Онлайн-кінотеатри та відеобібліотеки:** впровадження інтелектуальних систем навігації та добору фільмів для великих цифрових архівів фільмів і серіалів.
3. **Мобільні застосунки та Smart TV-платформи:** автоматизація побудови індивідуальних плейлистів для фільмів і серіалів з урахуванням контексту використання (час доби, жанрові вподобання, попередній досвід перегляду).
4. **Освітні платформи, що використовують фільми як навчальний інструмент:** рекомендація навчальних відео або фільмів з урахуванням рівня знань, тематики курсу або мови викладання.
5. **Маркетинг і просування контенту:** застосування рекомендаційних механізмів для аналізу поведінки користувача, підвищення ефективності кампаній просування відеоконтенту.
6. **UX/UI-дизайн та розробка цифрових продуктів:** формування інтерфейсів користувача з урахуванням адаптивних рекомендацій, що знижує когнітивне навантаження та покращує досвід взаємодії.

Таким чином, розроблена концептуальна модель може слугувати основою для подальшої інтеграції у повнофункціональні системи, орієнтовані на користувача, в умовах стрімкого розвитку цифрових технологій та індустрії розваг.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Li L., Chu W., Langford J., Schapire R. E. A contextual-bandit approach to personalized news article recommendation // *Proceedings of the 19th International Conference on World Wide Web*. – 2010. – P. 661–670.
2. Bugaychenko D., Dzuba A. Musical recommendations and personalization in a social network // *RecSys '13: Proceedings of the 7th ACM Conference on Recommender Systems*. – October 2013. – P. 367–370.
3. Auer P., Cesa-Bianchi N., Fischer P. Finite-time analysis of the multiarmed bandit problem // *Machine Learning*. – 2002. – Vol. 47, No. 2–3. – P. 235–256.
4. Квашук В. В. Машинне навчання: методи, алгоритми, задачі. – Львів: Видавництво ЛНУ імені Івана Франка, 2017. – 256 с.
5. Омаров Р. З., Востротіна А. В., Лі А. Д. Проблема "холодного старту" // *Молодий учений*. – 2019. – № 26 (264). – С. 85–88.
6. Cano E., Morisio M. Hybrid Recommender Systems: A Systematic Literature Review [Електронний ресурс]. – Режим доступу: <https://arxiv.org/abs/1901.03888> (дата звернення: 23.05.2025).
7. Ricci F., Rokach L., Shapira B. Introduction to Recommender Systems Handbook. – Springer, 2011.
8. TMDb Movie Metadata Dataset [Електронний ресурс]. – Режим доступу: <https://www.kaggle.com/datasets/tmdb/tmdb-movie-metadata> (дата звернення: 23.05.2025).
9. Python Software Foundation. Python Language Reference, version 3.x [Електронний ресурс]. – Режим доступу: <https://www.python.org/doc/> (дата звернення: 26.05.2025).
10. Project Jupyter. Jupyter Notebook Documentation [Електронний ресурс]. – Режим доступу: <https://jupyter.org/documentation> (дата звернення: 26.05.2025).

11. Streamlit Inc. Streamlit Documentation [Электронный ресурс]. – Режим доступа: <https://docs.streamlit.io/> (дата звернения: 26.05.2025).

ДОДАТКИ

Додаток А

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Харківський національний університет імені В. Н. Каразіна

Навчально-науковий інститут комп'ютерних наук та штучного інтелекту
Кафедра комп'ютерних систем та робототехніки
Рівень вищої освіти (освітньо-кваліфікаційний рівень) **Бакалавр**
Галузь знань: 12 – Інформаційні технології
Спеціальність: 123 «Комп'ютерна інженерія»
Освітня програма «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри комп'ютерних
систем та робототехніки
к. ф.-м. н., доц. ХРУСЛОВ М. М.
«02» жовтня 2024 року

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

СУХИНА Василь Васильович
(прізвище, ім'я, по батькові студента)

1. Тема роботи **«МОДЕЛЬ ІНФОРМАЦІЙНОЇ СИСТЕМИ РЕКОМЕНДАЦІЙ ФІЛЬМІВ ДЛЯ ПОКРАЩЕННЯ КОРИСТУВАЦЬКОГО ДОСВІДУ»**

керівник роботи **Мороз Ольга Юріївна**, PhD з інформ. технологій, доцент з во
кафедри КСР,

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від **16 квітня 2025 року № 4101-5/962**

2. Строк подання студентом роботи **30 травня 2025 року**

3. Перелік питань, які потрібно розробити:

- 1) Аналіз сучасних інформаційних систем для вибору фільмів.
- 2) Аналіз рекомендаційних алгоритмів та їх застосування в подібних системах.
- 3) Вибір підходу для побудови рекомендаційних моделей.
- 4) Проектування інтерфейсу користувача для зручної взаємодії із системою.
- 5) Реалізація та тестування алгоритму.

4. План роботи

№ з/п	Назви етапів роботи	Термін виконання етапів роботи
1	Обґрунтування актуальності теми	2.10.2024-25.11.2024
2	Огляд сучасних інформаційних систем для вибору фільмів.	26.11.2024-17.12.2024
3	Аналіз рекомендаційних алгоритмів та їх застосування в подібних системах.	02.01.2025-29.01.2025
4	Вибір підходу для побудови рекомендаційних моделей.	03.02.2025-02.03.2025
5	Розробка рекомендаційної моделі.	03.03.2025-31.03.2025
6	Реалізація та тестування алгоритму	04.04.2025-15.04.2025
7	Розробка інтерфейсу системи та тестування	16.04.2025-17.05.2025
8	Підготовка та оформлення звітних матеріалів	01.05.2025-30.05.2025-
9	Оформлення пояснювальної записки та додатків	01.05.2025-30.05.2025-
10	Представлення кваліфікаційної роботи керівнику та рецензенту	01.05.2025-30.05.2025-

5. Дата видачі завдання *02 жовтня 2024 року.*

Студент

Сушина В. В.

ініціали, прізвище


 підпис

Керівник роботи

Мороз О. Ю.

ініціали, прізвище


 підпис

Додаток Б

Затверджую

« _____ » _____ 2025 р.

Технічне завдання

на розробку програмного виробу «Модель інформаційної системи рекомендацій фільмів для покращення користувацького досвіду»

1.	Введення	1.1. Назва: Модель інформаційної системи рекомендацій фільмів для покращення користувацького досвіду 1.2. Галузь застосування: Цифрові розваги.
2.	Підстава для розробки	2.1. Навчальний план за спеціальністю 123 – Комп'ютерна інженерія 2.2. Завдання на кваліфікаційну роботу бакалавра № 4101-5/962 від «16» квітня 2025 (представити як Додаток А до пояснювальної записки до кваліфікаційної роботи).
3.	Призначення розробки	3.1. Мета розробки: розробка моделі інформаційної системи рекомендацій фільмів, яка забезпечує точне і персоналізоване надання контенту користувачам. 3.2. Призначення розробки забезпечення ефективного функціонування системи рекомендацій фільмів шляхом аналізу вподобань користувачів, історії переглядів та взаємодії з контентом для підвищення задоволеності, залученості та тривалості використання платформи. 3.3. Вихідні дані розробки: вхідні – Поведінкові дані користувача, вихідні – список персоналізованих рекомендацій фільмів.
4.	Технічні вимоги до програмного виробу	4.1. Вимоги до функціональних характеристик: система повинна використовувати оновлену інформацію про фільми (жанри, рейтинги, новинки тощо), генерація персоналізованого списку фільмів на основі аналізу вхідних даних. 4.2. Вимоги до надійності: забезпечення безперебійної роботи програмного виробу при будь-яких вимогах користувача в рамках призначення виробу . 4.3. Вимоги до умов експлуатації: немає

		4.4. Вимоги до складу і параметрів технічних засобів: для виконання програми повинен підходити ПК із будь-якою операційною системою сімейства Windows, Linux/Unix, Mac OS X, OS/2, Amiga. Крім того, для роботи потрібний інтерпретатор мови програмування. 4.5. Вимоги до інформаційної та програмної сумісності: підтримка ОС Linux або Windows 10\11, підтримка мови програмування, підтримка різних платформ. 4.6. Вимоги до маркування та упаковки: вимоги до маркування та упакування не представляються. 4.7. Вимоги до транспортування і зберігання: вимоги до транспортування та зберігання не представляються. 4.8. Спеціальні вимоги: спеціальні вимоги до програмного виробу не пред'являються.	
5.	Вимоги до програмної документації	Програмною документацією до виробу «Модель інформаційної системи рекомендацій фільмів для покращення користувацького досвіду» вважати: 1) Справжнє Технічне завдання на розробку виробу (представити у вигляді Додатку Б до пояснювальної записки до кваліфікаційної роботи). 2) Методику розрахунку інформативності змінних стану (у вигляді глав 3.2 та 3.3 пояснювальної записки до кваліфікаційної роботи). 3) Опис виробу (представити в розділі 3 пояснювальної записки до кваліфікаційної роботи)	
6.	Вимоги до техніко-економічних показників	Програмною документацією до виробу «Модель інформаційної системи рекомендацій фільмів для покращення користувацького досвіду» вважати: 1) Справжнє Технічне завдання на розробку виробу (представити у вигляді Додатку Б до пояснювальної записки до кваліфікаційної роботи). 2) Опис програмного виробу (представити в Розділі 3 пояснювальної записки до кваліфікаційної роботи). 3) Джерела базової інформації.	
7.	Стадії і етапи розробки	Дата	Назва етапу
		2.10.2024-25.11.2024	Обґрунтування актуальності теми
		26.11.2024-17.12.2024	Огляд сучасних інформаційних систем для вибору фільмів.
		02.01.2025-29.01.2025	Аналіз рекомендаційних алгоритмів та їх застосування в подібних системах.

		03.02.2025-02.03.2025	Вибір підходу для побудови рекомендаційних моделей.
		03.03.2025-31.03.2025	Розробка рекомендаційної моделі.
		04.04.2025-15.04.2025	Реалізація та тестування алгоритму
		16.04.2025-17.05.2025	Розробка інтерфейсу системи та тестування
		17.05.2025-29.05.2025	Розробка пояснювальної записки.
		30.05.2025	Представлення кваліфікаційної роботи керівнику та
		30.05.2025	рецензенту. Оформлення пояснювальної записки та підготовка презентації.
8.	Порядок контролю і приймання програмного продукту (моделі)	1. Перевірку ходу розробки програми виконувати раз в 3 тижні. 2. Захист розробленої моделі провести на засіданні Атестаційної комісії. 3. Пояснювальну записку подати на паперових носіях в 1 примірнику і в електронному вигляді в 1 примірнику на CD-R компакт-диску.	

Виконавець
студент групи КІ- 41
Сушина В. В.



Замовник

канд. техн.наук, доцент
Мороз О. Ю.



Додаток В

**Програма і методика випробувань програмного виробу
«Модель інформаційної системи рекомендацій фільмів для покращення
користувацького досвіду»**

1. Об'єкт випробувань

1. Назва програмного виробу: «Модель інформаційної системи рекомендацій фільмів для покращення користувацького досвіду»
2. Галузь застосування: Інформаційні технології. А саме: платформи потокового відео (Streaming services); мобільні застосунки та Smart TV-платформи; освітні платформи, що використовують фільми як навчальний інструмент а також UX/UI-дизайн та розробка цифрових продуктів.
3. Перераховані відомості запозичуються з відповідних розділів Технічного завдання.

2. Мета випробувань

Перевірка відповідності функціональності програмної реалізації системи заявленим функціональним можливостям в технічному завданні (Додаток Б до пояснювальної записки до кваліфікаційної роботи).

3. Загальні положення**1. Підстави для проведення випробувань**

Підставою для проведення випробувань є наказ про призначення атестаційної комісії.

2. Місце і тривалість випробувань

Приймальні (приймально-здавальні) випробування проводяться на базі комп'ютерного класу кафедри в період роботи атестаційної комісії.

3. Обсяг випробувань

Приймальні випробування програмного виробу проводяться в обсязі відповідному цієї програми і методики випробувань.

4. Організації, які беруть участь у випробуваннях

Приймальні випробування проводяться атестаційною комісією напередодні засідання (або в процесі засідання) за участю Замовника, Виконавця та інших осіб, присутніх на засіданні.

4. Вимоги до програми або програмного виробу

Модель повинна задовольняти наступним вимогам:

- 4.1. Вимоги до функціональних характеристик: система повинна використовувати оновлену інформацію про фільми (жанри, рейтинги, новинки тощо), генерація персоналізованого списку фільмів на основі аналізу вхідних даних.

4.2. Вимоги до надійності: забезпечення безперебійної роботи програмного виробу при будь-яких вимогах користувача в рамках призначення виробу .

4.3. Вимоги до умов експлуатації: немає

4.4. Вимоги до складу і параметрів технічних засобів: для виконання програми повинен підходити ПК із будь-якою операційною системою сімейства Windows, Linux/Unix, Mac OS X, OS/2, Amiga. Крім того, для роботи потрібний інтерпретатор мови програмування.

4.5. Вимоги до інформаційної та програмної сумісності: підтримка ОС Linux або Windows 10\11, підтримка мови програмування, підтримка різних платформ.

4.6. Вимоги до маркування та упаковки: вимоги до маркування та упакування не представляються.

4.7. Вимоги до транспортування і зберігання: вимоги до транспортування та зберігання не представляються.

4.8. Спеціальні вимоги: спеціальні вимоги до програмного виробу не пред'являються.

5. Вимоги до програмної документації

Документацією до виробу «Модель інформаційної системи рекомендацій фільмів для покращення користувацького досвіду» вважати:

- 1) Документація по мові програмування та додаткові мануали.
- 2) Програму і методику випробувань розробленої програми (представити як Додаток В до пояснювальної записки до кваліфікаційної роботи).
- 3) Опис програмного виробу (представити в Розділі 3 пояснювальної записки до кваліфікаційної роботи).
- 4) Джерела базової інформації.

6. Засоби і порядок випробувань

6.1 Засоби випробувань

Засоби випробувань представлено на ПК на яких встановлено наступні програмні засоби: інтерпретатор мови програмування.

6.2 Порядок проведення випробувань

Як правило, випробування проводяться в два етапи:

- ознайомчий (1-й етап);
- власне випробування програмного виробу (2-й етап).

Перелік перевірок, що проводяться на 1 етапі випробувань, включає в себе:

1) Перевірку комплектності складу програмної документації здійснюється за критерієм наявності зазначеної в ТЗ документації.

2) Перевірку якості програмної документації. Перевірку здійснювати за критерієм відповідності вимогам ГОСТ 19.301-79 ЕСПД. «Програма і методика випробувань».

Перелік перевірок, що проводяться на 2 етапі випробувань, включає в себе:

- 1) Перевірку відповідності технічних характеристик програми вимогам технічного завдання.
 - 2) Перевірку ступеня виконання функціональних вимог до програми.
 - 3) Методику проведення перевірок:
 - a) Запустити програмне забезпечення.
 - b) Порядок проведення випробувань:
 - Зробити налаштування.
 - Перевірити чи працює програма.
 - Перевірити чи формується звіт.
 - 4) Якщо перевірки на першому та другому етапах виконано успішно, то виріб вважається таким, що пройшов випробування.
- Для проведення випробувань пропонується тест 1, тест 2 та тест 3.

Тест 1

1. Перевірка виконання програми;
2. Запуск головної сторінки додатку.

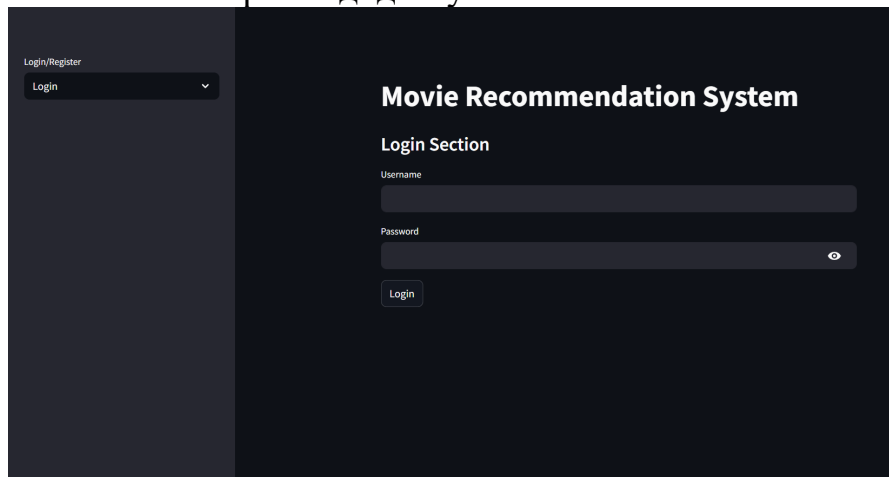


Рис. В.1 Тест 1

Тест 2

1. Перевірка виконання програми
2. Авторизація у додатку;

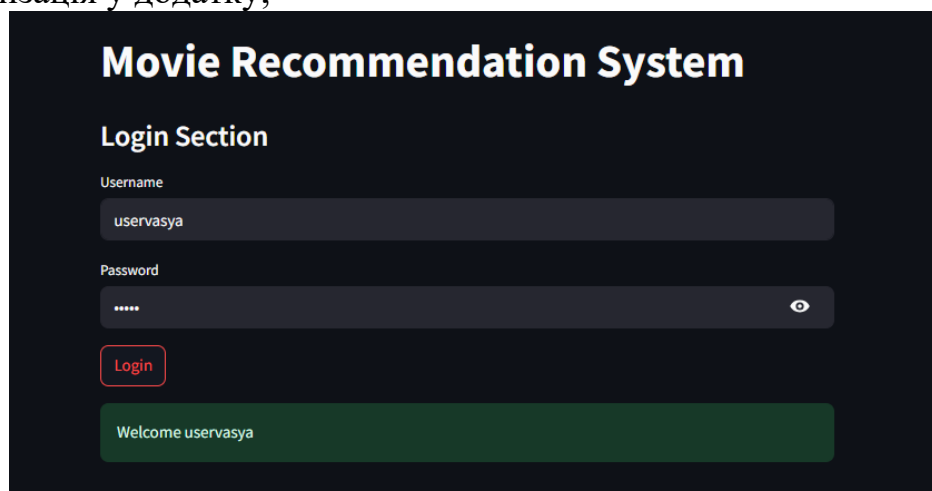


Рис. В.2 Тест 2

Тест 3

- 1.Перевірка виконання програми
2. Рекомендації від застосунку на основі уподобань;
3. Отримання результату.

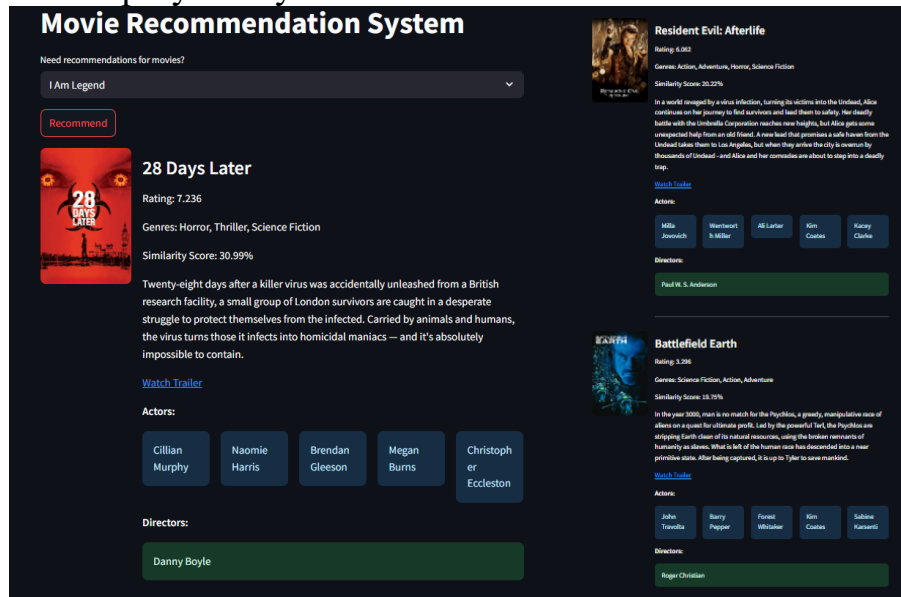


Рис. В.3 Тест 3

Тест вважається пройденим, якщо відбуваються вказані операції і їх відображення у програмному продукті.

Висновки: тест 1 успішно пройшов випробування, тест 2 успішно пройшов випробування і тест 3 успішно пройшов випробування. Випробування пройшло успішно.

Виконавець: студент групи КІ-41, Сухина В. В.