

Вінницький національний технічний університет

(повне найменування вищого навчального закладу)

Факультет інтелектуальних інформаційних технологій та автоматизації

(повне найменування інституту, назва факультету (відділення))

Кафедра комп'ютерних наук

(повна назва кафедри (предметної, циклової комісії))

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

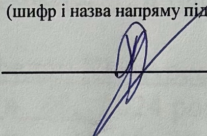
на тему:

«Інформаційна технологія щодо рекомендації

фільмів глядачам»

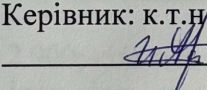
Виконав: студент 2-го курсу, групи ЗКН-23м
спеціальності 122 «Комп'ютерні науки»

(шифр і назва напрямку підготовки, спеціальності)

 Катеринчук В. С.

(прізвище та ініціали)

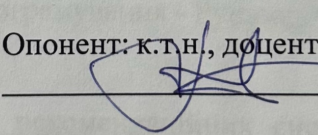
Керівник: к.т.н., доцент каф. КН

 Арсенюк І. Р.

(прізвище та ініціали)

« 05 » 12 2024 р.

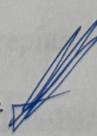
Опонент: к.т.н., доцент каф. АІТ

 Маслій Р. В.

(прізвище та ініціали)

« 05 » 12 2024 р.

Допущено до захисту
Завідувач кафедри КН
д.т.н., проф. Яровий А. А.

(прізвище та ініціали) 

« 06 » 12 2024 р.

Вінниця ВНТУ - 2024 рік

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра комп'ютерних наук
Рівень вищої освіти II-й (магістерський)
Галузь знань – 12 «Інформаційні технології»
Спеціальність – 122 «Комп'ютерні науки»
Освітньо-професійна програма – «Системи штучного інтелекту»

ЗАТВЕРДЖУЮ
Завідувач кафедри КН
д.т.н., проф. Яровий А.А.
11.10. 2024 року

ЗАВДАННЯ

НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Катеринчуку Вадиму Сергійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Інформаційна технологія щодо рекомендування фільмів глядачам

керівник роботи: к.т.н., доцент кафедри КН Арсенюк І. Р.
затверджені наказом ВНТУ від «17» 09. 2024 року № 310

2. Строк подання студентом роботи 11.11.2024 р.

3. Вихідні дані до роботи:

Кількість фільмів у базі - не менше 2 000; кількість параметрів рейтингування не менше 3; формат обкладинок - .jpeg, .jpg, .png, алгоритм рекомендування - на основі історії переглядів та оцінок; формат вхідних даних - .csv або .json (режисер, жанр, рейтинг), мова програмування - Python;

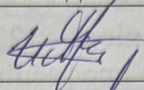
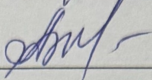
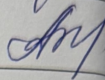
4. Зміст текстової частини:

Вступ, огляд та аналіз сучасних рекомендаційних систем, моделювання інформаційної технології щодо рекомендування фільмів глядачам, програмна реалізація інформаційної технології щодо рекомендування фільмів, економічна частина, висновки, список використаних джерел, додатки

5. Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень)

Модель дерева рішень щодо рекомендування фільмів; схема загального алгоритму функціонування системи; загальний алгоритм роботи інформаційної технології щодо рекомендування фільмів глядачам; UML-діаграма діяльності модулю фільтрації на основі вмісту інформаційної технології; приклад роботи програми.

Консультанти розділів роботи

Розділ	Прізвище, ініціалита посада консультанта	Підпис, дата	
		завдання видав	викона прийм
1-3	Арсенюк І.Р., к.т.н., доц. каф. КН		
4	Адлер О.О., к.т.н., . доц. ЕПВМ		

1 Дата видачі завдання 02.09.2024

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Строк виконання етапів роботи	При
1	Аналіз інформаційної технології аналізу щодо рекомендування фільмів глядачам	02.09.24 – 14.09.24	
2	Огляд існуючих систем-аналогів	15.09.24 – 28.09. 24	
3	Обґрунтування вибору програмних засобів для реалізації інформаційної технології	09.09.24 – 05.10.24	
4	Розробка структурної схеми інформаційної технології аналізу та рекомендування фільмів глядачам	06.10.24 – 16.10.24	
5	Економічна частина	17.10.24 – 24.10.24	
6	Апробація результатів дослідження	25.10.24 – 01.11.24	
7	Оформлення пояснювальної записки, графічного матеріалу, презентації	02.11.24 – 11.11.24	

Студент

(підпис)

Катеринчук В. С.

Керівник роботи

(підпис)

Арсенюк І. Р.

АНОТАЦІЯ

УДК 004.8

Катеринчук В. С. Інформаційна технологія щодо рекомендування фільмів глядачам.

Магістерська кваліфікаційна робота зі спеціальності 122 «Комп'ютерні науки», освітня програма «Системи штучного інтелекту». Вінниця: ВНТУ, 2024. 96 с.

Укр. мовою. Бібліогр.: 43 назв; рис.: 24; табл.: 9.

Робота присвячена розробці інформаційної технології щодо рекомендування фільмів глядачам. Проведено аналіз сучасних підходів і алгоритмів, які використовуються у рекомендаційних системах, таких як фільтрація на основі вмісту, спільна фільтрація та гібридний підхід. Виявлено переваги та недоліки цих підходів, що дозволило обґрунтувати необхідність створення інформаційної технології з розширеним функціоналом, яка базується на гібридному підході. У ході дослідження виконано постановку задачі.

Розроблено математичну модель рекомендаційної системи, сформовано її структурну схему та алгоритм функціонування. Проаналізовано та обрано програмне забезпечення, мову програмування та середовище розробки для реалізації інформаційної технології. Описано процес розробки та тестування системи.

Додатково реалізовано функціонал вибору рекомендованих фільмів за жанрами, акторами чи іншими критеріями. Графічна частина складається з 9 плакатів, які відображають результати моделювання та функціонування системи.

В економічній частині розраховано витрати на розробку 397927 грн та визначено період окупності – 0,56 року.

Ключові слова: рекомендаційна система, гібридний підхід, персоналізація, рекомендація фільмів, штучний інтелект

ABSTRACT

Katerynychuk V. S. Information Technology for Recommending Movies to Viewers.

Master's qualification thesis in the specialty 122 "Computer Science," educational program "Artificial Intelligence Systems." Vinnytsia: VNTU, 2024. 96 p.

In English. Bibliogr: 43 titles; figures: 24; tables: 9.

The work is dedicated to the development of an information technology for recommending movies to viewers. An analysis of modern approaches and algorithms used in recommendation systems, such as content – based filtering, collaborative filtering, and hybrid methods, has been conducted. The advantages and disadvantages of these approaches were identified, justifying the need to create an information technology with extended functionality based on a hybrid approach. The study included the formulation of the problem.

A mathematical model of the recommendation system was developed, along with its structural diagram and functional algorithm. The software, programming language, and development environment for implementing the information technology were analyzed and selected. The process of system development and testing is described.

Additionally, functionality was implemented to allow the selection of recommended movies based on genres, actors, or other criteria. The graphical section consists of eight posters illustrating the modeling results and system performance.

In the economic section, development costs were calculated at 397927 UAH, and the payback period was determined to be 0,56 years.

Keywords: recommendation system, hybrid approach, personalization, movie recommendations, artificial intelligence.

ЗМІСТ

ВСТУП	4
1 ОГЛЯД ТА АНАЛІЗ СУЧАСНИХ РЕКОМЕНДАЦІЙНИХ СИСТЕМ	9
1.1 Поняття рекомендаційних систем	9
1.2 Обґрунтування використання методів штучного інтелекту щодо рекомендування фільмів глядачам	10
1.3 Дослідження рекомендаційних систем на основі колаборативної фільтрації	13
1.4 Аналіз рекомендаційних систем на контентної фільтрації	20
1.5 Аналіз рекомендування систем на основі знань	22
1.6 Огляд існуючих рекомендаційних систем	24
1.7 Постановка задачі дослідження	30
1.8 Висновок до розділу 1	32
2 МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ЩОДО РЕКОМЕНДУВАННЯ ФІЛЬМІВ ГЛЯДАЧАМ	33
2.1 Розробка методу розв’язання задачі рекомендування фільмів	33
2.2 Обґрунтування вибору базового методу щодо рекомендування фільмів	34
2.3 Аналіз сценаріїв роботи інформаційної технології	36
2.4 Розробка математичної моделі надання рекомендацій щодо підбору фільмів	39
2.5 Розробка алгоритму функціонування інформаційної технології щодо рекомендування фільмів	43
2.6 Розробка структури інформаційної технології щодо рекомендування фільмів глядачам	46
2.7 Розробка алгоритму інформаційної технології щодо рекомендування фільмів глядачам	49
2.8 Висновок до розділу 2	51

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ЩОДО РЕКОМЕНДУВАННЯ ФІЛЬМІВ ГЛЯДАЧАМ	53
3.1 Опис мови програмування	53
3.2 Аналіз датасету «The Movies Dataset».....	54
3.3 Реалізація модулю колаборативної фільтрації.....	64
3.4 Реалізація модулю контентної фільтрації	66
3.5 Тестування програмного забезпечення щодо рекомендування фільмів глядачам	70
3.6 Висновок до розділу 3	75
4 ЕКОНОМІЧНА ЧАСТИНА.....	77
4.1 Проведення комерційного та технологічного аудиту інформаційної технології щодо рекомендування фільмів глядачам	77
4.2 Розрахунок витрат на здійснення науково-дослідної роботи	80
4.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором.....	89
4.4 Висновок до розділу 4.....	92
ВИСНОВКИ	94
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	96
Додаток А (обов'язковий) Протокол перевірки кваліфікаційної роботи на наявність текстових запозичень	99
Додаток Б (обов'язковий) Лістинг програми.....	100
Додаток В (обов'язковий) Ілюстративна частина	108
Додаток Г (довідниковий) Інструкція користувача	114

ВСТУП

Актуальність теми дослідження. На сьогоднішній день індустрія кінематографії активно інтегрує інформаційні технології для покращення взаємодії з глядачами та оптимізації процесу рекомендацій фільмів. Рекомендаційні системи, які допомагають користувачам знаходити фільми відповідно до їхніх уподобань, стають важливою складовою інтернет – платформ для перегляду контенту. Вони забезпечують персоналізацію рекомендацій і дозволяють значно зменшити час, витрачений на пошук відповідного контенту.

Зростаюча кількість користувачів онлайн – кінотеатрів та платформ для стримінгу фільмів створює величезний обсяг даних, який необхідно ефективно обробляти для надання точних рекомендацій. Однак, багато існуючих систем рекомендацій мають обмежену ефективність, через що користувачі часто отримують менш релевантні пропозиції або стикаються з проблемами перегляду контенту без урахування їхніх індивідуальних смаків.

Інформаційні технології, зокрема методи штучного інтелекту та машинного навчання, набирають популярності в процесі рекомендаційного аналізу. Вони дозволяють створювати більш точні і персоналізовані рекомендації, враховуючи не тільки історію переглядів, але й інші важливі фактори, такі як оцінки інших користувачів, популярність фільмів, жанрові вподобання, тощо.

Сучасні підходи до рекомендацій базуються на алгоритмах колаборативної фільтрації, контентній фільтрації або гібридних методах. Кожен із цих підходів має свої переваги і недоліки, але їх поєднання дозволяє досягати високої точності та ефективності рекомендацій. Однак на даний момент існують обмеження щодо їхньої масштабованості, адаптивності до змін в уподобаннях користувачів та обробки великих обсягів даних у реальному часі.

Отже, необхідність в удосконаленні існуючих систем рекомендацій для фільмів є актуальною, оскільки вона дозволить значно покращити користувацький досвід та допомогти глядачам знайти цікаві для них фільми з мінімальними затратами часу. Важливою є також розробка нових методів, що базуються на аналізі даних великих обсягів і можуть забезпечити ефективні рекомендації в умовах різноманіття контенту.

Крім того, інтеграція технологій штучного інтелекту в процеси рекомендацій дозволить створити гнучкі та масштабовані рішення, які будуть знижувати вплив людського фактору і забезпечувати більш об'єктивне прогнозування уподобань користувачів.

Таким чином, розробка інформаційної технології щодо рекомендування фільмів глядачам є актуальною.

Зв'язок роботи з науковими програмами, планами, темами. Магістерська робота виконана відповідно до напрямку наукових досліджень кафедри комп'ютерних наук Вінницького національного технічного університету НДР: «22 К1. Розробка прикладних інтелектуальних інформаційних технологій та систем» та плану наукової та навчально-методичної роботи кафедри.

Мета та завдання дослідження. Метою дослідження магістерської кваліфікаційної роботи є розширення функціональних можливостей інформаційної технології для рекомендації фільмів глядачам.

Для досягнення поставленої мети необхідно виконати такі завдання:

- здійснити аналіз існуючих підходів до створення систем рекомендацій у сфері кінематографу;
- дослідити сучасні алгоритми машинного навчання та обрати той, що найкраще підходить для персоналізації рекомендацій на основі вподобань користувачів;
- розробити математичну модель інформаційної системи рекомендації фільмів та адаптувати її під обрану мету;

- визначити основні етапи роботи інформаційної технології та на їх основі створити структуру й алгоритм роботи програмного забезпечення;
- реалізувати програмну систему для надання рекомендацій фільмів;
- провести тестування створеної системи та виконати аналіз отриманих результатів.

Об'єктом дослідження: є процес рекомендування фільмів глядачам.

Предмет дослідження – це програмні засоби для рекомендування фільмів глядачам та їх функціональні можливості.

Методи дослідження. У роботі використано такі методи наукових досліджень: метод системного аналізу для вивчення структури інформаційної технології рекомендації фільмів; алгоритми машинного навчання для персоналізації рекомендацій на основі вподобань користувачів; методи математичної статистики для оцінки точності та ефективності запропонованої системи рекомендацій; методи об'єктно-орієнтованого програмування для розробки та впровадження програмного забезпечення, що автоматизує процес надання рекомендацій.

Наукова новизна одержаних результатів. Удосконалено інформаційну технологію рекомендування фільмів, яка відрізняється використанням комбінованого підходу, що поєднує методи колаборативної фільтрації та контентної фільтрації, з урахуванням багатфакторного аналізу, включаючи жанри, акторський склад та історію вподобань користувачів. Це дозволяє розширити функціонал формування рекомендацій та скоротити час на їх створення.

Система відрізняється застосуванням елементної бази, яка дозволяє інтегрувати дані з різних джерел, таких як соціальні мережі та потокові платформи. Це сприяє збільшенню кількості релевантних рекомендацій та оптимізації роботи платформи для різних груп користувачів.

Враховано нові критерії аналізу даних, які включають не лише основні характеристики фільмів, але й специфічні запити користувачів, що дозволяє

створювати персоналізовані рекомендації для кожного глядача. Такий підхід забезпечує покращення користувацького досвіду та підвищення ефективності функціонування системи.

Практичне значення одержаних результатів полягає у такому:

1. Розроблено новий підхід до рекомендування фільмів, який використовує аналіз уподобань глядачів та попередній вибір;
2. Розроблено алгоритм навчання рекомендаційної системи на основі колаборативної фільтрації для розширення функціональних можливостей
3. Розроблено алгоритм функціонування рекомендаційної системи для поліпшення релевантності та своєчасності рекомендацій;
4. Розроблено програмний засіб щодо рекомендування фільмів глядачам на основі вподобань користувачів, який може бути інтегрований у стрімінгові платформи або онлайн-кінотеатри.

Розроблені алгоритми можуть бути впроваджені в навчальний процес як лекції на тему «Рекомендаційні системи для персоналізованого рекомендування контенту» в рамках дисципліни «Інформаційні технології та штучний інтелект».

Достовірність теоретичних положень магістерської кваліфікаційної роботи підтверджується точністю формулювання задач, правильним використанням методів машинного навчання при реалізації алгоритмів рекомендацій, коректністю аналітичних обчислень, а також порівнянням отриманих результатів з існуючими підходами. Крім того, результати моделювання були перевірені на збіжність із фактичними даними під час тестування програмного забезпечення для рекомендування фільмів, що підтвердило ефективність запропонованої інформаційної технології.

Особистий внесок магістранта. Усі результати, наведені у магістерській кваліфікаційній роботі, отримані самостійно. У роботах, опублікованих у співавторстві, автору належать такі результати: [1] –

розробка моделі вибору фільмів на основі гібридного підходу; [2] – проектування та розробка узагальненого алгоритму інформаційної технології щодо рекомендації фільмів глядачам.

Апробація результатів роботи. Результати роботи були представлені та обговорені на 16th International Scientific and Practical Conference «Scientific Horizon in the Context of Social Crises» (November 26-28, 2024; Tokyo, Japan) [1]

Публікації. За результатами дослідження опубліковано: 1 тези [1] та 1 стаття у збірнику матеріалів міжнародної науково-технічної конференції [2], а також подано до реєстрації в Український національний офіс інтелектуальної власності та інновацій свідоцтво про реєстрацію авторського права на твір (комп'ютерна програма), реєстраційний номер Вх-44681/2024 від 07.11.2024.

1 ОГЛЯД ТА АНАЛІЗ СУЧАСНИХ РЕКОМЕНДАЦІЙНИХ СИСТЕМ

1.1 Поняття рекомендаційних систем

Системи рекомендацій є важливою частиною інформаційних технологій, які спрямовані на фільтрацію та підбір інформації, адаптованої до потреб користувачів. Ці системи розроблені з метою передбачити, які об'єкти (у випадку нашої теми – фільми) будуть найбільш цікавими для конкретного глядача. В основі їхньої роботи лежить аналіз різних типів даних, таких як історія переглядів, оцінки, вподобання або профіль користувача. За допомогою інтерактивних інтерфейсів системи рекомендацій намагаються взаємодіяти з користувачем, створюючи персоналізований досвід, а не просто надаючи статичну інформацію.

Такі системи широко використовуються у багатьох сферах. Наприклад, у відеосервісах (Netflix, Megogo, IMDb, YouTube), музичних платформах (Spotify), інтернет-магазинах (Amazon, eBay) і навіть у соціальних мережах (Facebook, Twitter). Вони допомагають оптимізувати пошук контенту або товарів, економлячи час і зусилля користувачів. Для стрімінгових платформ, як-от Netflix, ці системи є основою для залучення та утримання аудиторії, забезпечуючи персоналізовані рекомендації, які підвищують задоволеність користувачів [3].

Застосування рекомендаційних систем є особливо корисним на платформах із великим обсягом контенту. Наприклад, у відеосервісах, де бібліотека може містити тисячі фільмів і серіалів, ручний пошук релевантного контенту був би складним завданням. Системи рекомендацій використовують алгоритми для аналізу інтересів глядачів, допомагаючи їм швидше знаходити те, що відповідає їхнім вподобанням. Для цього часто застосовуються методи колаборативної фільтрації, фільтрації на основі вмісту або їхні гібридні комбінації.

Рекомендаційні системи не лише покращують користувацький досвід, але й є потужним інструментом для монетизації платформ. Їхня здатність адаптуватися до індивідуальних потреб робить їх незамінним елементом сучасних інформаційних технологій. У випадку з рекомендаціями фільмів, такі системи дозволяють глядачам швидко знаходити контент, що відповідає їхнім інтересам, забезпечуючи зручність та ефективність використання платформи.

Більше того, аналогічні технології можна легко знайти на різних соціальних мережах, порталах, присвячених літературі, подорожам, новинним ресурсам і багатьох інших. Фактично, цей метод дуже популярний.

У сучасний період існують три основні та широко використовувані стратегії створення систем рекомендацій [4]:

Контентна фільтрація (content – based filtering).

Колаборативна фільтрація (collaborative filtering).

Гібридний підхід, що комбінує в собі елементи обох попередніх методів.

1.2 Обґрунтування використання методів штучного інтелекту щодо рекомендування фільмів глядачам

Сучасні системи рекомендацій у кіноіндустрії активно використовують можливості штучного інтелекту (ШІ) для забезпечення персоналізованого підходу до кожного користувача. ШІ дозволяє аналізувати величезні обсяги даних, враховувати індивідуальні вподобання глядачів і пропонувати найбільш релевантний контент. Завдяки цьому технології ШІ стали важливим компонентом у роботі стрімінгових сервісів, таких як Netflix, Amazon, YouTube та інших [5].

Роль ІІІ у функціонуванні систем рекомендацій

Штучний інтелект забезпечує автоматизацію процесів аналізу даних, використовуючи алгоритми, які дозволяють швидко знаходити закономірності у вподобаннях користувачів. Основними завданнями ІІІ у системах рекомендацій є:

- Персоналізація вибору. ІІІ обробляє історію переглядів, оцінки фільмів, жанрові вподобання та інші параметри, створюючи профіль користувача. Це дозволяє пропонувати фільми, які найбільш відповідають інтересам конкретного глядача.
- Обробка великих обсягів даних. ІІІ здатний аналізувати мільйони фільмів, серіалів і відгуків, забезпечуючи швидке генерування рекомендацій навіть для нових користувачів без великої історії переглядів.
- Адаптація до змін. Системи на основі ІІІ можуть враховувати зміни у вподобаннях користувачів, адаптуючи свої рекомендації залежно від нових переглядів чи оцінок.

Використання технологій ІІІ для покращення рекомендацій

Однією з найважливіших функцій ІІІ є аналіз взаємодії користувачів із платформою. Це включає вивчення:

1. Поведінки глядача. Наприклад, якщо користувач переглядає трейлери певного жанру, але не обирає для перегляду повнометражні фільми, система може запропонувати фільми з коротшою тривалістю.
2. Показників популярності. ІІІ аналізує, які фільми чи серіали є найбільш популярними серед користувачів із подібними смаками, і рекомендує їх.
3. Відгуків і рейтингів. Використовуючи дані з баз, таких як IMDb, система може виділити контент із високими оцінками, який відповідає уподобанням користувача.

Переваги використання ІІІ у кіноіндустрії

Застосування штучного інтелекту у системах рекомендацій має низку переваг:

- Ефективність пошуку. ШІ скорочує час вибору контенту, пропонуючи варіанти, що відповідають інтересам користувача, без необхідності тривалого перегляду каталогів.
- Підвищення рівня задоволеності. Персоналізовані рекомендації забезпечують позитивний досвід для користувачів, підвищуючи їхню лояльність до платформи.
- Аналіз нових тенденцій. ШІ дозволяє виявляти тренди в уподобаннях користувачів, допомагаючи платформам адаптувати свій контент до актуальних запитів.

Виклики та обмеження використання ШІ

Попри значний потенціал, використання ШІ у рекомендаційних системах має свої складнощі. Зокрема, необхідно враховувати:

- Залежність від даних. Якість рекомендацій значною мірою залежить від обсягу та точності вхідних даних. Якщо історія переглядів користувача недостатньо велика, система може генерувати менш точні рекомендації.
- Проблема конфіденційності. Для ефективної роботи системи потрібна велика кількість персональних даних, що може викликати занепокоєння у користувачів щодо їхньої приватності.

Застосування ШІ у системах рекомендацій фільмів [6]

ШІ допомагає користувачам знайти контент, який відповідає їхнім інтересам, навіть якщо вони самі не можуть чітко визначити свої вподобання. У стрімінгових сервісах використовуються моделі, що дозволяють аналізувати поведінку глядачів, наприклад, час перегляду, вибір жанрів і оцінки. Це забезпечує користувачам доступ до контенту, який вони, можливо, не обрали б самостійно, але який відповідає їхнім очікуванням.

Штучний інтелект є невід'ємною частиною сучасних систем рекомендацій фільмів, допомагаючи вирішувати проблему вибору серед великої кількості контенту. Завдяки таким технологіям глядачі отримують не лише

персоналізовані рекомендації, але й кращий досвід взаємодії зі стрімінговими сервісами.

1.3 Дослідження рекомендаційних систем на основі колаборативної фільтрації

Колаборативна фільтрація [7] є одним із найпоширеніших підходів у сучасних рекомендаційних системах. Її основа полягає на аналізі даних про вподобання користувачів, що дозволяє формувати персоналізовані рекомендації. Ідея цього підходу полягає в припущенні, що користувачі, які раніше оцінювали схожі об'єкти однаково, ймовірно, матимуть подібні оцінки й у майбутньому. Цей принцип активно використовується для рекомендацій фільмів або серіалів на стрімінгових платформах, де система пропонує нові об'єкти на основі попередніх переглядів і оцінок.

Процес колаборативної фільтрації можна поділити на три основні етапи:

1. Збір даних про взаємодії користувачів з контентом. На першому етапі формується база даних, яка містить інформацію про оцінки, історію переглядів, лайки та інші взаємодії користувачів з фільмами чи серіалами. Ця інформація стає основою для подальшого аналізу схожості між користувачами або об'єктами. Наприклад, якщо користувач переглянув кілька фільмів і поставив їм високі оцінки, система зробить висновок про його інтерес до конкретного жанру.
2. Побудова матриці зв'язків між користувачами або об'єктами. Далі зібрані дані перетворюються у вигляді матриць, зокрема "Матриця 'елемент-користувач'". У такій матриці кожен стовпець представляє об'єкт (наприклад, фільм), а кожен рядок – користувача. У клітинках цієї матриці містяться оцінки або взаємодії користувача з певним елементом. Також існує матриця "користувач – елемент", де кожен рядок представляє користувача, а кожен стовпець – конкретний елемент. Вона дозволяє

обчислити схожість між користувачами, визначаючи, наскільки подібними є їхні вподобання.

3. Генерація рекомендацій. Після побудови матриці система використовує інформацію для визначення того, які об'єкти будуть цікаві конкретному користувачеві. Якщо інші користувачі, що мають схожі оцінки, оцінили фільм, який активний користувач ще не переглядав, система пропонує цей фільм у рекомендаціях. В основі такого процесу лежить визначення схожості між користувачами або елементами, що дозволяє персоналізувати рекомендації.

Колаборативну фільтрацію можна реалізувати через два основні підходи: "користувач – користувач" і "елемент – елемент".

- Спільна фільтрація "користувач – користувач" зосереджується на пошуку користувачів із подібними вподобаннями. Наприклад, якщо два користувачі мають схожі оцінки для кількох фільмів, система пропонує інші фільми, які сподобались одному з цих користувачів. Таким чином, для кожного активного користувача вибираються найбільш схожі "сусіди", а потім формуються рекомендації на основі вподобань цих користувачів.
- Спільна фільтрація "елемент – елемент" акцентує увагу на об'єктах. У цьому підході система шукає схожість між фільмами чи іншими елементами.
- Якщо два фільми отримують схожі оцінки від великої кількості користувачів, система вважає їх подібними. Відтак, якщо користувач оцінив один з них, йому рекомендується інший. Це дозволяє рекомендувати об'єкти, що часто оцінюються разом.

Додатково до цього, для покращення точності рекомендацій використовуються методи машинного навчання, такі як матричне розкладання. Це дозволяє зменшити розмірність матриці "елемент – користувач", виділяючи приховані фактори, які впливають на уподобання користувачів, і допомагає знаходити менш очевидні зв'язки між користувачами та об'єктами. Також

використовується кластеризація для групування користувачів із подібними вподобаннями, що дає змогу формувати рекомендації на основі загальних інтересів групи.

Поєднання методів "користувач – користувач" і "елемент – елемент" дозволяє досягти більш високої точності в рекомендаціях. Це дає змогу враховувати як популярність об'єкта серед користувачів, так і схожість між ними, що особливо ефективно для великих платформ з різноманітною аудиторією.

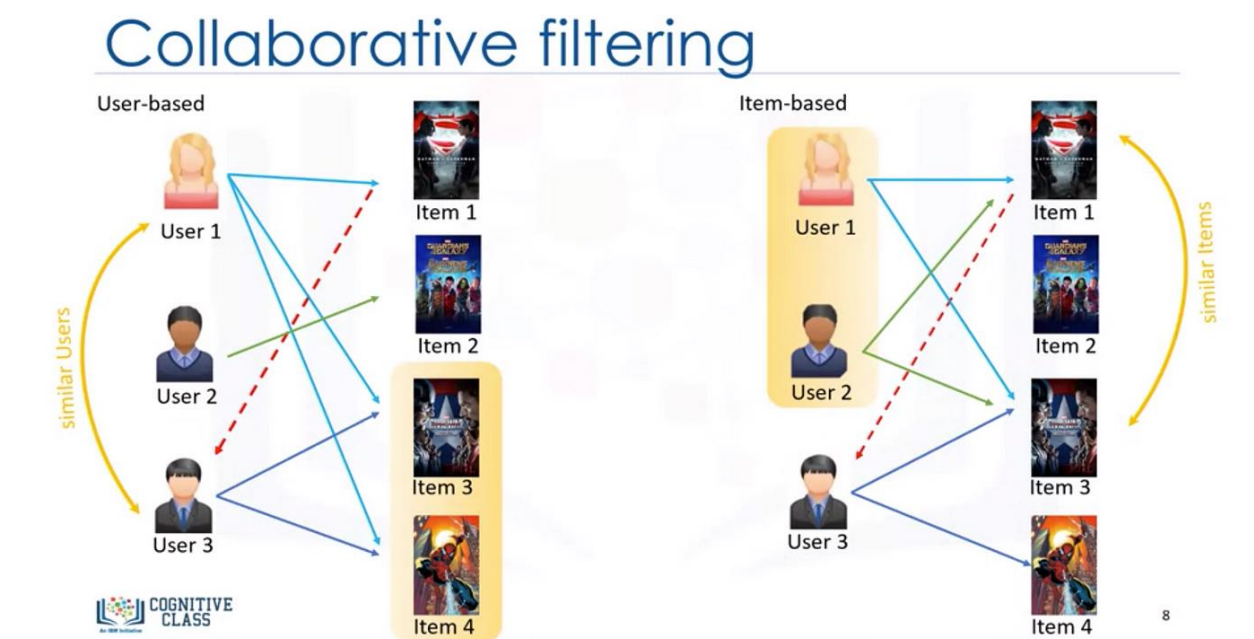


Рисунок 1.1 – Колаборативна фільтрація «користувач – користувач»

У колаборативній фільтрації використовується матриця, де кожен рядок відповідає користувачеві, а кожен стовпець – елементу, таким як фільм. Якщо користувач ще не оцінив певний елемент, то відповідна клітинка матриці залишається порожньою. У процесі обчислення рекомендацій система аналізує вподобання користувачів і виявляє схожі патерни у їхніх оцінках. Для цього необхідно враховувати два фактори, користувачів з однаковим балом на одному наборі однакових елементів.

Для того, щоб сформувати рекомендації, спочатку обчислюється міра схожості між користувачами. Це зроблено за допомогою різних методів, таких

як косинусна подібність або коефіцієнт кореляції Пірсона, які вимірюють, наскільки схожі профілі користувачів на основі оцінок для однакових об'єктів. Якщо два користувачі оцінили фільми однаково, система припускає, що вони мають схожі вподобання.

Після визначення схожості між користувачами, система формує групи користувачів з подібними інтересами, так званих "сусідів". Для кожного активного користувача система вибирає об'єкти, які популярні серед цих сусідів, але ще не були оцінені активним користувачем. Цей підхід дозволяє сформувати рекомендації, що базуються на вподобаннях подібних користувачів.

Колаборативна фільтрація особливо корисна у випадках, коли система не має достатньо даних про конкретний об'єкт (наприклад, новий фільм). У таких ситуаціях рекомендації базуються на колективному досвіді інших користувачів із подібними смаками, що дозволяє заповнити прогалини в інформації. Це робить підхід універсальним і ефективним навіть для платформ із великим обсягом контенту та різноманітними інтересами аудиторії.

							
	4				5		Historical data
	4	5		5			
		5		4		3	
 Lizzy	5	4	?	?	?	?	New user

Рисунок 1.2 – Матриця «елемент – користувач

Метод "елемент – елемент" є одним з популярних підходів у побудові рекомендаційних систем, який базується на аналізі схожості між об'єктами, що взаємодіють з користувачами. У цьому підході важливо зрозуміти, як користувачі оцінюють або взаємодіють із різними елементами, такими як фільми, товари чи серіали. Оскільки взаємодія користувачів з певними елементами має величезне значення для системи, елементи, які мають подібні оцінки або схожі інші характеристики, будуть класифіковані як схожі.

Цей метод дає змогу створювати рекомендації, які ґрунтуються на груповій поведінці користувачів, а не на індивідуальних вподобаннях. Наприклад, якщо певний фільм або серіал набрав високі оцінки від багатьох користувачів, система може порекомендувати його іншим користувачам, які ще не переглядали його, але схильні до перегляду схожого контенту. Таким чином, алгоритм зменшує залежність від індивідуальних оцінок, що інколи дозволяє уникнути сильних перекосів, пов'язаних з обмеженим обсягом даних для певного користувача.

Проте метод "елемент – елемент" має певні обмеження у порівнянні з підходом "користувач – користувач". Оскільки він менш орієнтований на індивідуальні вподобання, його результати можуть бути менш персоналізованими. Це означає, що рекомендовані фільми чи серіали можуть бути менш точними для користувачів із дуже специфічними вподобаннями, що може привести до менш ефективних рекомендацій для певної аудиторії. Тим не менш, у великих платформах, де наявність даних про кожного окремого користувача є обмеженою, метод "елемент – елемент" показує вищу надійність, оскільки не потребує глибокого аналізу індивідуальних смаків і переваг.

Проблеми колаборативної фільтрації та їх вирішення [8]

1. Розрідженість матриці "елемент – користувач"

Одна з основних проблем цього методу полягає в тому, що оцінки фільмів або інших елементів часто надаються лише обмеженою кількістю користувачів. У комерційних рекомендаційних системах може бути мільйони товарів, але лише невелика частина користувачів оцінила кожен

з них. Це призводить до сильно розрідженої матриці "об'єкт–користувач", що ускладнює точність рекомендацій через відсутність достатньої кількості даних для порівняння. У таких випадках часто використовуються алгоритми, що дозволяють «доповнювати» порожні клітинки матриці за допомогою інших моделей, зокрема через комбінування підходів, таких як спільна фільтрація і контентна фільтрація.

2. Масштабованість

Масштабованість є ще однією проблемою, особливо на великих платформах із мільйонами користувачів та товарами. Зі збільшенням кількості користувачів, кількість взаємодій, яку потрібно обробити, також збільшується, що ускладнює обчислення подібності між елементами та накладає додаткові вимоги до обчислювальних ресурсів.

3. Проблема "холодного старту"

Проблема "холодного старту" є однією з найбільших складнощів для рекомендаційних систем, особливо коли з'являються нові елементи (наприклад, нові фільми чи продукти) або нові користувачі, які не мають достатньої кількості оцінок у системі. Без цих даних система не може точно передбачити, які фільми чи серіали будуть подобатися новим користувачам або які нові елементи варто порекомендувати. У такому випадку використовується комбінований підхід, що включає аналіз контенту.

Переваги:

- **Більш надійність:** Порівняно з підходом "користувач – користувач", метод "елемент – елемент" не залежить від індивідуальних вподобань, що дозволяє створювати більш стабільні рекомендації для нових користувачів.

- Широка застосовність: Застосовується на великих платформах із великою кількістю елементів, оскільки дозволяє ефективно використовувати масову інформацію про взаємодії.

Недоліки:

- Менша персоналізація: Рекомендації можуть бути менш точними для користувачів із дуже специфічними смаками, оскільки вони базуються на загальних оцінках і взаємодіях серед більшої групи користувачів.
- Проблема розрідженості даних: У разі наявності значної кількості неоцінених елементів для користувачів, точність рекомендацій знижується.

Метод "елемент – елемент" є важливою частиною рекомендаційних систем, особливо в контексті платформ з великим обсягом контенту та великими аудиторіями. Хоча він може бути менш персоналізованим порівняно з іншими підходами, такими як "користувач – користувач", він залишається надійним і ефективним для створення рекомендацій в реальних умовах, де важлива масштабованість та стабільність. На рисунку 1.3 зображена колаборативна фільтрація на основі елементу.

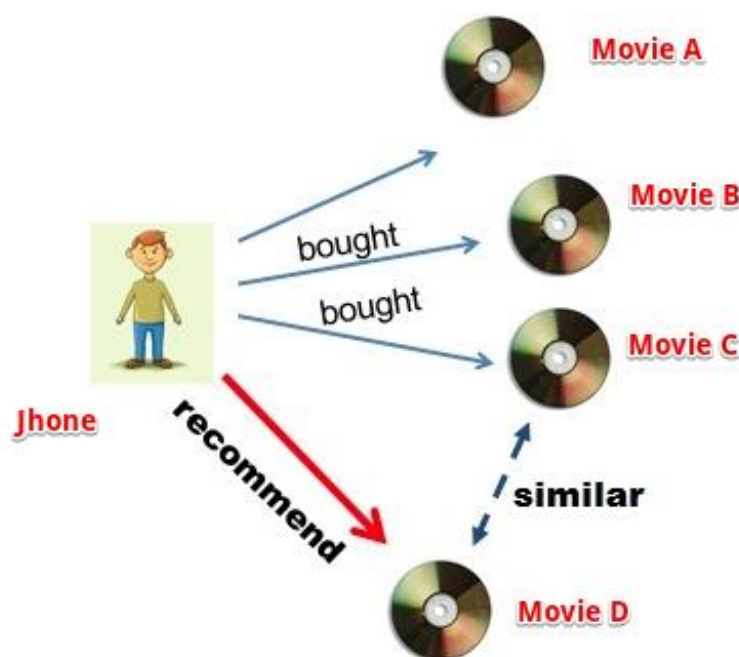


Рисунок 1.3 – Основа «елемент – елемент»

1.4 Аналіз рекомендаційних систем на основі контентної фільтрації

Метод контентної фільтрації є одним із основних підходів у створенні рекомендаційних систем, який орієнтується на аналіз характеристик об'єктів і формування профілю користувача. Цей метод дозволяє створювати персоналізовані рекомендації навіть за відсутності детальної інформації про самих користувачів.

У контентної фільтрації рекомендаційні системи створюють вибір для користувача, орієнтуючись на характеристики фільмів, серіалів чи інших елементів. Це можуть бути такі атрибути, як назва, жанр, опис, акторський склад, режисер, рік випуску та інші параметри, що характеризують об'єкт. Наприклад, якщо користувач часто переглядає науково – фантастичні фільми або фільми за участю певного актора, система буде пропонувати йому інші фільми з подібними характеристиками. Це дозволяє системі створити індивідуальний підхід до кожного користувача, не залежачи від оцінок інших людей [10,11].

Для коректної роботи методу фільтрації на основі вмісту необхідно виконати кілька основних кроків:

1. Створення профілю користувача.

Профіль користувача визначається на основі його вподобань і взаємодій з контентом. Це може бути історія переглядів, фільмів або жанри, до яких користувач виявляє інтерес. Наприклад, якщо користувач часто оцінює фільми в жанрі драми високими балами, система зробить припущення, що інші драми ймовірно зацікавлять його.

2. Формування профілю елемента.

Кожен фільм, серіал або інший об'єкт у системі має набір атрибутів, що характеризують його. Наприклад, для фільму це можуть бути жанр, актори, режисер, рік випуску, мова та інші характеристики. Це дозволяє системі порівнювати фільми один з одним і на основі цих характеристик здійснювати пошук схожих елементів для подальших рекомендацій.

3. Зіставлення профілю користувача з атрибутами елементів.

Після створення профілю користувача і профілю елементів система зіставляє ці дані, щоб зробити прогноз про те, які фільми або серіали можуть бути цікавими для користувача. Наприклад, якщо користувач активно переглядає фільми в жанрі наукової фантастики, система пропонує йому нові релізи з цього жанру або класичні фільми з подібним акторським складом або тематикою.

Зображення цього підходу можна побачити на рисунку 1.4.

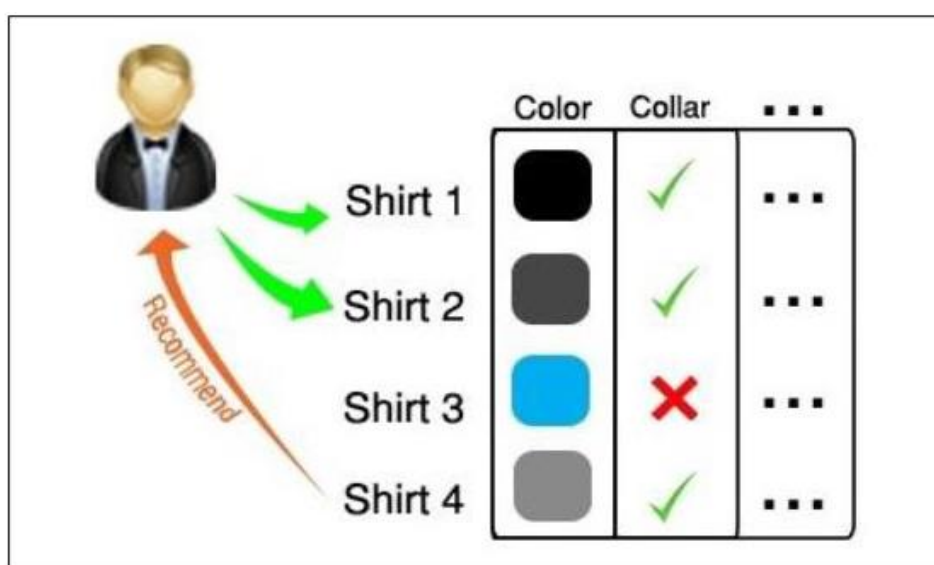


Рисунок 1.4 – Зображення підходу зі створенням профілю користувача і профілю елемента

Алгоритми таких систем орієнтовані на аналіз дій користувачів над контентом і використання отриманих даних для генерації пропозицій. Система аналізує попередні перегляди, рейтинги або обрані користувачем об'єкти й рекомендує схожі елементи.

Серед переваг методу можна виділити:

- Незалежність від взаємодії з іншими користувачами: система аналізує лише самі об'єкти й уподобання конкретного користувача.
- Прозорість рекомендацій: зрозумілі механізми роботи системи.

- Відсутність проблеми "холодного старту" для нових елементів, оскільки рекомендації можуть формуватися навіть за відсутності історії їхніх оцінок.

Однак цей підхід має і свої недоліки:

1. Аналіз контенту може бути недостатньо детальним для точного розрізнення схожих елементів, що іноді знижує якість рекомендацій.
2. Система схильна до надмірної спеціалізації, коли вона пропонує користувачу тільки ті елементи, які точно відповідають його попереднім уподобанням. Це може обмежити новизну пропозицій і призвести до зниження інтересу.
3. Для нових користувачів, про яких немає даних, система може працювати менш ефективно, поки не накопичиться необхідна інформація.

У підсумку, метод контентної фільтрації є потужним інструментом для створення персоналізованих рекомендацій. Однак для максимізації його ефективності доцільно поєднувати його з іншими підходами, наприклад колаборативною фільтрацією, щоб компенсувати недоліки та створити більш адаптивну систему.

1.5 Аналіз рекомендування систем на основі знань

Системи рекомендацій на основі знань [12] – це особливий тип технологій, який базується на чітких знаннях про предмети, які рекомендуються, а також на інформації про уподобання користувачів та правилах, які визначають, який саме товар або послуга повинні бути рекомендовані в конкретному контексті. Такі системи використовуються, коли стандартні методи, такі як колаборативна фільтрація чи фільтрація за змістом, не можуть бути застосовані ефективно.

Системи рекомендацій на основі знань особливо корисні в складних сферах, де продукти або послуги купуються не часто, наприклад, у випадку з фільмами, де клієнти мають специфічні вимоги. Вони дозволяють чітко визначити, який фільм буде найбільш релевантним для користувача, враховуючи

його вподобання, наприклад, жанр, акторський склад або режисера. У цьому контексті система може рекомендувати фільм на основі чітких характеристик, таких як "максимальний час перегляду", "перевага жанру" або "актори". Інші підходи, такі як колаборативна фільтрація або фільтрація за змістом, не можуть враховувати такі аспекти в потрібному обсязі.

Ще однією перевагою систем рекомендацій, заснованих на знаннях, є відсутність проблеми "холодного старту", оскільки ці системи використовують визначені правила для надання рекомендацій, навіть якщо користувач не має історії взаємодії з платформою.

Для створення таких систем використовуються кілька ключових елементів:

1. "Діалог" – експертні системи рекомендацій часто мають діалогову природу. Це означає, що уподобання користувача виявляються через зворотний зв'язок під час взаємодії з системою. Це особливо важливо в складних областях, де не всі вимоги можуть бути визначені одразу, а точні уподобання користувача можна зібрати тільки за допомогою процесу взаємодії. У системах рекомендацій на основі знань часто запитують додаткові деталі або параметри, щоб точніше підібрати рекомендації.
2. "Обмеження" – система може працювати на основі заданих користувачем обмежень, таких як вибір конкретного жанру, тривалості фільму чи акторського складу. Наприклад, користувач може вибрати "екшн" або "драму" як основний жанр, а система буде фільтрувати фільми саме за цими параметрами. Відповідні обмеження можуть бути введені під час пошуку, що дозволяє зосередити рекомендації на найбільш релевантних варіантах.
3. "Критика" – користувачі можуть надавати відгуки, які слугують як критичні зауваження, що вказують на необхідність змін у рекомендаціях. Наприклад, користувач може сказати: "Мені подобаються ці фільми, але хочу бачити більше комедій". Це дозволяє системі адаптувати свої

рекомендації для наступних пошуків, враховуючи змінені побажання. Критика допомагає уточнити вимоги користувача, покращуючи точність подальших рекомендацій.

Найбільшим недоліком таких систем є їх залежність від визначених асоціативних правил, які повинні бути побудовані експертами в галузі. Розробка таких правил потребує значних зусиль і ресурсів, оскільки кожна система є унікальною для своєї тематики або галузі. Крім того, ці системи зазвичай не можуть бути перенесені на інші предметні області без суттєвого перегляду або створення нових асоціативних правил, що збільшує їхню вартість і складність.

У підсумку, хоча системи рекомендацій на основі знань є надзвичайно корисними для глибокої персоналізації рекомендацій у складних сферах, вони потребують значних інвестицій у розробку та підтримку, що робить їх дорогими у порівнянні з іншими методами.

1.6 Огляд існуючих рекомендаційних систем

Сучасний ринок онлайн – платформ пропонує велику кількість сервісів, які спеціалізуються на рекомендаціях фільмів і часто одночасно виступають місцем для їх перегляду. Ці сервіси забезпечують користувачів базовими функціями, такими як підбір контенту на основі історії переглядів чи загальних вподобань. Проте багато з існуючих платформ мають свої недоліки, наприклад, обмежений функціонал, недостатня точність рекомендацій або неврахування індивідуальних особливостей глядачів [13].

У цьому контексті важливим завданням є розробка інформаційної технології, яка не лише відповідатиме сучасним вимогам до персоналізації, але й подолає недоліки, притаманні аналогам. Така технологія має забезпечувати розширення функціоналу, враховувати широкий спектр характеристик фільмів і вподобань користувачів, а також бути зручною у використанні.

Для кращого розуміння переваг і недоліків існуючих платформ проведемо порівняльний аналіз характеристик кількох популярних програм – аналогів. Результати такого аналізу наведені в таблиці 1.1.

Таблиця 1.1 – Порівняння характеристик існуючих програм – аналогів

Програмна система	Переваги	Недоліки
Netflix	Надає індивідуальні рекомендації, враховуючи історію переглядів фільмів і серіалів; Пропонує списки найкращих фільмів за різними критеріями, які формуються індивідуально для кожного користувача.	Кількість фільмів та серіалів обмежена; Не можна оцінити чи прокоментувати фільм
Megogo	Пропонує персоналізовані рекомендації на основі історії переглядів користувача; формує списки найкращих фільмів та серіалів,	Обмежена точність рекомендацій для нових користувачів; обмежений контент в деяких регіонах; складний інтерфейс для новачків.
IMDB	Детальна інформація про фільми та серіали, рецензії і рейтинги, персоналізовані рекомендації, списки найкращих фільмів, широкий вибір жанрів.	Залежність від рейтингової системи, яка може бути суб'єктивною, обмеження контенту в деяких регіонах, наявність реклами.
YouTube	Персоналізовані рекомендації, величезний вибір контенту, можливість взаємодії, підтримка високоякісного відео, інтеграція з Google.	Реклама, обмеження контенту в деяких регіонах, не завжди точні рекомендації.

Netflix [19] – це популярна платформа потокового відео, яка використовує передові технології штучного інтелекту та машинного навчання для формування персоналізованих рекомендацій фільмів і серіалів. Завдяки аналізу великої кількості даних про уподобання користувачів, платформа здатна забезпечувати точні рекомендації, що враховують як історію переглядів, так і індивідуальні інтереси кожного з понад 100

мільйонів підписників. Цей підхід дозволяє зменшити час на пошук релевантного контенту та підвищити задоволеність користувачів.

Одним із ключових напрямків роботи Netflix є створення персоналізованих трейлерів, що ґрунтуються на перевагах конкретного глядача. Наприклад, якщо користувач надає перевагу динамічним екшн – сценам або романтичним епізодам, алгоритм виділить ці моменти навіть у трейлерах до фільмів інших жанрів. Таким чином, кожен глядач отримує унікальний контент, який відповідає його уподобанням.

Інформаційна технологія, що аналізує вподобання користувачів, здатна враховувати жанрові уподобання, популярність фільмів, акторський склад та інші характеристики. Використання таких інструментів дозволяє підвищити точність рекомендацій, роблячи їх релевантними для різних категорій глядачів. Подібно до Netflix, система може також адаптувати рекомендації для нових користувачів, формуючи базові пропозиції на основі популярного контенту або початкових уподобань.

Застосування технологій штучного інтелекту у сфері рекомендаційних систем не лише спрощує пошук контенту, але й робить взаємодію з платформою більш комфортною, сприяючи залученню нових користувачів і збереженню існуючих.

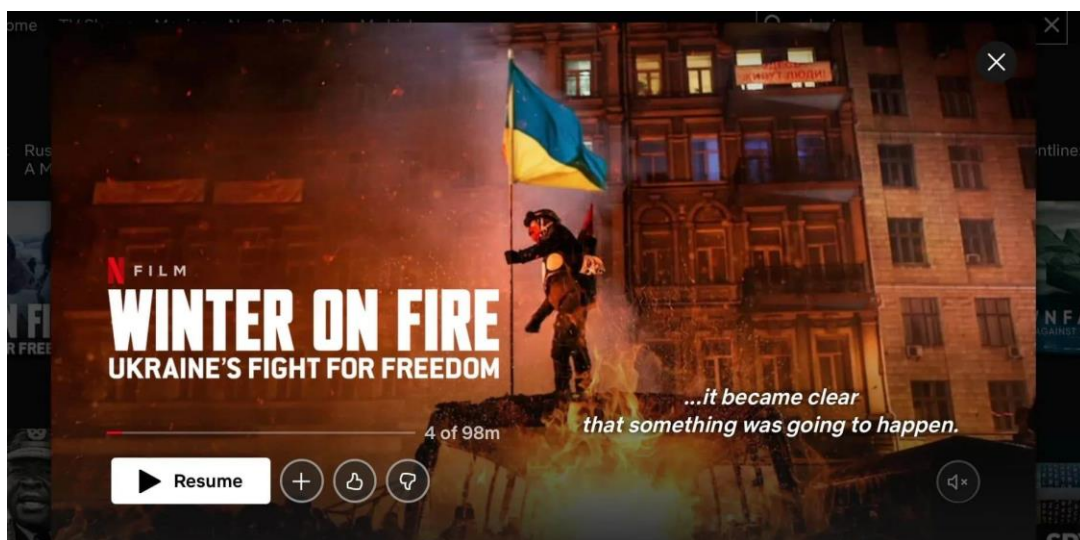


Рисунок 1.5 – Сторінка рекомендацій сервісу «Netflix»

Megogo [20] – це один із провідних відеосервісів, який пропонує широкий спектр медіа – контенту для онлайн – перегляду. Його аудиторія нараховує понад 50 мільйонів глядачів, а бібліотека на 2024 рік включала близько 77 000 матеріалів, серед яких художні та документальні фільми, мультфільми, серіали, телепередачі, а також спортивні, культурні програми, новини й музичні відео. Такий обсяг контенту створює необхідність у застосуванні ефективних рекомендаційних систем, які допомагають користувачам швидко знаходити цікавий для них матеріал.

Система рекомендацій WatchStat, розроблена в межах платформи Megogo, є потужним інструментом для персоналізації контенту на основі аналізу переглянутого відео. Вона пропонує рекомендації користувачам, які вже переглянули принаймні два відео, що дозволяє системі враховувати попередні вподобання глядача. Цей підхід дозволяє створювати персоналізовані списки фільмів і серіалів, які найбільше відповідають інтересам конкретного користувача, на основі його попередніх виборів.

Застосування подібного підходу може бути інтегровано й у створення нових інформаційних технологій для рекомендування фільмів. Аналіз вмісту, жанрових характеристик, рейтингу, акторського складу та інших параметрів дозволяє створювати більш точні й персоналізовані рекомендації. Наприклад, система може враховувати як історію переглядів користувача, так і додаткові критерії, такі як улюблені актори, жанри чи режисери. Це дає можливість сформувати рекомендації, які максимально релевантні і точні для кожного глядача.

Крім того, подібні системи можуть бути адаптовані для роботи з великими базами даних, що містять тисячі фільмів і серіалів. В результаті впровадження таких рішень значно підвищується зручність використання відео платформи та задоволеність користувачів. Вони можуть швидше знаходити контент, який відповідає їхнім інтересам, а також насолоджуватися більш якісними та персоналізованими рекомендаціями. Це є важливим аспектом розвитку сучасних

відеосервісів, оскільки дозволяє платформам забезпечити користувачам позитивний досвід до сервісу. На рисунку 1.6 можна побачити рекомендований список фільмів сайту Megogo.

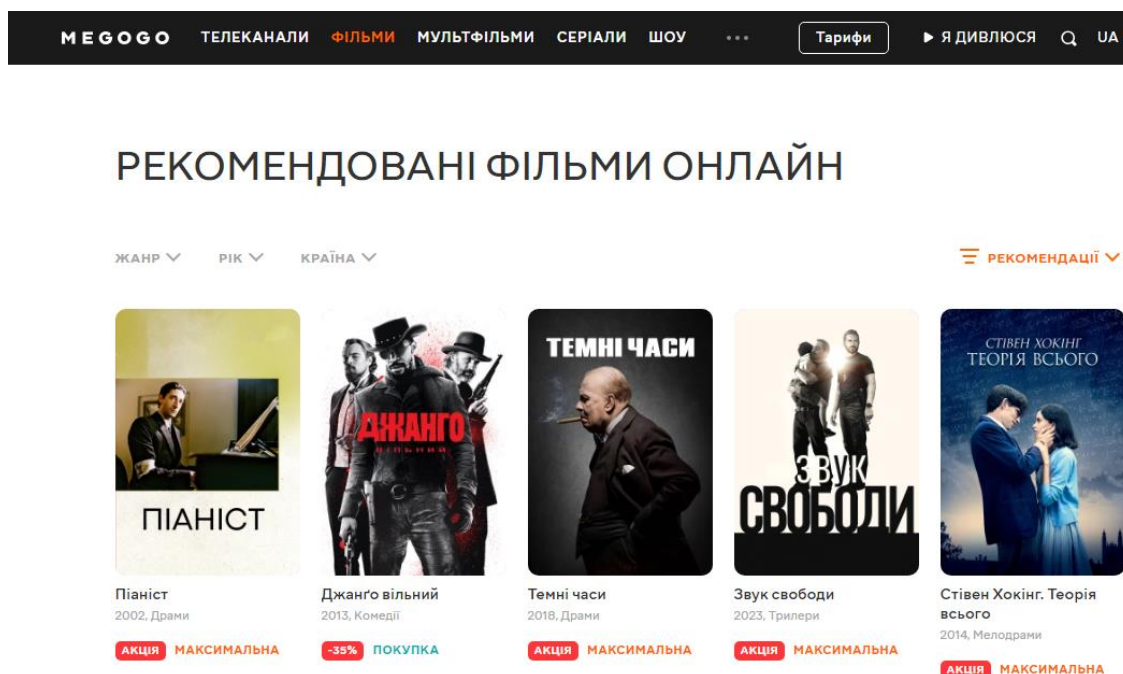


Рисунок 1.6 – Рекомендовані фільми сайту «Megogo»

IMDb [18] – це одна з найбільших онлайн-платформ для пошуку інформації про фільми, серіали, акторів та творців контенту. Кожного дня на сайті з'являються нові відгуки, рейтинги та рецензії, що допомагають користувачам знаходити релевантний контент на основі їхніх вподобань. Щомісяця IMDb відвідують мільйони користувачів, які шукають фільми, серіали та інші медіа-ресурси, що відповідають їхнім інтересам.

Для забезпечення високої ефективності пошуку та рекомендацій, IMDb активно використовує алгоритми штучного інтелекту та машинного навчання, що дозволяють аналізувати величезний обсяг даних і пропонувати користувачам найбільш релевантний контент. Це дозволяє створити персоналізовані рекомендації, які враховують історію переглядів, оцінки фільмів, жанри та уподобання користувачів. На рисунку 1.7 представлена сторінка IMDb, що демонструє, як платформа пропонує фільми та серіали, які відповідають

інтересам користувача, забезпечуючи швидкий доступ до актуального та цікавого контенту.

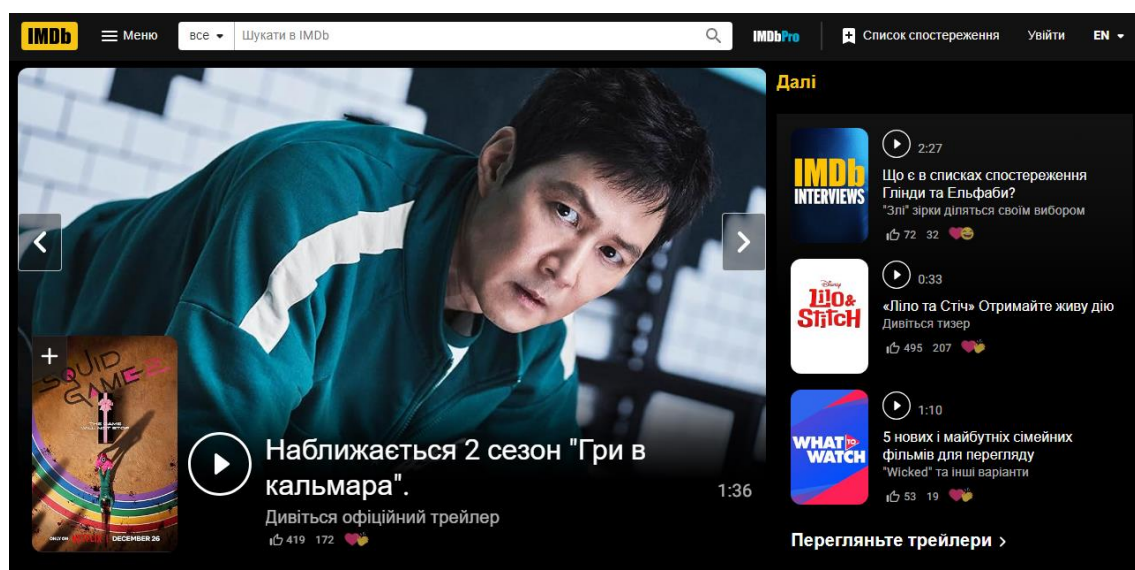


Рисунок 1.7 – Загальний вигляд інтерфейсного вікна системи IMDb

YouTube [21] – це величезна платформа, на якій щохвилини завантажуються щонайменше 300 годин нових відео різного жанру та тематики. Щомісяця її відвідують близько 3 мільярдів користувачів, які переглядають понад годину відео. Це робить YouTube одним із найпопулярніших сервісів у світі, де можна знайти контент на будь-який смак – від навчальних відео та влогів до музичних кліпів і також фільмів.

Для ефективної обробки таких величезних обсягів даних компанія активно впроваджує технології штучного інтелекту. Завдяки цьому вдалося спростити безліч процесів, зокрема рекомендацію контенту, автоматичну модерацію та аналіз переглядів. Це дозволяє користувачам отримувати персоналізовані пропозиції, які відповідають їхнім інтересам. На рисунку 1.8 представлена сторінка з фільмами, що ілюструє, як YouTube намагається задовольнити потреби своїх користувачів, пропонуючи актуальний і цікавий контент.

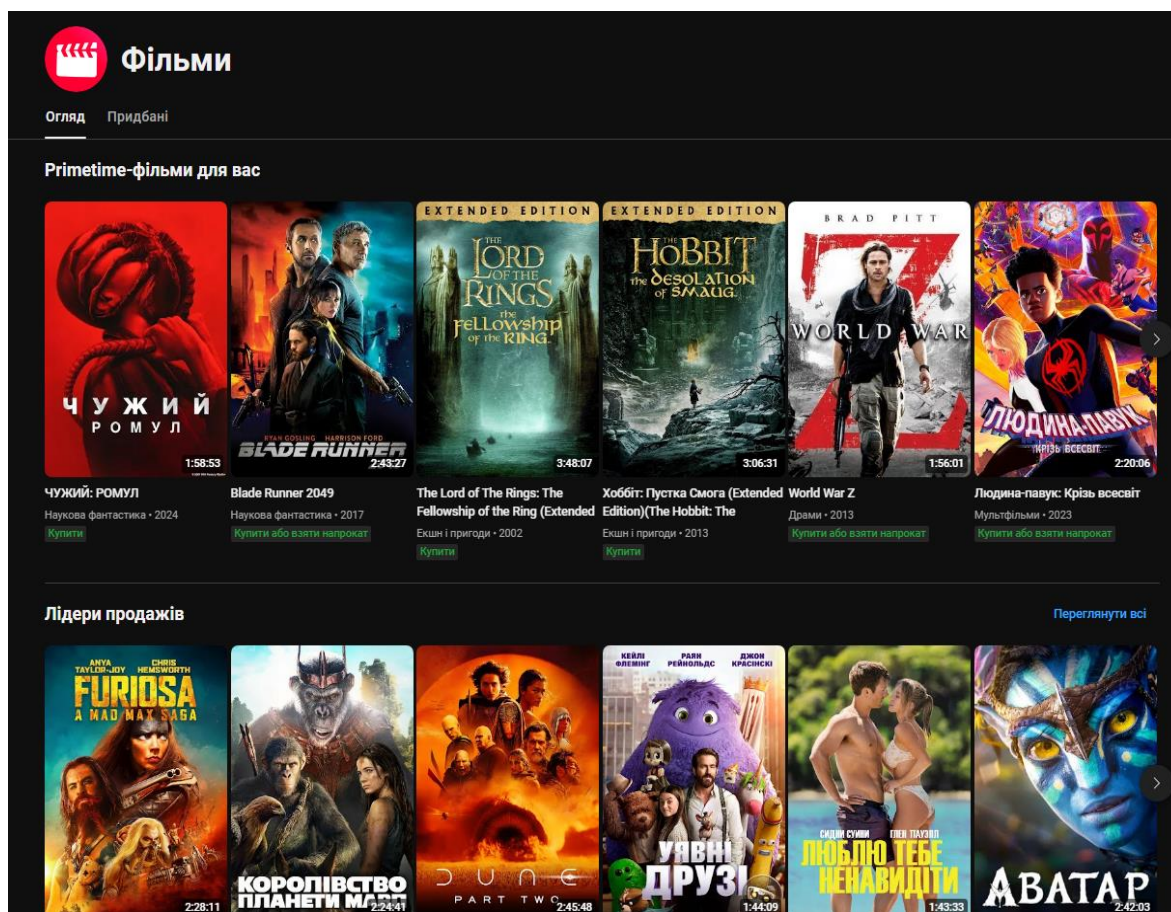


Рисунок 1.8 – YouTube пропонований контент

1.7 Постановка задачі дослідження

Протягом останніх років розвиток інтернет-технологій та зростання популярності стримінгових платформ створили великий обсяг даних, які відображають уподобання користувачів. Завдяки цьому стало можливим використовувати інформаційні технології для більш точного прогнозування інтересів глядачів. Рекомендаційні системи відіграють важливу роль у персоналізації контенту, дозволяючи глядачам швидко знаходити фільми та серіали, які відповідають їхнім інтересам.

Сучасні алгоритми рекомендацій включають колаборативну фільтрацію, контентну фільтрацію та їхні гібридні поєднання. Наприклад, такі платформи, як Netflix і YouTube, активно застосовують ці алгоритми для автоматичної адаптації рекомендацій на основі історії переглядів і оцінок

користувачів. Такі підходи не тільки підвищують точність рекомендацій, але й стимулюють залученість користувачів, що є важливим для конкурентоспроможності платформ.

У межах цього дослідження буде розроблено систему, яка аналізуватиме вподобання глядачів для побудови ефективної моделі рекомендацій. Запропонована система зможе налаштовувати рекомендації відповідно до конкретних вимог користувача, таких як вибір фільмів за жанром, акторським складом або режисером. Це зробить систему універсальною та адаптованою до різних платформ, включаючи стрімінгові сервіси та онлайн-кінотеатри.

Інтеграція гібридного підходу дозволяє максимально враховувати як індивідуальні вподобання кожного користувача, так і загальні групові тенденції, що дає змогу забезпечити високий рівень персоналізації рекомендацій. Колаборативна фільтрація допомагає використовувати досвід інших користувачів для визначення схожих уподобань, у той час як аналіз контенту дозволяє глибше зрозуміти атрибути фільмів, які можуть зацікавити конкретного глядача.

Система інтегрована в додаток з графічним інтерфейсом, що дозволить користувачам ефективно взаємодіяти з платформою. Метою буде надання рекомендацій, ґрунтуючись на різноманітних факторах, таких як жанри, рейтинги і жанрові вподобання користувача. Такий підхід дозволить створити систему, яка не лише точно відповідатиме інтересам користувачів, але й розширюватиме можливості з урахуванням зміни вподобань з часом.

Окрім базових функцій персоналізованих рекомендацій, система має можливість для інтеграції з зовнішніми базами даних. Це відкриває перспективу для гнучкої адаптації до нових джерел даних та можливості розширення функціональності програми. Користувачі зможуть завантажувати додаткову інформацію, що дозволить платформі постійно розвиватися та адаптуватися до нових тенденцій у цифровому середовищі.

1.8 Висновок до розділу 1

У першому розділі розглянуто основні методи побудови рекомендаційних систем, серед яких фільтрація на основі вмісту, спільна фільтрація та їхня комбінація у вигляді гібридного підходу. Проведений аналіз виявив, що фільтрація на основі вмісту дозволяє створювати рекомендації, враховуючи характеристики контенту, з яким взаємодівав користувач. Водночас цей метод має обмеження щодо врахування різноманітності та новизни контенту. Спільна фільтрація забезпечує врахування вподобань інших користувачів, але стикається з проблемами на початкових етапах роботи, коли недостатньо даних для аналізу.

Гібридний підхід, який поєднує переваги колаборативної фільтрації і контентної фільтрації, довів свою ефективність у створенні адаптивних і масштабованих рекомендаційних систем. Він дозволяє враховувати індивідуальні уподобання користувачів і водночас використовувати колективний досвід інших для підвищення точності рекомендацій. Це робить гібридний метод найбільш перспективним для побудови сучасних систем у сфері кінематографії.

У рамках аналізу також було розглянуто практичне застосування цих методів на популярних платформах. Результати показали, що використання комбінованих алгоритмів дозволяє цим платформам адаптувати рекомендації до змін у поведінці користувачів і швидко обробляти великі обсяги даних.

Проведений аналіз підкреслив необхідність інтеграції гібридного підходу для створення рекомендаційної системи, яка враховуватиме різні фактори, зокрема жанрові вподобання, акторський склад та історію переглядів. Така система дозволить вирішити проблеми масштабованості, адаптивності та релевантності рекомендацій, що особливо важливо для сучасних стрімінгових платформ.

Запропонований гібридний підхід дозволяє створити інформаційну технологію, яка відповідатиме потребам сучасних користувачів, забезпечуючи точність, персоналізацію та зручність взаємодії.

2 МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ЩОДО РЕКОМЕНДУВАННЯ ФІЛЬМІВ ГЛЯДАЧАМ

2.1 Розробка методу розв'язання задачі рекомендації фільмів

Розвиток рекомендаційних систем значно прискорився завдяки сучасним алгоритмам, які аналізують уподобання користувачів на основі їхніх попередніх переглядів, рейтингів та пошукових запитів. Це особливо важливо в умовах великої кількості контенту, що доступний на стримінгових платформах та в онлайн-кінотеатрах. Попри ефективність цих систем, якість рекомендацій часто знижується через кілька ключових факторів.

По-перше, обмеженість бази фільмів або недостатність інформації про конкретного користувача створює ситуацію, коли система надає менш релевантні рекомендації. Проблема «холодного старту» виникає, коли новий користувач ще не має історії переглядів або уподобань, що ускладнює індивідуальне налаштування рекомендацій. По-друге, відмінності у вподобаннях різних вікових груп і культурних середовищ також ускладнюють прогнозування, оскільки смаки глядачів можуть варіюватися не лише за жанрами, а й за стилістикою, акторами чи режисерськими почерками. Крім того, обмеження за жанровою специфікою і популярністю фільмів іноді спричиняють надмірну орієнтацію на загальні рекомендації, що знижує точність для кожного окремого користувача [10].

Для вирішення цих проблем сучасні рекомендаційні системи зазвичай проходять через кілька основних етапів, що дозволяють підвищити точність та персоналізацію. Перший етап включає попередній аналіз даних, під час якого формується база фільмів із детальною інформацією про кожен із них, включаючи жанр, акторський склад, режисера тощо. Для користувачів створюються профілі на основі їхніх вподобань і переглядів. Наступний етап – це витяг ознак, коли система виділяє ключові характеристики для кожного фільму та визначає, які елементи можуть зацікавити конкретного

користувача. Важливо, щоб ці ознаки були релевантними і могли точно відображати смаки користувача на основі його попередніх уподобань.

Для побудови рекомендацій використовується поєднання алгоритмів колаборативної та контентної фільтрації. Колаборативна фільтрація базується на тому, що користувачі з подібними інтересами схильні оцінювати однакові фільми, тому система може прогнозувати уподобання одного користувача на основі вподобань інших. Контентна фільтрація, у свою чергу, фокусується на аналізі самих характеристик фільмів, підбираючи їх за схожістю з контентом, який користувач раніше оцінював позитивно.

2.2 Обґрунтування вибору базового методу щодо рекомендування фільмів

Вибір методу побудови рекомендаційної технології є критичним для ефективного рекомендування фільмів, особливо в умовах різноманітних потреб користувачів та великого обсягу контенту, що постійно зростає. Рекомендаційна система повинна забезпечувати швидкий доступ до релевантних фільмів, враховуючи інтереси кожного користувача, і водночас залишатися гнучкою та масштабованою. У сучасній практиці використовуються кілька основних підходів, серед яких контентна фільтрація, колаборативна фільтрація та їхні комбінації. Кожен із цих підходів має свої переваги, обмеження та специфічні сценарії застосування.

Контентна фільтрація. Цей метод базується на аналізі характеристик самих фільмів, таких як жанр, акторський склад, режисер, сюжетні елементи та рейтинг. Рекомендації формуються на основі подібних характеристик, відбираючи контент, що відповідає попереднім уподобанням користувача. Наприклад, якщо користувач часто дивиться науково-фантастичні фільми, система буде пропонувати інші стрічки цього жанру [14]. Основна перевага контентної фільтрації полягає в можливості надання рекомендацій навіть для

нових користувачів без історії переглядів, оскільки вона спирається виключно на дані про контент. Однак цей підхід має обмеження:

1. Система часто пропонує контент зі схожими характеристиками, що знижує різноманітність рекомендацій.
2. Не враховує зміни вподобань користувача з часом, що робить її менш гнучкою для тривалої персоналізації.

Колаборативна фільтрація. На відміну від контентної фільтрації, колаборативна фільтрація орієнтована на аналіз взаємодії між користувачами. Система використовує інформацію про вподобання та перегляди інших глядачів із подібними смаками, щоб створювати рекомендації. Наприклад, якщо певній групі користувачів подобається однаковий набір фільмів, система може передбачити, що новий фільм, переглянутий одним із них, може бути цікавий і для інших членів групи. Переваги колаборативної фільтрації включають:

- можливість виявлення прихованих взаємозв'язків між користувачами, які можуть не бути очевидними через жанрові характеристики фільмів.
- сприяння персоналізації завдяки врахуванню широкого спектру користувацьких вподобань.

Однак колаборативна фільтрація має недоліки:

- вона потребує значної кількості даних для нових користувачів або фільмів, що створює так звану проблему "холодного старту".
- рекомендації можуть бути неточними для користувачів, які мають унікальні або вузькі інтереси, через відсутність аналогічних профілів серед інших глядачів.

Поєднання методів. Використання лише одного з методів у чистому вигляді може обмежувати гнучкість та ефективність рекомендаційної системи. Наприклад, для нових користувачів контентна фільтрація є більш корисною, оскільки система може створити базові рекомендації на основі характеристик фільмів. Колаборативна фільтрація, натомість, є ефективною

для користувачів із більш розширеною історією переглядів та оцінок, оскільки враховує соціальний аспект та загальні вподобання. Тому, обираючи підхід для побудови рекомендаційної системи, доцільно враховувати комбінацію цих методів для підвищення точності та персоналізації рекомендацій, що дозволяє ефективно обслуговувати як нових, так і досвідчених користувачів.

2.3 Аналіз сценаріїв роботи інформаційної технології

Перед початком розробки інформаційної технології для аналізу та рекомендації фільмів важливо визначити можливі сценарії її роботи. Це допомагає зрозуміти, як система буде взаємодіяти з користувачами, які дані вона використовуватиме для аналізу, а також як забезпечуватиметься персоналізований підхід до формування рекомендацій. Запропонована технологія включає три основні сценарії [17], кожен із яких пропонує різний рівень персоналізації, відповідаючи потребам широкого кола користувачів.

У першому сценарії користувач отримує рекомендації виключно на основі свого списку. У цьому випадку система використовує дані, вже доступні в її базі, без потреби у додаткових налаштуваннях або введенні нової інформації. Такий підхід є особливо ефективним для користувачів, які мають багатий досвід взаємодії з платформою, оскільки алгоритм аналізує історію їхніх оцінок та вподобань для створення персоналізованих рекомендацій. Наприклад, якщо користувач регулярно оцінює фільми певного жанру або з участю конкретного актора, система враховує ці дані та пропонує схожі варіанти. Цей сценарій забезпечує мінімальне навантаження на користувача та дозволяє швидко отримати рекомендації, релевантні його смакам.

Другий сценарій передбачає ситуацію, коли користувач хоче знайти фільми, схожі на ті, що йому вже сподобалися. Для цього достатньо ввести назву переглянутого фільму, на основі якого система створить нові

пропозиції. Алгоритм аналізує ключові характеристики цільового фільму, такі як жанр, акторський склад, режисер, сюжетні елементи та рейтинг, і формує рекомендації на основі подібних ознак. Наприклад, якщо користувач вводить назву популярного трилера, система може запропонувати інші фільми цього жанру або роботи того ж режисера. Такий підхід дозволяє значно звужити діапазон пошуку та отримати рекомендації, що максимально відповідають очікуванням користувача. Це особливо зручно для тих, хто шукає контент, схожий на вже переглянутий.

Третій сценарій надає користувачеві можливість самостійно налаштовувати параметри рекомендацій, забезпечуючи найвищий рівень персоналізації. У цьому випадку користувач може вибрати бажаний жанр, зазначити акторів або режисерів, які його цікавлять, а також уточнити інші характеристики фільмів. Наприклад, користувач може попросити систему знайти драми з участю конкретного актора. Крім того, він може комбінувати кілька критеріїв одночасно для отримання більш точних результатів. Цей підхід дозволяє користувачеві контролювати процес пошуку та робить систему зручною для тих, хто має специфічні запити або унікальні інтереси.

Щоб краще зрозуміти функціонування системи, нижче представлено схему, яка ілюструє описані сценарії. Ця діаграма допомагає візуалізувати послідовність кроків, які виконує система під час аналізу даних і формування рекомендацій.

1. Початок роботи;
2. Створення рекомендації;
3. Вихідні дані;
4. Завершення роботи;

Цей сценарій є найпростішим із трьох, оскільки не потребує від користувача додаткових дій. Він ідеально підходить для ситуацій, коли користувач хоче швидко отримати рекомендації, використовуючи вже наявну інформацію. Це забезпечує зручність і швидкість роботи, що важливо

для платформ із великою кількістю контенту.

Кожен із запропонованих сценаріїв враховує специфіку використання рекомендаційної системи, що дозволяє забезпечити максимальну зручність і ефективність у роботі з різними категоріями користувачів.

На рисунку 2.1 представлені етапи сценарію функціонування інформаційної технології:

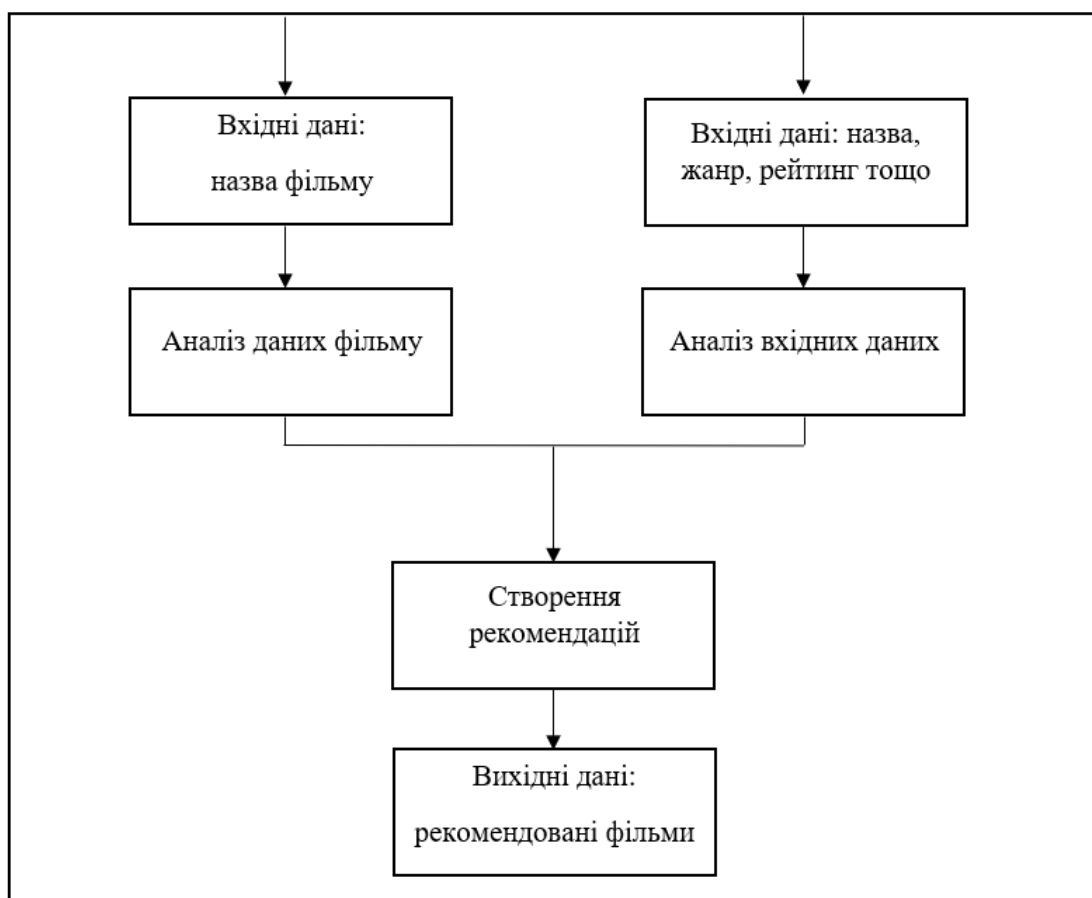


Рисунок 2.1 – Діаграма сценаріїв роботи технології аналізу та рекомендації фільмів.

Другий сценарій включає такі етапи:

1. Початок роботи;
2. Введення даних (назва фільму);
3. Аналіз даних цільового фільму;
4. Генерація рекомендацій;

5. Завершення роботи.

Цей сценарій передбачає, що користувач повинен ввести назву фільму, на основі якого він хоче отримати рекомендації. Завдяки аналізу введених даних система може запропонувати фільми, які подібні за жанровою, тематичною або іншими ознаками до вибраного фільму, що підвищує точність рекомендацій.

Третій сценарій включає такі етапи:

1. Початок роботи;
2. Введення даних (жанри, актори, режисер тощо);
3. Аналіз вхідних даних;
4. Генерація рекомендацій;
5. Завершення роботи.

Цей сценарій має подібний алгоритм до другого, проте користувачеві потрібно ввести конкретні дані, які описують фільм, такі як жанри, актори, режисери тощо.

Таким чином, розробка інформаційної технології щодо рекомендування фільмів глядачам ґрунтується на трьох основних сценаріях роботи. Кожен сценарій відповідає різним потребам користувача і забезпечує адаптацію системи до рівня персоналізації, який найбільше підходить для конкретної ситуації. Це робить технологію гнучкою, універсальною та зручною у використанні для широкого спектру користувачів, незалежно від їхніх вподобань та інтересів.

2.4 Розробка математичної моделі надання рекомендацій щодо підбору фільмів

Дерево рішень – це метод ухвалення рішень, який ґрунтується на послідовних перевірках умов. Такий підхід дозволяє поетапно обмежувати кількість варіантів і знаходити найбільш релевантний результат. Метод особливо корисний у задачах, де вибір залежить від набору умов, які перевіряються у

певній послідовності. У рамках інформаційної технології щодо рекомендації фільмів глядачам цей підхід використовується для аналізу таких характеристик, як жанр, рейтинг та режисер, щоб надати користувачеві найбільш підходящі рекомендації.

Опис роботи дерева рішень [16]:

Уявімо, що є множина об'єктів B (фільми), а також набір функцій F , які здійснюють перевірки певних умов (наприклад, перевіряють жанр фільму, дату його виходу чи режисера). Кожна функція виконує перевірку і повертає результат у вигляді значення з множини $E_k = \{0, 1, \dots, k - 1\}$, яке визначає результат перевірки (наприклад, "фільм – комедія" чи "рік випуску – 2024").

Процес аналізу виглядає так: для кожного об'єкта з множини B послідовно перевіряються визначені умови. Вузли структури відповідають конкретним перевіркам, а гілки – результатам цих перевірок (наприклад, "жанр – бойовик", "рейтинг – середній", "режисер – Стівен Спілберг"). У результаті вибір поступово звужується, і система пропонує найбільш релевантний варіант.

Завдяки чіткій структурі, цей підхід дозволяє ефективно визначати, який фільм найбільше відповідає потребам користувача. Наприклад, на першому етапі аналізується жанр, потім рейтинг, а завершальним критерієм може бути режисер.

Цей метод також дозволяє легко інтегрувати додаткові параметри для більш точних рекомендацій. Такі модифікації підвищують точність і гнучкість системи, дозволяючи підлаштовувати її під індивідуальні вподобання користувачів.

Під час реалізації цього підходу важливо враховувати, що дерево рішень може бути налаштоване таким чином, щоб в кінці користувач отримував рекомендації на основі декількох ключових факторів. Це дозволяє системі враховувати, наприклад, не тільки жанр і рейтинг, але й переваги користувача щодо режисера або акторів. Дерево рішень інформаційної технології щодо рекомендації фільмів наведено на рисунку 2.2.

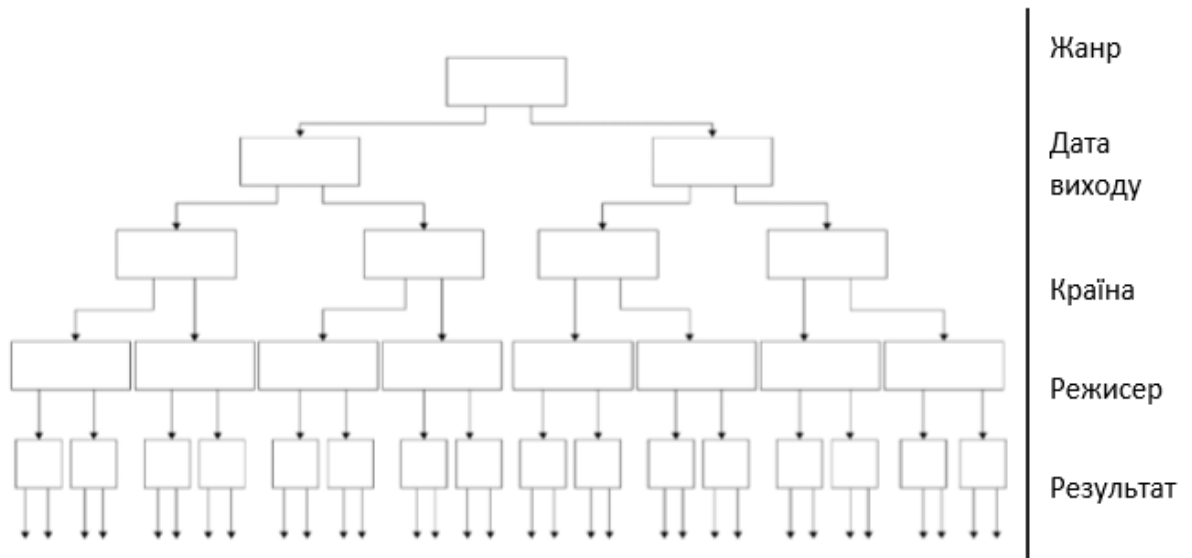


Рисунок 2.2 – Структурна схема дерева рішень інформаційної технології щодо рекомендування фільмів глядачам

Гібридний підхід: колаборативна фільтрація та контентний аналіз [9]

Для покращення точності рекомендацій використовуємо гібридний підхід, що поєднує два основні методи: колаборативну фільтрацію та контентний аналіз. Це дозволяє отримати більш комплексні рекомендації, які враховують як характеристики самих фільмів, так і уподобання інших користувачів, схожих на поточного.

1. Колаборативна фільтрація

Колаборативна фільтрація базується на припущенні, що користувачі, які мають схожі вподобання, ймовірно, матимуть схожі вподобання і в майбутньому. Цей метод використовує взаємодії користувачів з елементами (наприклад, рейтинг фільмів), щоб рекомендувати нові фільми.

Формула для обчислення схожості між двома користувачами, використовуючи метрику косинусної схожості, виглядає так:

$$S(u, v) = \frac{\sum_{i=1}^n r^2 \cdot Tvi}{\sqrt{\sum_{i=1}^n r^2 ui} \cdot \sqrt{\sum_{i=1}^k r^2 vi}}, \quad (2.1)$$

де: $S(u, v)$ – схожість між користувачами u і v ; r_{ui} – оцінка користувача u для фільму i ; r_{vi} – оцінка користувача v для фільму i ; n – кількість фільмів, на основі оцінок, які поставили користувач.

За допомогою цієї метрики система може знайти схожі уподобання поточного користувача і на основі цього рекомендувати нові фільми.

2. Контентна фільтрація

Контентна фільтрація зосереджується на характеристиках самих фільмів (наприклад, жанр, актори, режисер, рік випуску). Для кожного фільму створюється профіль, що описує його характеристики, і на основі цього порівнюються фільми, які найбільше відповідають вимогам користувача.

Формула для оцінки схожості між фільмами на основі контенту виглядає так:

$$S(c1, c2) = \frac{\sum_{i=1}^m w^2 \cdot (x_{i1} \cdot x_{i2})}{\sqrt{\sum_{i=1}^m r^2 w^2 \cdot x_{i1}^2} \cdot \sqrt{\sum_{i=1}^m w^2 \cdot x_{i2}^2}}, \quad (2.2)$$

де: $S(c1, c2)$ – схожість між двома фільмами $c1$ і $c2$, w_i – вага параметра i (наприклад, жанр, режисер), x_{i1}, x_{i2} – значення параметра i для фільмів $c1$ і $c2$, m – кількість параметрів, що використовуються для оцінки схожості (наприклад, жанр, рейтинг, режисер).

3. Гібридний підхід

Для покращення точності рекомендацій можна використовувати гібридний підхід, який поєднує методи колаборативної фільтрації та контентної фільтрації, доповнюючи їх додатковими параметрами, що впливають на якість рекомендацій. Такий підхід дозволяє системі більш гнучко враховувати різноманітні аспекти користувацьких вподобань і характеристик фільмів, забезпечуючи більш точне прогнозування інтересів користувачів. Гібридний підхід враховує не лише базові параметри, такі як оцінки або жанр фільму, але й додаткові фактори, як, наприклад, контекстні змінні, що можуть включати час

доби, сезонні тенденції або навіть культурні відмінності в сприйнятті контенту. Формула для гібридної оцінки схожості:

$$Shybrid(i, j) = w1 \cdot Scontent(i, j) + w2 \cdot Scollab(i, j), \quad (2.3)$$

де: $Shybrid(i, j)$ – гібридна оцінка схожості між фільмами i і j , $Scontext(i, j)$ – оцінка схожості за контентом, $Scollab(i, j)$ – оцінка схожості за колаборативною фільтрацією, $w1$ та $w2$ – вагові коефіцієнти для кожного підходу, що зазвичай визначають їхню значимість в кінцевій оцінці.

Цей підхід дозволяє врахувати як індивідуальні вподобання користувача, так і характеристики фільмів, що дозволяє забезпечити найбільш точні та персоналізовані рекомендації.

2.5 Розробка алгоритму функціонування інформаційної технології щодо рекомендування фільмів

Дану інформаційну технологію можна поділити на чотири ключові модулі, кожен з яких виконує важливу роль у формуванні рекомендацій для користувачів. Завдяки взаємодії цих модулів створюється цілісна система, здатна ефективно аналізувати дані та забезпечувати персоналізовані пропозиції.

Одним із найважливіших компонентів є модуль навчання, який відповідає за оптимізацію роботи алгоритмів машинного навчання. Основне завдання цього модуля полягає у виявленні закономірностей у даних, що дозволяє системі вдосконалювати свої прогнози та рекомендації. Завдяки модулю навчання програма може аналізувати переваги користувачів і виявляти залежності між параметрами, такими як жанр фільму, рік випуску, режисер або рейтинг. Це дозволяє системі поступово адаптуватися до індивідуальних потреб кожного користувача.

Для забезпечення роботи модуля навчання використовується набір даних, що включає інформацію про фільми та історію переглядів користувачів. Дані

зберігаються у форматах .csv або .json, які є універсальними і зручними для роботи з великими масивами інформації. Зазвичай цей набір даних включає такі атрибути, як назва фільму, жанр, рейтинг, а також інформацію про користувачів: їхні оцінки, вподобання та час перегляду. На основі цих даних система здатна аналізувати минулі уподобання користувачів та пропонувати релевантні варіанти для перегляду.

Ще однією ключовою особливістю модуля навчання є можливість постійного оновлення бази даних. Завдяки регулярному додаванню нової інформації про фільми, що з'являються, та змін у вподобаннях користувачів, система залишається актуальною та ефективною. Це також дозволяє враховувати зміну трендів у кінематографі або появу нових популярних жанрів.

Крім того, модуль навчання інтегрується з іншими компонентами системи, такими як модуль обробки даних і модуль аналізу, що підвищує ефективність і точність рекомендацій.

Таким чином, модуль навчання є центральним елементом інформаційної технології рекомендацій фільмів, що забезпечує високий рівень персоналізації та релевантності результатів.

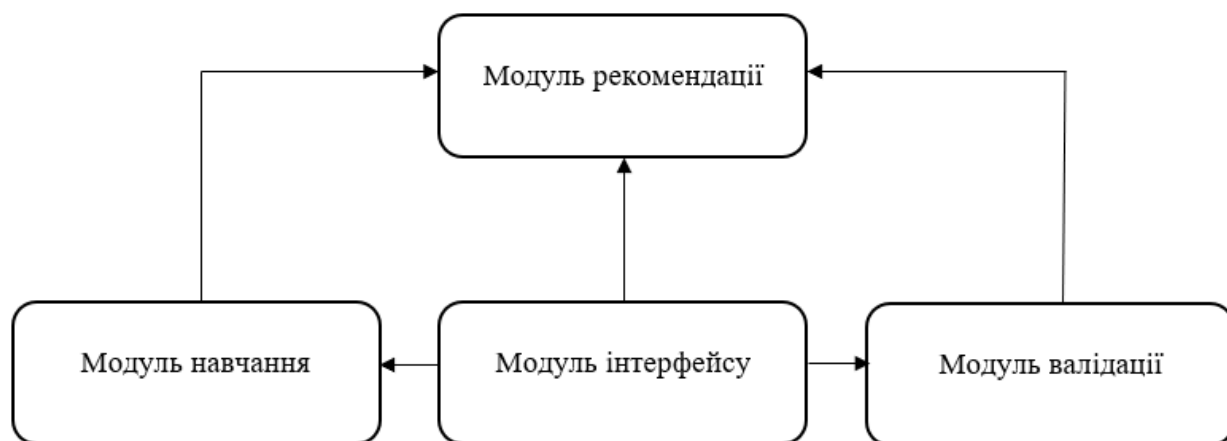


Рисунок 2.3 – Загальна структура інформаційної системи щодо рекомендування фільмів глядачам

Модуль графічного інтерфейсу користувача є важливою та невід’ємною частиною системи рекомендування фільмів. Через модуль графічного інтерфейсу користувач може швидко переглядати персоналізовані списки фільмів, шукати стрічки за різними параметрами (наприклад, жанром, роком випуску чи акторами), а також налаштовувати систему для своїх потреб. Користувач може змінювати пріоритетність жанрів, встановлювати додаткові фільтри для пошуку та обирати бажаний формат результатів.

Модуль рекомендацій виконує ключову функцію всієї системи, надаючи користувачеві персоналізовані рекомендації. Для цього система аналізує інформацію про вподобання користувача, його історію переглядів і формує список фільмів, які найбільше відповідають його інтересам. У процесі створення рекомендацій використовуються складні алгоритми аналізу даних, які обробляють велику базу даних, що включає тисячі фільмів із детальною інформацією про жанри, рейтинги, відгуки глядачів та акторський склад. Завдяки цьому система забезпечує точність та релевантність запропонованих фільмів.

Модуль валідації рекомендацій відповідає за аналіз ефективності та надійності системи. Він дозволяє користувачеві отримати доступ до детальної статистики, яка демонструє, наскільки точно рекомендовані фільми відповідають його вподобанням. Це дозволяє візуально оцінити ефективність роботи алгоритмів. Для тестування роботи системи використовувалися два набори даних: основний набір (train), який включає близько 5 000 фільмів, і тестовий набір (val), який містить 3 000 фільмів. Такий підхід дозволяє ретельно перевірити якість рекомендацій і забезпечити їх високу точність у реальних умовах.

Модуль навчання відповідає за оновлення, переналаштування та вдосконалення алгоритмів системи рекомендацій. Після додавання нового набору даних, який може включати як нові фільми, так і змінені вподобання користувачів, система автоматично запускає процес перенавчання моделі. Це дозволяє адаптувати систему до нових вимог і гарантувати, що рекомендації

залишатимуться максимально актуальними та релевантними. Процес перенавчання забезпечує інтеграцію нових даних без втрати точності попередньо створених моделей [11,14].

Наприклад, через модуль рекомендацій користувач отримує персоналізовані пропозиції фільмів, засновані на його вподобаннях, переглядах та інтересах. Модуль навчання забезпечує можливість оновлення системи для врахування нових даних, а модуль валідації перевіряє точність роботи системи, забезпечуючи її надійність та стабільність. Таким чином, усі модулі тісно пов'язані між собою, що дозволяє створити ефективну інформаційну систему для формування якісних рекомендацій.

2.6 Розробка структури інформаційної технології щодо рекомендування фільмів глядачам

На вхід система отримує запити користувача у вигляді тексту або вибраних параметрів. Це можуть бути жанри фільмів, актори, а також конкретні запити, наприклад: "Ключові слова фільму".

Після отримання запиту користувача інформація передається у модуль обробки даних, який відповідає за первинний відбір релевантних фільмів із великої бази даних. База даних системи є структурованим джерелом інформації, яке містить докладні характеристики кожного фільму, такі як жанр, акторський склад, рейтинг, опис фільму тощо. Завдяки таким даним модуль обробки враховує задані користувачем фільтри і формує попередній список фільмів, що відповідають зазначеним критеріям. Крім того, на цьому етапі здійснюється попередня фільтрація, яка виключає варіанти, що не відповідають технічним чи тематичним вимогам.

Далі система використовує методи багатofакторного аналізу для оцінки відповідності фільмів запиту користувача. У процесі роботи враховуються різноманітні параметри, такі як жанр, оцінка, популярність, країна виробництва

та акторський склад. Ці фактори дозволяють створити більш точну і персоналізовану картину вподобань користувача. Важливою особливістю цього методу є можливість налаштування ваг кожного параметра як самою системою, так і користувачем залежно від його конкретних вподобань. Наприклад, якщо користувач надає перевагу певному жанру чи країні виробництва, система автоматично збільшить вагу цих критеріїв під час обробки запиту, що дозволить згенерувати більш точні рекомендації.

Результатом роботи модуля обробки даних стає персоналізований список фільмів, які ранжуються за рівнем релевантності до запиту користувача. У цьому списку фільми розташовуються в порядку зменшення відповідності заданим критеріям. Наприклад, на перших позиціях знаходяться стрічки, які найкраще відповідають жанровим і тематичним уподобанням користувача, мають високі оцінки від інших глядачів і значну популярність серед аудиторії. Такий підхід дозволяє користувачу швидко знаходити цікаві для нього варіанти, не витрачаючи час на перегляд великих списків фільмів.

Сформований список передається в модуль обробки вихідних даних, який відповідає за підготовку результатів у зручному для користувача форматі. Цей модуль додатково додає до кожного фільму детальну інформацію: назву, жанр, рік випуску, рейтинг, акторський склад, короткий опис сюжету та інші релевантні дані. Завдяки цьому користувач отримує не лише рекомендації, але й повну картину фільму, що значно покращує процес прийняття рішення про перегляд. Крім того, додаткові параметри можуть включати такі фактори, як тривалість фільму, доступні мови субтитрів чи наявність в системі рейтингів з різних джерел, що ще більше підвищує точність рекомендацій.

Такий підхід дозволяє користувачам отримувати найбільш релевантні рекомендації без необхідності переглядати великий обсяг інформації, забезпечуючи зручний та швидкий доступ до контенту, який відповідає їхнім інтересам.

Структура інформаційної технології рекомендації фільмів зображена на рисунку 2.4.

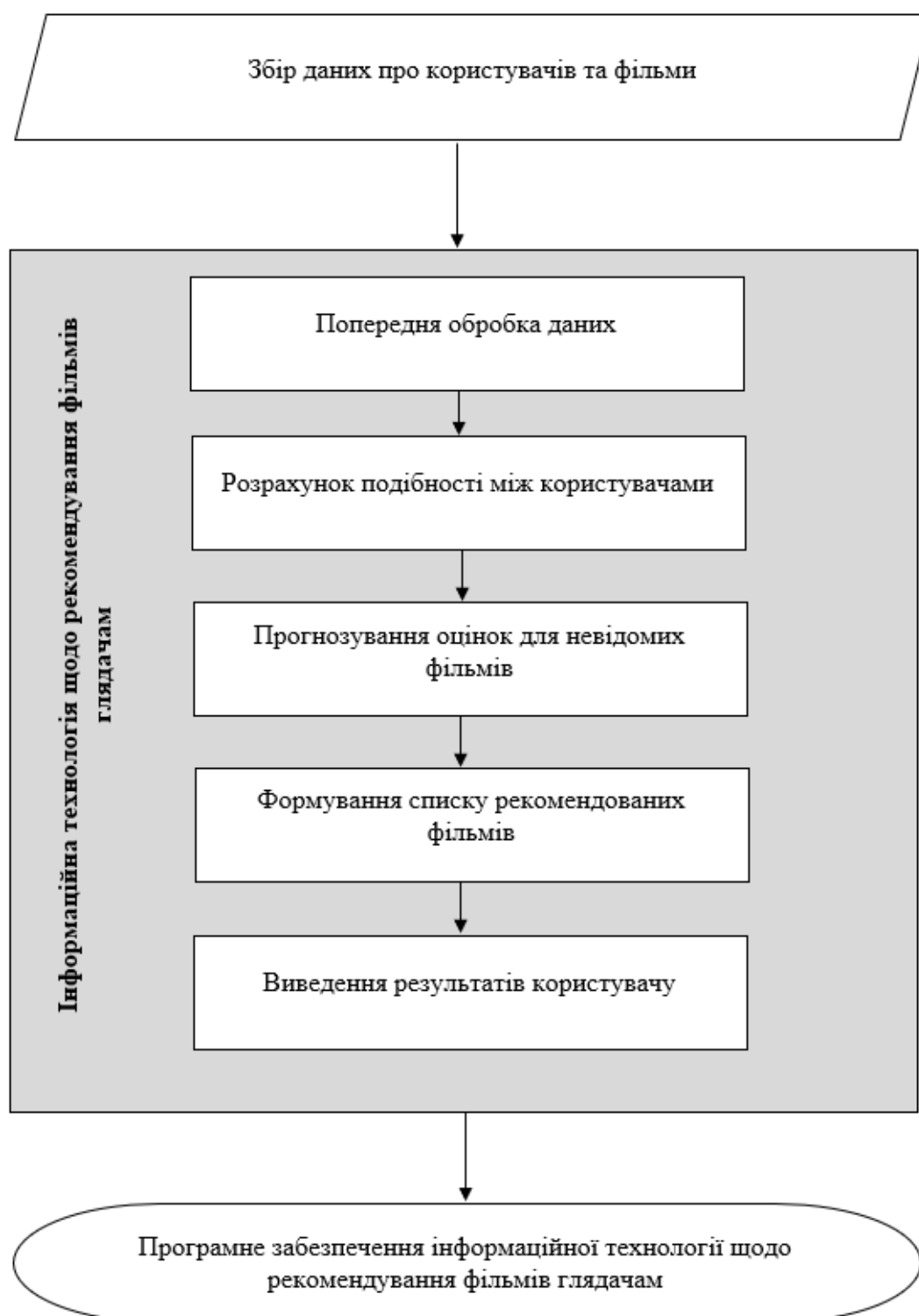


Рисунок 2.4 – Структура інформаційної технології щодо рекомендації фільмів глядачам

2.7 Розробка алгоритму інформаційної технології щодо рекомендації фільмів глядачам

Розглянемо загальний алгоритм роботи додатку, що реалізує інформаційну технологію рекомендації фільмів (рис. 2.5).

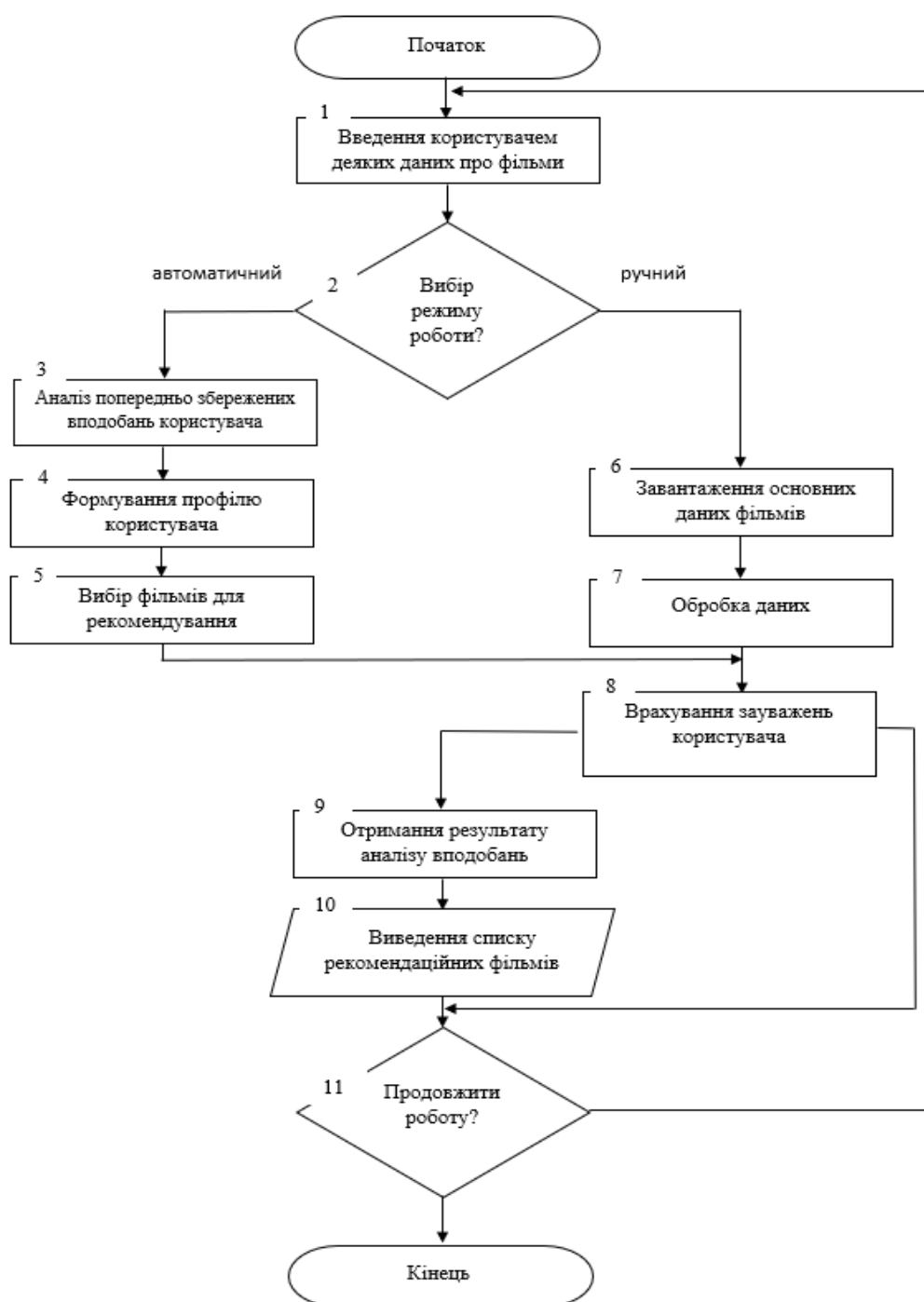


Рисунок 2.5 – Загальний алгоритм роботи інформаційної технології щодо рекомендації фільмів глядачам

Даний алгоритм складається з наступних кроків:

Крок 1. Ведення користувачем даних фільму.

На першому етапі користувач вводить базову інформацію про фільми, яка може включати назви, жанри, режисерів, оцінки тощо.

Крок 2. Вибір режиму роботи.

Користувач визначає бажаний режим роботи системи. Є два варіанти:

- автоматичний режим: система самостійно використовує попередньо збережені дані, щоб створити рекомендації;
- ручний режим: користувач вручну вводить параметри та уточнення, які будуть враховані в процесі.

Крок 3. Аналіз попередньо збережених вподобань користувача.

У випадку автоматичного режиму програма звертається до бази даних, де зберігається історія переглядів, оцінок і вподобань користувача.

Крок 4. Формування профілю користувача.

На основі аналізу вподобань програма створює персоналізований профіль користувача.

Крок 5. Вибір фільмів для рекомендації.

Використовуючи зібраний профіль, система проводить пошук у базі даних фільмів і формує список найбільш відповідних варіантів.

Крок 6. Завантаження основних даних про фільми.

Якщо обрано ручний режим, користувач вводить основні критерії пошуку. Це може бути жанр, рік випуску, режисер, або інші параметри, що відповідають його інтересам.

Крок 7. Обробка введених даних.

Система перевіряє введені дані на коректність, обробляє їх і адаптує для подальшого використання. Це забезпечує точність і узгодженість результатів.

Крок 8. Врахування зауважень користувача.

Користувач може додати уточнення або зауваження до отриманих рекомендацій.

Крок 9. Отримання результатів аналізу.

На цьому етапі система завершує аналіз, об'єднуючи всю доступну інформацію, та готує підсумковий результат.

Крок 10. Виведення списку рекомендованих фільмів.

Користувач отримує список рекомендацій, що відповідають його уподобанням і введеним критеріям. Кожен фільм у списку супроводжується описом, рейтингом чи іншими корисними деталями.

Крок 11. Продовження роботи.

Після отримання рекомендацій користувач має можливість вирішити, чи хоче він продовжувати роботу з системою, наприклад, змінити критерії або отримати новий список рекомендацій.

Крок 12. Завершення роботи з системою.

2.8 Висновок до розділу 2

У другому розділі було описано концепцію інформаційної технології рекомендування фільмів глядачам, яка базується на використанні гібридного підходу. Особливу увагу приділено аналізу вимог до системи, обґрунтуванню вибору методів і алгоритмів, а також формуванню структурної схеми, що відображає ключові етапи роботи технології: аналіз даних, генерацію рекомендацій та взаємодію з користувачем.

Також описано створення модулів спільної фільтрації та фільтрації на основі вмісту. Модуль спільної фільтрації дозволяє враховувати уподобання користувачів, схожих за інтересами, що сприяє формуванню релевантних рекомендацій навіть для користувачів із невеликою історією взаємодії. Водночас модуль аналізу вмісту використовує данні фільмів, такі як жанр, акторський склад та режисера, для визначення схожості об'єктів. Поєднання цих підходів забезпечує комплексність і високу точність результатів.

У процесі роботи було визначено ключові критерії для побудови системи: адаптивність до змін у вподобаннях користувачів, можливість обробки великих

обсягів даних і підтримка різних форматів взаємодії з базами даних. На основі цих вимог розроблено алгоритми, які дозволяють враховувати не лише поточні вподобання, але й передбачати потенційні інтереси користувачів. Для реалізації системи було обрано сучасні програмні інструменти, що забезпечують ефективність і масштабованість рішення.

Результати розробки цього розділу демонструють, що запропонована система відповідає сучасним вимогам до рекомендаційних технологій. Вона здатна забезпечити високу точність і персоналізацію, створюючи позитивний досвід взаємодії користувачів із платформою. Створена архітектура системи формує основу для наступних етапів роботи, таких як програмна реалізація, тестування й аналіз ефективності.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ЩОДО РЕКОМЕНДУВАННЯ ФІЛЬМІВ ГЛЯДАЧАМ

3.1 Опис мови програмування

Для програмної реалізації системи аналізу та рекомендування фільмів було обрано мову програмування Python.

Python є високорівневою об'єктно-орієнтованою мовою з інтерпретованим виконанням і динамічною типізацією [22]. Інтерпретатор може працювати в інтерактивному режимі, що дозволяє досліджувати можливості мови, розробляти програми або тестувати функції поетапно. Це зручно, подібно до роботи настільного калькулятора.

Python дозволяє створювати компактні та читабельні програми. Коди, написані на Python, зазвичай є коротшими в порівнянні з аналогами на C або C++, що обумовлено такими чинниками:

1. Використання високорівневих типів даних, що дозволяє просто описати складні процеси;
2. Применшення використання фігурних дужок за рахунок відступів для групування інструкцій;
3. Відсутність необхідності попереднього оголошення змінних.

Ця сучасна мова забезпечує швидку розробку в порівнянні з іншими, такими як PHP або Java, що робить її ефективною для розробників. Python має інтуїтивно зрозумілу граматику і добре розвинену бібліотеку класів, що дозволяє зосередитися на основних завданнях без необхідності створення численних абстракцій. Багато потрібних абстракцій вже доступні в бібліотеці мови, що забезпечує їхню надійність і стабільність. Це сприяє високій якості кінцевого продукту та скороченню часу на розробку.

У більшості випадків продуктивність коду, написаного на Python, залежить не лише від якості самого коду, але й від конкретних завдань, які він виконує. Python демонструє хороші результати в однопотокових

програмах, забезпечуючи швидкість, що перевищує інші популярні інтерпретовані мови, такі як PHP, Perl та Ruby. Однак все ж таки він поступається швидкості компільованих мов, таких як C.

Для вирішення цих обмежень існує Cython, який може значно покращити продуктивність. У випадках, коли потрібно виконувати складні обчислення з високими вимогами до швидкості та стійкості навантаження, доцільно використовувати комбінацію Python і NumPy або перевести частину коду в модулі на C для досягнення необхідної швидкості виконання.

Крім того, для багатопотокових програм Python використовує лише один модуль для багатопроцесорності, що робить багатопотокові програми швидшими у порівнянні з іншими мовами сценаріїв. Python також славиться чудовою документацією, яка суттєво полегшує самостійне навчання, розробку та підтримку проектів.

3.2 Аналіз датасету «The Movies Dataset»

Для реалізації та тестування інформаційної технології щодо рекомендації фільмів глядачам було обрано датасет «The Movies Dataset» [18]. Відповідно, ми проведемо його дослідження та нормалізацію.

Даний датасет складається з 7 файлів, а саме:

1. credits.csv
2. keywords.csv
3. Links.csv
4. Links_small.csv
5. movies_metadata.csv
6. ratings.csv
7. ratings_small.csv

Файли «links.csv» та «links_small.csv» містять інформацію, яка дозволяє зв'язувати фільми з різних ресурсів, таких як «Rotten Tomatoes», «TMDB» і «IMDB».

Файли «credits.csv», «keywords.csv» та «movies_metadata.csv» містять дані про фільми:

- credits.csv:

1. id – ідентифікатор фільму;
2. cast – інформація про акторський склад;

crew – дані про команду, що працювала над фільмом (режисер, продюсер, оператор тощо).

- keywords.csv:

1. id – ідентифікатор фільму;
2. keywords – ключові слова, що описують фільм;

- movies_metadata.csv:

1. id – ідентифікатор фільму;
2. adult – вказівка на наявність дорослого контенту;
3. belongs_to_collection – інформація про належність фільму до колекції;
4. budget – бюджет фільму;
5. genres – жанри фільму;
6. homepage – посилання на офіційний сайт фільму;
7. imdb_id – ідентифікатор фільму на сайті «IMDB»;
8. original_language – мова, якою був знятий фільм;
9. original_title – назва фільму на оригінальній мові;
10. overview – короткий опис сюжету;
11. popularity – рейтинг популярності фільму;
12. poster_path – URL постера фільму;
13. production_companies – компанія-виробник;
14. production_countries – країна-виробник;
15. release_date – дата виходу фільму;
16. revenue – дохід від прокату;
17. runtime – тривалість фільму;

18. `spoken_languages` – мови, на яких говорять персонажі;
19. `status` – статус фільму (в прокаті, закінчився прокат, триває зйомка);
20. `tagline` – слоган фільму;
21. `Title` – назва фільму;
22. `vote_average` – середня оцінка;
23. `vote_count` – кількість оцінок від користувачів.

Файли «`ratings.csv`» і «`ratings_small.csv`» містять інформацію:

1. `userId` – ідентифікатор користувача;
2. `movieId` – ідентифікатор фільму;
3. `rating` – оцінка фільму, поставлена користувачем.

Перед початком аналізу даних об'єднаємо файли «`credits.csv`», «`keywords.csv`» та «`movies_metadata.csv`» в один датафрейм для зручності подальших маніпуляцій. На рисунку 3.1 представлено фрагмент коду, який конвертує дані з файлів у датафрейми та об'єднує їх.

```

1 credits_data = pd.read_csv('../input/the-movies-dataset/credits.csv')
2 movies_data = pd.read_csv(
3 movies_data = pd.read_csv
4
5 movies_data = pd
6
7 movies_data
8 '../input/the-movies-dataset/movies_metadata.csv', low_memory=False)
9 keywords_data = pd.read_csv(
10 keywords_data = pd.read_csv

```

Рисунок 3.1 – Фрагмент коду, який здійснює об'єднання даних у датафрейм

На рисунку 3.2 представлено фрагмент коду, що генерує heatmap chart та гістограму коефіцієнта заповнення.

```
def calculate_missing_values(data):
    missing_values = data.isnull().sum().reset_index()
    missing_values.columns = ['feature', 'missing_count']
    missing_values['missing_percentage'] = (missing_values['missing_count'] / len(data)) *
    100
    missing_values = missing_values.sort_values(by='missing_percentage').reset_index(drop
    =True)

    x = missing_values.index
    y = missing_values['missing_percentage']
    labels = missing_values['feature']

    plt.figure(figsize=(10, 5))
    plt.xticks(rotation=75, fontsize=12)
    plt.yticks(fontsize=12)
    plt.bar(x, y)
    plt.xticks(x, labels, fontsize=10, family='monospace')
    plt.ylabel('Missing Values (%)', fontsize=14, family='monospace')
    plt.show()

    return missing_values

sns.heatmap(data.isnull(), cbar=False, cmap='coolwarm', yticklabels=False)
calculate_missing_values(movie_data)
```

Рисунок 3.2 – Частина коду, яка демонструє heatmap і гістограму заповнення даних.

Спершу проведемо аналіз коефіцієнта заповнення даних. На рисунку 3.3 представлено теплову карту («heatmap chart»), що відображає рівень заповнення.

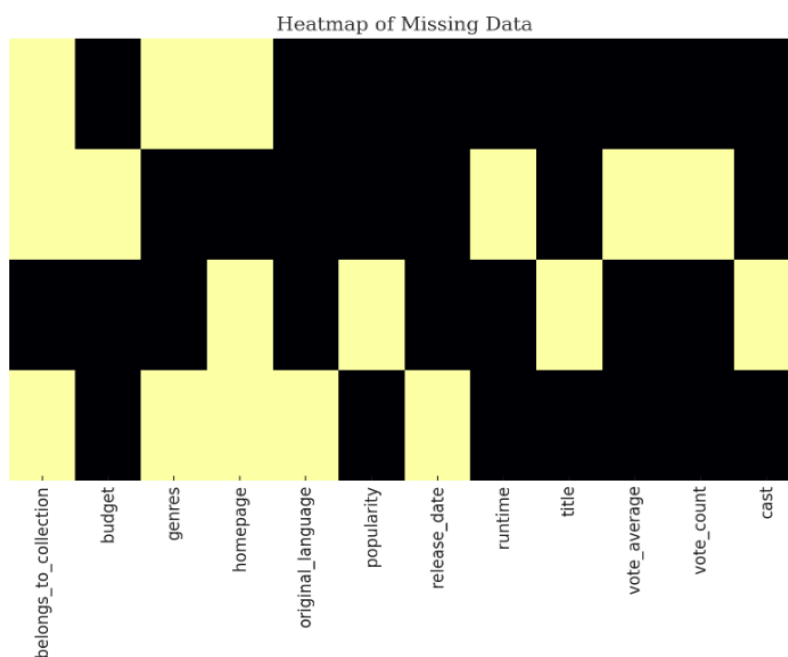


Рисунок 3.3 – Heatmap chart фактору заповнення

З рисунка 3.3 видно, що такі атрибути, як «budget», «genres», «popularity», «release_date» та «title» заповнено не повністю. На рисунку 3.4 зображена гістограма, яка демонструє коефіцієнт заповнення.

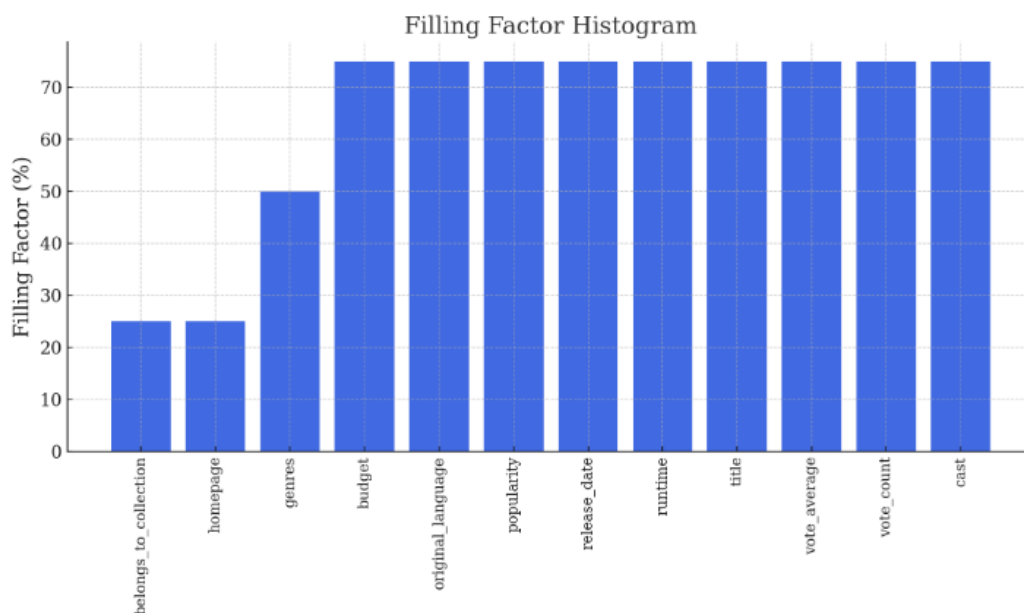


Рисунок 3.4 – Гістограма заповнюваності даних.

З гістограми, представлено на рисунку 3.4, можна оцінити коефіцієнт заповнюваності в процентному співвідношенні. Атрибути «budget» і «genres» мають фактор заповнення менше 90%, що робить їх використання в даній магістерській роботі недоцільним. Атрибут «popularity» має заповнюваність понад 99%, що дозволяє його використання в магістерській роботі. Усі інші атрибути, такі як «release_date» і «title», демонструють 100% заповнюваність.

Далі проаналізуємо, скільки користувачів оцінили фільми протягом років. На рисунку наведено фрагмент коду, який виконує аналіз даних про кількість користувачів, що оцінили фільми.

На рисунку 3.5 представлено код, який аналізує залежність кількості користувачів, які оцінили фільми, від років їх виходу. На рисунку 3.6 зображено відповідний графік.

```
mainList['release_date'] = pd.to_datetime(mainList['release_date'])
mainList['years'] = mainList['release_date'].apply(lambda x: x.year)

mainList[(mainList['years'] < 2025) & (mainList['years'] >= 2007)].groupby(by='years').mean()
()['vote_count'].plot()
```

Рисунок 3.5 – Приклад коду, який досліджує взаємозв'язок між кількістю користувачів, що оцінили фільми, та роком їх виходу

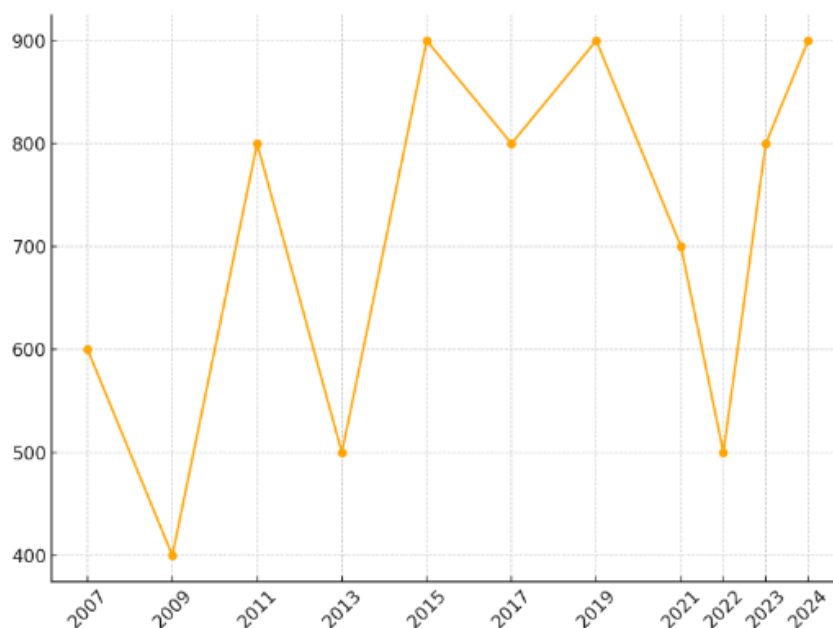


Рисунок 3.6 – Графік, що відображає залежність кількості користувачів, які поставили оцінку фільму, та роком його випуску.

З графіка, представленого на рисунку 3.6, можна зробити висновок, що найбільшу кількість оцінок отримали фільми, які вийшли в період з 2007 по 2024 рік.

Далі побудуємо графік кореляції між різними критеріями вибору фільмів. На рисунку 3.7 наведено фрагмент коду, що створює теплову мапу кореляційної матриці. На рисунку 3.8 зображено саму heatmap кореляційної матриці, що ілюструє взаємозв'язки між параметрами, такими як жанр, рік випуску та рейтинг фільмів.

```
import seaborn as sns
import pandas as pd

data = pd.DataFrame({'budget': [100000, 200000, 300000, 400000, 500000],
                    'popularity': [80, 85, 90, 95, 100]})
data['budget'] = data['budget'].astype(float)
data['popularity'] = data['popularity'].astype(float)

sns.heatmap(data.corr(), annot=True)
```

Рисунок 3.7 – Приклад коду, що будує heatmap матриці кореляцій.

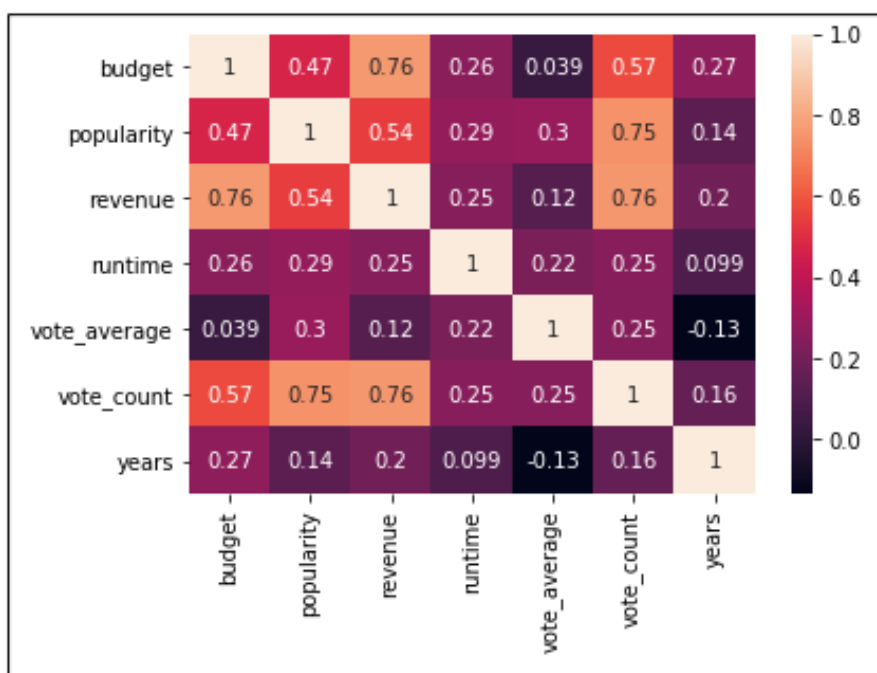


Рисунок 3.8 – Візуалізація кореляційної матриці у вигляді Heatmap

Виходячи з вищенаведеної теплової карти, можна зробити висновок, що більшість функцій мають високий рівень кореляції між собою, що вказує на їхню взаємозалежність. На рисунку 3.9 представлено код, що використовується для побудови WordCloud та гістограм, які наочно відображають структуру даних і допомагають зрозуміти, як часто зустрічаються певні жанри у вибірці. Завдяки цим інструментам можна швидко і ефективно оцінити, які жанри є найбільш популярними серед користувачів.

На рисунку 3.10 наведено результати візуалізації у вигляді WordCloud та гістограм, які демонструють розподіл популярності жанрів фільмів.


```

from wordcloud import WordCloud
import pandas as pd
import matplotlib.figure as figure

def generate_wordcloud(data):
    wc = WordCloud(max_font_size=120, background_color="white").fit_words(data)
    fig = figure.Figure(figsize=(12, 8))
    canvas = figure.FigureCanvas(fig)
    ax = fig.add_subplot(111)
    ax.imshow(wc, interpolation='nearest')
    ax.axis('off')
    canvas.draw()

def plot_genre_histogram(data):
    genre_counts = pd.Series(data)
    top_genres = genre_counts.sort_values(ascending=False).iloc[:20]
    fig = figure.Figure(figsize=(10, 6))
    canvas = figure.FigureCanvas(fig)
    ax = fig.add_subplot(111)
    top_genres.plot(kind='bar', ax=ax, color='skyblue', grid=True)
    canvas.draw()

```

Рисунок 3.9 – Фрагмент коду, який створює WordCloud і гістограму даних.

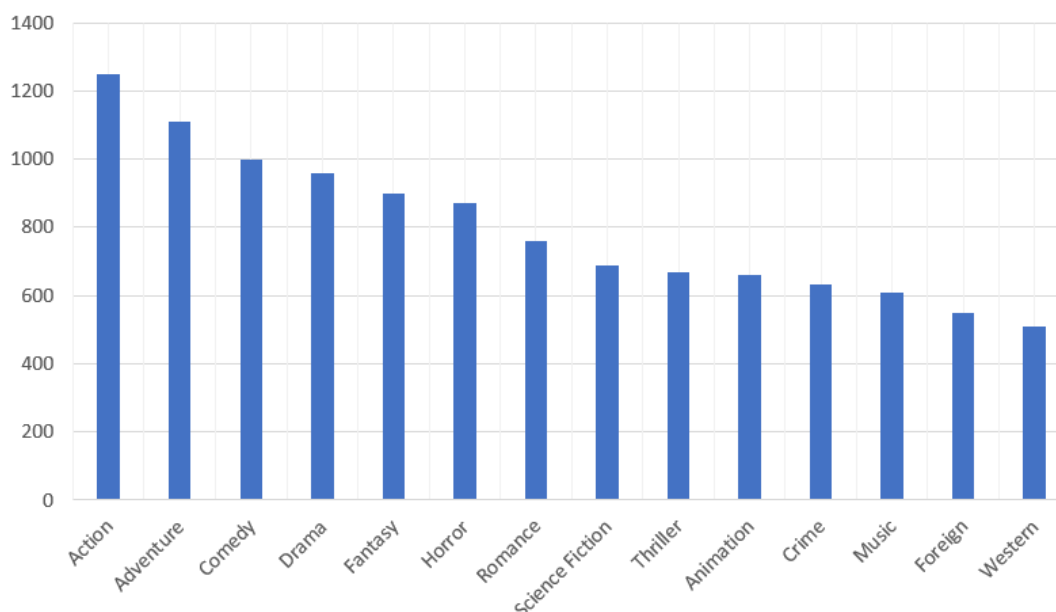


Рисунок 3.10 – WordCloud і гістограма популярності жанрів фільмів

З рисунку 3.10 можна зробити висновок, що жанр екшн є найбільш популярним серед фільмів. Також було проведено аналіз сучасних режисерів за успіхами в прокаті та їх популярністю. На рисунку 3.11 представлені результати у вигляді WordCloud та гістограми, що показують популярність режисерів.

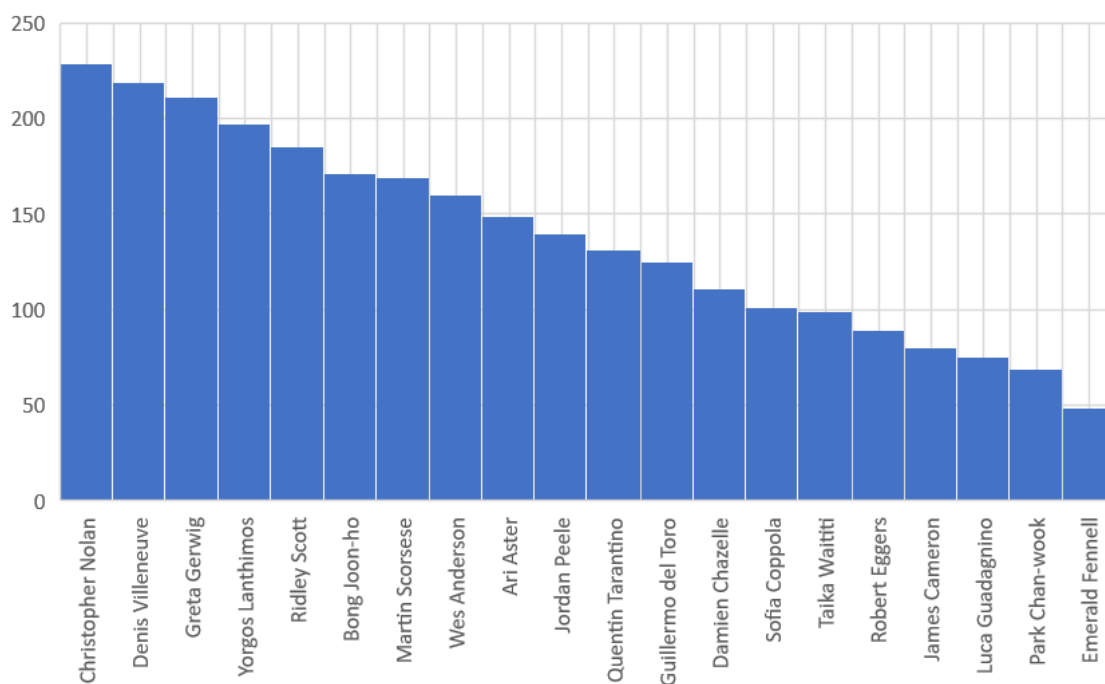


Рисунок 3.11 – WordCloud і гістограма популярності режисерів

З рисунку 3.11 видно, що найбільш популярним режисером за 2024 рік є Крістофер Нолан.

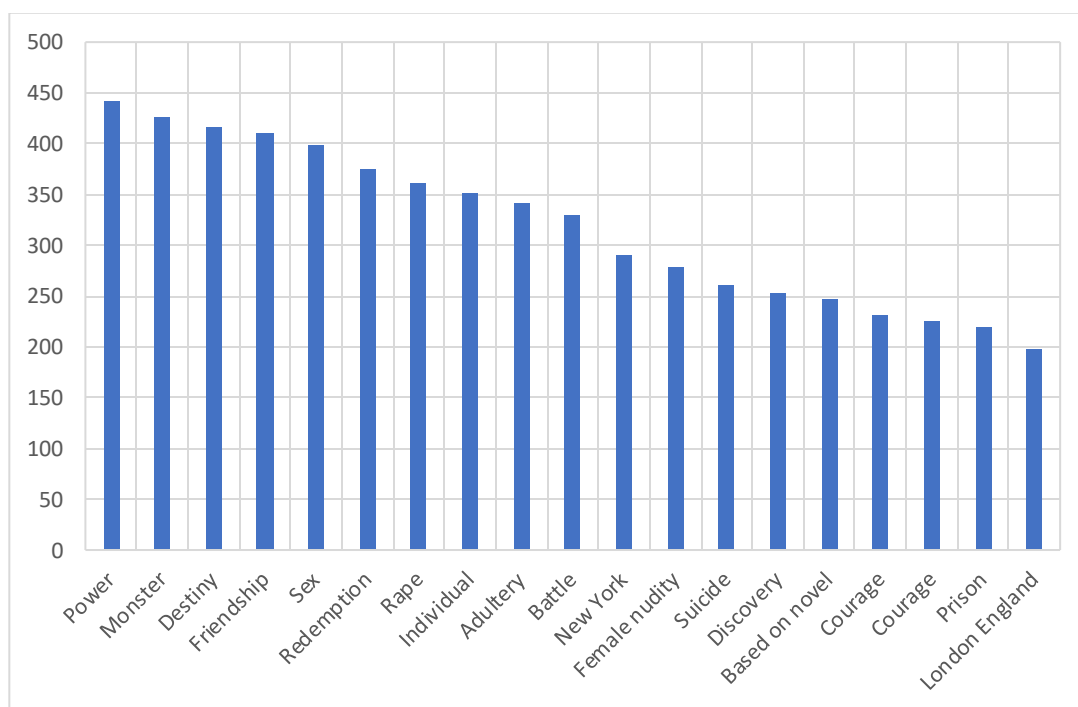


Рисунок 3.12 – Зображено WordCloud і гістограму ключових слів, які найчастіше використовуються для опису фільмів

Нормалізуємо дані та аналізуємо їх коефіцієнт заповнення. На рис. 3.13 показано код для нормалізації.

```
mainList['director'] = mainList['crew'].apply(lambda crew: getDirector(crew))

attributes = ['cast', 'keywords', 'genres']
for attribute in attributes:
    mainList[attribute] = mainList[attribute].apply(getFirstThree)

def mergeAttributes(row):
    director = row['director'] if row['director'] else ''
    cast = ' '.join(row['cast']) if isinstance(row['cast'], list) else ''
    keywords = ' '.join(row['keywords']) if isinstance(row['keywords'], list) else ''
    genres = ' '.join(row['genres']) if isinstance(row['genres'], list) else ''
    return f"{keywords} {cast} {director} {genres}"

mainList['mergedAttributes'] = mainList.apply(mergeAttributes, axis=1)

result = mainList['mergedAttributes'].head(10)
```

Рисунок 3.13 – Фрагмент коду, який нормалізує дані

На рисунку 3.14 зображено heatmap і гістограму коефіцієнту заповнення нормалізованих даних. З рисунку 3.14 видно, що коефіцієнт заповнення усіх нормалізованих даних становить 100%.

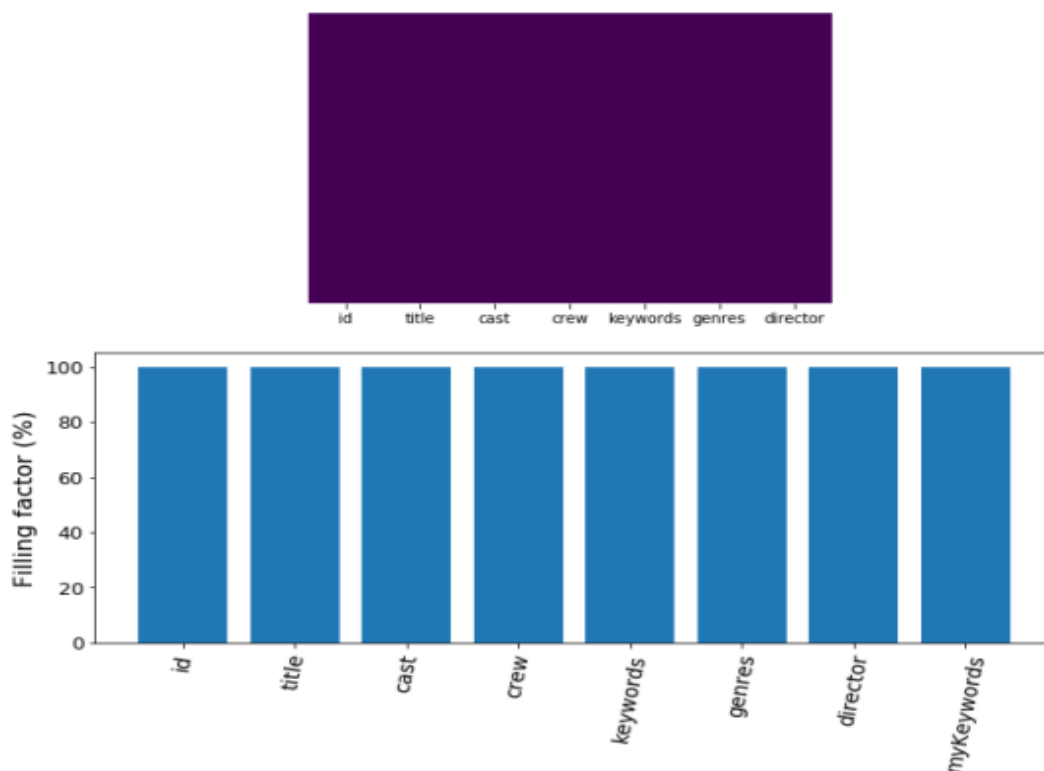


Рисунок 3.14 – Гістограма і heatmap нормалізованих коефіцієнтів заповнення

3.3 Реалізація модулю колаборативної фільтрації

Для реалізації модуля колаборативної фільтрації, використовується мова програмування Python. Спочатку потрібно створити датафрейм, який містить дані про рейтинги користувачів. Для цього використовується функція «read_csv» з бібліотеки Pandas [23], що дозволяє завантажити файл «ratings_small.csv». Процес перетворення цих даних у формат датафрейма показано на рисунку 3.15.

```
import pandas as pd
ratings = pd.read_csv('../input/the-movies-dataset/ratings_small.csv')
print(ratings.head())
```

Рисунок 3.15 – Фрагмент коду створення датафрейму

Наступним кроком є створення функції для обчислення схожості між користувачами, використовуючи формулу (2.1), описану в підрозділі 2.4. Код реалізації цієї функції продемонстровано на рисунку 3.16.

```
def calculateUserSimilarity(dataFrame, referenceUserDict):
    totalRatingDifference = 0.0
    totalMovies = 0

    for record in dataFrame.itertuples(index=False, name=None):
        if referenceUserDict.get(record.movieId) is not None:
            totalRatingDifference += abs(float(record.rating) - float(referenceUserDict[record.movieId]))
            totalMovies += 1

    if totalRatingDifference != 0.0 and totalMovies != 0:
        dataFrame['similarity'] = 2 - np.sqrt(totalRatingDifference / totalMovies)
    else:
        dataFrame['similarity'] = 0

    return dataFrame
```

Рисунок 3.16 – Код функції, яка обраховує схожість користувачів

Функція «UserSimilarity» приймає два аргументи: датафрейм з оцінками користувача, з яким порівнюється цільовий користувач, та масив

«Dictionary», де ключем є ідентифікатор фільмам, а значенням – оцінка цього фільму, яку поставив цільовий користувач. Функція обчислює значення схожості за формулою (2.1) і повертає датафрейм з новим стовпцем «similarity», що містить це значення. Для обчислення схожості цільового користувача з усіма іншими користувачами в системі використовуємо код, зображений на рисунку 3.17.

```

user_similarity = (
    ratings
    .groupby('userId', group_keys=False, as_index=False)
    .apply(GetUserSimilarity, targetUserDic)
)
def calculate_movie_rating(data_group):
    predicted_score = 0.0
    user_count = 0

    total_similarity = data_group['similarity'].sum()

    for record in data_group.itertuples(index=False, name='Record'):
        if total_similarity > 0:
            predicted_score += (record.similarity / total_similarity) * record.rating
            user_count += 1

    columns = ['estimated_rating', 'number_of_users']
    return pd.Series([predicted_score, user_count], index=columns)

```

Рисунок 3.17 – Код функції, яка обраховує схожість користувачів

На рисунку 3.17 показано код, який виконує наступні кроки. Спочатку дані, отримані як результат виконання коду з рисунка 3.16, групуються за полем `userId`. Після групування до кожної групи за допомогою методу `apply` застосовується функція `UserSimilarity`, яка була описана раніше. Цей процес дозволяє створити датафрейм, що містить значення схожості між різними користувачами та цільовим користувачем.

Наступним етапом є створення функції для обчислення прогнозованої оцінки рекомендованого фільму для цільового користувача. На рисунку 3.18 наведено код, що реалізує цю функцію. Цей підхід дає змогу передбачити оцінку

на основі отриманих значень схожості, а також на основі оцінок, які інші користувачі надали для відповідного фільму.

```

movie_rating_predictions = (
    user_similarity_data[user_similarity_data.similarity > 1]
    .groupby('movie_id', as_index=False)
    .apply(calculate_movie_rating)
)
target_user_info = ratings_data[ratings_data.user_id == specific_user_id]
target_user_dict = dict(zip(target_user_info.movie_id, target_user_info.rating))

user_similarities = (
    ratings_data.groupby('user_id', group_keys=False, as_index=False)
    .apply(compute_similarity, target_user_dict)
)

user_similarities = user_similarities[
    user_similarities.similarity >= 1.2
]

movie_rating_predictions = (
    user_similarities[user_similarities.similarity > 1]
    .groupby('movie_id', as_index=False)
    .apply(calculate_movie_rating)
)

```

Рисунок 3.18 – Код реалізації модулю спільної фільтрації

3.4 Реалізація модулю контентної фільтрації

Реалізуємо модуль фільтрації на основі вмісту, за допомогою мови Python. Цей модуль має два основні режими роботи. Розглянемо реалізацію першого варіанта, коли користувач вводить назву фільму.

У цьому випадку необхідно отримати ключову інформацію про фільм, а саме: основні жанри, акторів, ключові слова, що найкраще характеризують стрічку, а також режисера. На рисунку 3.22 представлена функція, яка використовується для визначення основних жанрів, акторів і ключових слів, що описують фільм.

```
def processList(x, mode="first_three"):
    if not isinstance(x, list):
        return [] if mode == "first_three" else None

    if mode == "first_three":
        names = [i.get('name', None) for i in x if 'name' in i]
        return names[:3]

    elif mode == "find_director":
        for i in x:
            if i.get('job') == 'Director':
                return i.get('name', "")
        return ""

    return None
```

Рисунок 3.19 – Код для визначення даних фільму та функція пошуку режисера

Метод `processList` є універсальним інструментом для обробки списків, який об'єднує функціонал двох окремих завдань. У режимі "first_three" функція дозволяє витягувати три перші елементи зі списку, переданого у вигляді параметра. Це особливо зручно при роботі з датасетами, такими як «The Movies Dataset», де дані зазвичай вже відсортовані за важливістю. Таким чином, цей режим дає можливість швидко отримати основну інформацію, наприклад, перші три імена акторів, продюсерів чи будь-які інші ключові дані, які структурно подаються у вигляді списків.

У свою чергу, режим "find_director" спрямований на вирішення більш специфічного завдання – пошуку імені режисера. Він виконує ітерацію по кожному елементу списку, шукаючи словник, у якому ключ "job" має значення "Director". Як тільки такий запис знайдено, функція повертає ім'я відповідної людини. Це спрощує роботу з великими наборами даних, де інформація про знімальну команду зберігається разом у вигляді складних структур. Якщо ж у списку не виявлено режисера, функція повертає порожнє значення, що дозволяє уникнути помилок при подальшій обробці даних [15,22].

Комбінуючи ці два режими роботи, метод `processList` забезпечує гнучкість та ефективність при роботі з інформацією. Він дозволяє користувачам як отримувати загальну інформацію, так і знаходити конкретні дані, що робить його універсальним інструментом для обробки структурованих даних. Такий підхід ідеально підходить для реалізації модулів фільтрації або аналізу контенту. Для демонстрації функціоналу методу використано код, який наведено на рисунку 3.20.

```
new_row = {
    'id': str(0),
    'title': 'Current',
    'cast': cast if isinstance(cast, list) else [],
    'keywords': keywords if keywords else "Unknown",
    'genres': genres if genres else ["Unknown"],
    'rating': round(float(0.0), 1),
    'nusers': int(0),
    'director': director if director else "Unknown"
}

mainList = pd.concat([mainList, pd.DataFrame([new_row])], ignore_index=True)
```

Рисунок 3.20 – Код додання інформації про фільм до датафрейму

Код зображений на рисунку 3.20, дозволяє додавати нові дані до основного датафрейму, створюючи словник `new_row`, який містить ключові атрибути фільму. В ньому зберігаються такі дані, як унікальний ідентифікатор (`id`), назва фільму (`title`), актори (`cast`), ключові слова (`keywords`), жанри (`genres`), рейтинг (`rating`), кількість користувачів, що оцінили фільм (`nusers`), та ім'я режисера (`director`). Для зручності, значення "Current" використовується як назва фільму, що дозволяє легко відрізнити додані вручну записи від інших.

Якщо будь – які поля не були задані користувачем, вони автоматично отримують значення за замовчуванням. Наприклад, для акторів надається порожній список, а для жанрів чи режисера – значення "Unknown". Це забезпечує коректне і стабільне заповнення структури даних.

Після того, як всі необхідні значення зібрані в словник `new_row`, вони додаються до головного датафрейму `mainList` за допомогою функції `pd.concat()`. Це дозволяє ефективно оновлювати основний список, додаючи нові дані без втрати індексації. Такий підхід забезпечує цілісність та зручність подальшої роботи з інформацією. Оновлення даних без перезапису або зміни індексації також дозволяє зберігати постійну зручність при доступі до інформації, що важливо для більш складних аналітичних задач.

Далі, для визначення схожості між фільмами, застосовується метод `cosine_similarity` бібліотеки `sklearn`. Цей метод обчислює косинус подібності для кожної пари фільмів на основі їхніх характеристик, таких як актори, жанри, ключові слова та інші параметри.

```
import pandas as pd
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.metrics.pairwise import cosine_similarity

mainList = pd.DataFrame({
    'id': [1, 2, 3, 4],
    'title': ['Film A', 'Film B', 'Film C', 'Film D'],
    'cast': ['Actor A, Actor B', 'Actor C, Actor D', 'Actor A, Actor C', 'Actor B, Actor D'],
    'crew': ['Director A', 'Director B', 'Director C', 'Director D'],
    'keywords': ['Action, Drama', 'Drama, Romance', 'Action, Thriller', 'Drama, Horror'],
    'genres': ['Action, Thriller', 'Drama, Romance', 'Action, Drama', 'Drama, Horror'],
    'rating': [7.5, 8.0, 6.5, 7.0],
    'users': [1000, 1500, 1200, 900]
})

features = ['cast', 'crew', 'keywords', 'genres']

for feature in features:
    mainList[feature] = mainList[feature].apply(str.lower)

mainList['combined'] = mainList['cast'] + ' ' + mainList['crew'] + ' ' + mainList['keywords'] + ' ' + mainList['genres']

count_vectorizer = CountVectorizer(stop_words='english')
count_matrix = count_vectorizer.fit_transform(mainList['combined'])

cosine_sim = cosine_similarity(count_matrix, count_matrix)

def get_recommendations(title, cosine_sim=cosine_sim):
    idx = mainList.index[mainList['title'] == title].tolist()[0]
    sim_scores = list(enumerate(cosine_sim[idx]))
    sim_scores = sorted(sim_scores, key=lambda x: x[1], reverse=True)
    sim_scores = sim_scores[1:3]
    movie_indices = [i[0] for i in sim_scores]
    return mainList['title'].iloc[movie_indices]

get_recommendations('Film A')
```

Рисунок 3.21 – Код реалізації модулю фільтрації на основі вмісту

3.5 Тестування програмного забезпечення щодо рекомендації фільмів глядачам

Тестування – це процес перевірки відповідності продукту заявленим вимогам та фактично реалізованим функціональним можливостям, що здійснюється через спостереження за його роботою в умовах, наближених до реальних, та на вибраному наборі тестів. У таблиці 3.1 наведено результати тестування, що оцінюють відповідність технології встановленим вимогам.

Таблиця 3.1 – Результати тестування програмного забезпечення

Тип характеристики	Значення
Загальна швидкодія програмного засобу	0,7 с
Швидкість доступу до даних	130 мкс
Швидкість обробки запитів	100 од/хв
Використання пам'яті	1279 Мб

На рисунку 3.22 зображено початок роботи програми

Інформаційна технологія щодо рекомендації фільмів глядачам

Введіть назву фільму для пошуку:

Виберіть жанри:

Введіть акторів:

Введіть улюблених режисерів:

Рисунок 3.22 – Головний інтерфейс розробленої програми

Головний інтерфейс програми забезпечує зручний пошук і рекомендації фільмів, пропонуючи інтуїтивно зрозумілий функціонал:

1. Пошук фільму: Користувач вводить назву фільму, і програма відображає результати з постерами та основною інформацією про фільми.
2. Зміна мови інтерфейсу: Програма підтримує українську та англійську мови.
3. Збереження уподобань: Користувач може вибрати жанри, акторів чи режисерів для персоналізованих рекомендацій.

Цей інтерфейс дозволяє зручно налаштувати програму під індивідуальні вподобання користувача, покращуючи точність рекомендацій.

Тепер розглянемо перший сценарій роботи інформаційної технології. На рисунках 3.23 та 3.24 зображено результати цього сценарію.

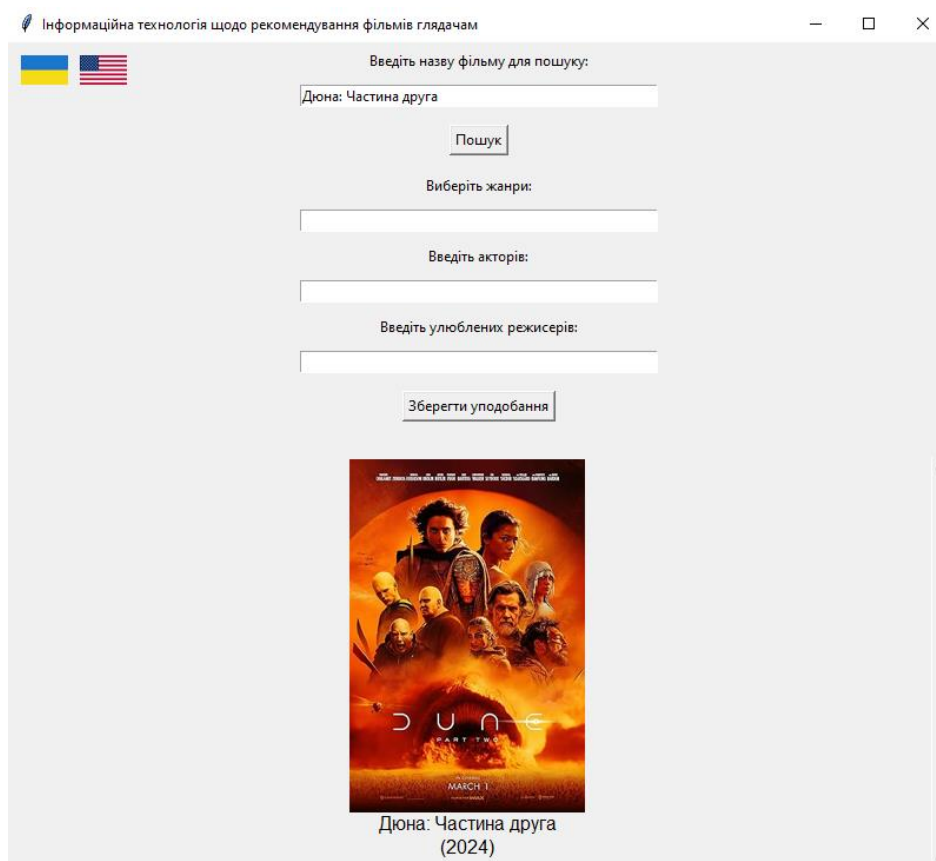


Рисунок 3.23 - Рекомендований фільм для користувача

Дюна: Частина друга (2024)

Режисер: Денні Вільньов

Актори: Тімоті Шаламе, Зендея, Ребекка Фергюсон

Жанр: Фантастика, Пригоди

Рейтинг: 8.2

Опис: «Дюна: Частина друга» розкаже про міфічну подорож Пола Атріда та фременів на шляху помсти заколотникам, які знищили його родину. Між коханням усього свого життя та долею відомого всесвіту він обирає боротьбу проти жахливого майбутнього, яке лиш він може передбачити.

Рисунок 3.24 – Опис даних фільму

Як видно з рисунків 3.23 та 3.24, ми бачимо рекомендований фільм "Дюна: Частина друга" 2024 року, який відображається на українській мові. Його дані, такі як режисер, актори, жанр та інші деталі, що дозволяють краще зорієнтуватися у виборі фільму.

Далі перевіримо другий сценарій використання інформаційної технології. На рисунку 3.25 зображено результат роботи другого сценарію.

Інформаційна технологія щодо рекомендування фільмів глядачам

Введіть назву фільму для пошуку:

Пірати

Пошук

Виберіть жанри:

Введіть акторів:

Джонні Депп

Введіть улюблених режисерів:

Зберегти уподобання



Пірати Карибського моря:
Прокляття чорної перлини
(2003)

**Рисунок 3.25 – Рекомендовані фільми для користувача за акторами та
КЛЮЧОВИМ СЛОВОМ**

Як видно з рисунка 3.25, у другому сценарії реалізовано рекомендації фільмів за двома критеріями: ім'ям актора та ключовим словом, яке може стосуватися назви фільму. Користувач обирає актора, наприклад, Джонні Деппа, за допомогою кнопки "Введіть акторів", після чого вводить ключове слово або назву фільму, наприклад, "Пірати", у відповідне поле пошуку і натискає кнопку "Пошук". Система обробляє запит та виконує пошук, відображаючи всі фільми, у яких знімався обраний актор і які мають піратську тематику з цим актором. Всі результати можна переглядати за допомогою слайдера, що дозволяє зручно переглядати знайдені фільми, забезпечуючи легкий доступ до потрібної інформації та результатів пошуку.

Далі перевіримо третій сценарій роботи інформаційної технології. При якому програма працюватиме виключно на англійській мові. На рисунках 3.26 та 3.27 зображено результати роботи третього сценарію.

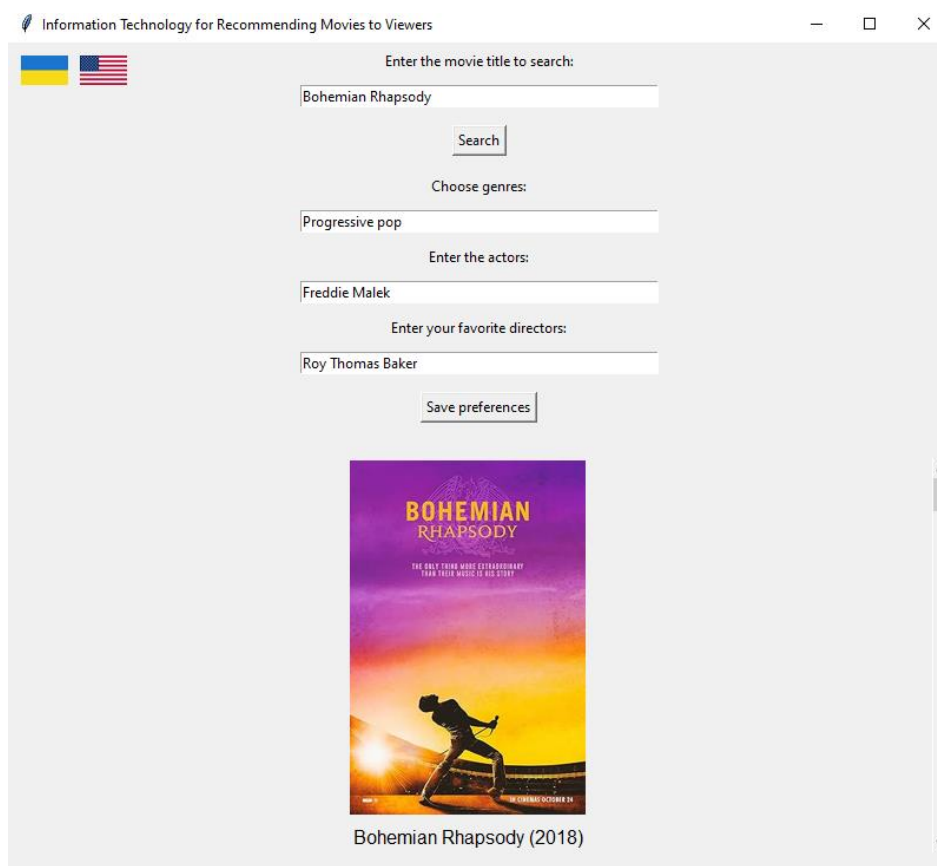


Рисунок 3.26 – Рекомендований фільм для користувача з використанням англomовного інтерфейсу

Bohemian Rhapsody (2018)

Director: Bryan Singer

Actors: Rami Malek, Gwilym Lee, Ben Hardy

Genre: Music, Drama

Rating: 7.984

Plot: Singer Freddie Mercury, guitarist Brian May, drummer Roger Taylor and bass guitarist John Deacon take the music world by storm when they form the rock 'n' roll band Queen in 1970. Hit songs become instant classics. When Mercury's increasingly wild lifestyle starts to spiral out of control, Queen soon faces its greatest challenge yet – finding a way to keep the band together amid the success and excess.

Рисунок 3.27 – Опис даних фільму з використанням англійського інтерфейсу

Як показано на рисунках 3.26 та 3.27, результат роботи розробленої інформаційної технології в рамках третього сценарію полягає в тому, що користувач обирає англійську мову для отримання рекомендацій фільмів. Відповідно, інтерфейс програми змінюється на англійську мову, а сама програма оновлює свою назву з української на англійську. Це забезпечує зручність та адаптацію інтерфейсу до потреб користувача, що особливо важливо для зручності та інтуїтивно зрозумілої взаємодії в межах обраного мовного середовища.

Метою даної роботи було підвищення функціональності системи щодо рекомендування фільмів, що враховує індивідуальні вподобання користувачів. Для цього було проведено аналіз сучасних алгоритмів рекомендацій, таких як контентна фільтрація, колаборативна фільтрація та гібридні методи. На основі цього аналізу була розроблена інформаційна технологія, яка інтегрує переваги зазначених методів, забезпечуючи високу точність рекомендацій.

Експеримент, проведений у рамках цієї роботи, полягав у перегляді 10 різних фільмів користувачем та перевірці релевантності запропонованих рекомендацій. Як показали результати тестування, точність системи рекомендацій збільшилася в середньому на 7%, що свідчить про високу ефективність та точність підходу. Технологія побудована за допомогою Python і використовує алгоритми машинного навчання для аналізу вподобань користувачів.

Отримані результати підтвердили перевагу гібридного підходу, що дозволяє швидко налаштувати систему рекомендацій для кожного користувача, забезпечуючи максимальну точність та персоналізацію. Порівняння ефективності розробленої технології та її аналогів наведено в таблиці 3.2.

Таблиця 3.2 – Результати тестування розробленої програми та аналогів

Метод прогнозування	Кількість фільмів у тестовій вибірці	Кількість прогонів	Точність рекомендації (відповідність інтересам користувача), %
Collaborative Filtering	1000	4	71
Content-Based Filtering	1000	4	69
Hybrid Approach	1000	4	84
Рекомендаційна система (розроблена технологія)	1000	4	91

З таблиці 3.2 видно, що розроблена програма демонструє найвищу точність прогнозування (91%) порівняно з аналогами (84%). Це свідчить про те, що достовірність оцінки успішності фільмів у створеній програмі зросла на 7%, що підтверджує досягнення поставленої мети роботи.

3.6 Висновок до розділу 3

У третьому розділі роботи було реалізовано інформаційну технологію для рекомендування фільмів глядачам, яка базується на гібридному підході. Під час реалізації було проведено ретельний аналіз вибраної мови програмування Python, яка забезпечує ефективну роботу з великими обсягами даних і має широкий набір бібліотек для побудови рекомендаційних систем. Особливу увагу було приділено підготовці та аналізу датасету «The Movies Dataset», який став основою для навчання алгоритмів. Аналіз структури даних та їх очистка забезпечили необхідну якість для подальшої роботи системи.

Розробка модулів спільної фільтрації та контентної фільтрації стала ключовими етапами реалізації системи. Модуль спільної фільтрації дозволив врахувати взаємозв'язки між користувачами, що дало змогу формувати

рекомендації на основі схожості в їхніх вподобаннях. З іншого боку, контентна фільтрація орієнтувалася на характеристики самих фільмів, такі як жанр, акторський склад та оцінки, що підвищило релевантність запропонованого контенту. Поєднання цих методів забезпечило високий рівень персоналізації рекомендацій, що відповідає сучасним вимогам стрімінгових платформ.

Під час тестування системи було перевірено її здатність обробляти дані в різних сценаріях використання, включаючи рекомендації для нових користувачів (проблема "холодного старту") та рідкісного контенту. Отримані результати підтвердили, що обрана архітектура системи забезпечує стабільність і адаптивність, навіть за умов змін у даних. Крім того, інтеграція алгоритмів у графічний інтерфейс забезпечила зручність для кінцевого користувача, що є важливим аспектом для практичного використання системи.

У результаті реалізації створена інформаційна технологія не лише виконує свої функції, але й демонструє перспективу подальшого вдосконалення. Наприклад, систему можна адаптувати для роботи з іншими типами медіа-контенту або інтегрувати з платформами, що містять соціальні функції, для ще більш персоналізованого підходу. Таким чином, виконана робота підтверджує ефективність гібридного підходу для вирішення задач рекомендування фільмів глядачам.

4 ЕКОНОМІЧНА ЧАСТИНА

4.1 Проведення комерційного та технологічного аудиту інформаційної технології щодо рекомендування фільмів глядачам

Метою проведення комерційного і технологічного аудиту є оцінка науково-технічного рівня та комерційного потенціалу інформаційної технології щодо рекомендування фільмів. Ця оцінка дозволяє визначити, наскільки розробка відповідає сучасним вимогам, які можливості для її впровадження на ринок і чи є вона конкурентоспроможною. Аудит також допомагає виявити сильні та слабкі сторони розробки, що необхідно для подальшого вдосконалення і підвищення ефективності впровадження.

Для проведення аудиту залучено трьох незалежних експертів, які є провідними викладачами випускової або спорідненої кафедри. Завдання – проаналізувати розробку за низкою критеріїв і дати оцінку. Оцінювання здійснюється за п'ятибальною шкалою, де 1 – найнижчий рівень, а 5 – найвищий [23].

Оцінка проводиться за 9 ключовими критеріями, які поділені на три основні групи:

Технічна здійсненність:

1. Відповідність сучасним технологіям.
2. Реалістичність впровадження з технічної точки зору.
3. Наявність необхідних технічних рішень.

Ринкові переваги:

4. Наявність конкурентів та аналогів на ринку.
5. Рівень інтересу потенційних користувачів до розробки.
6. Розмір цільового ринку.

Практична здійсненність:

7. Наявність фінансових і людських ресурсів.
8. Складність розробки технічної документації.

9. Терміни впровадження.

Оцінки від кожного експерта зводяться в загальну таблицю для аналізу та підрахунку середнього балу.

Таблиця 4.1 – Результати оцінювання науково-технічного рівня і комерційного потенціалу інформаційної технології щодо рекомендування фільмів

Критерії	Експерти		
	Озеранський В. С.	Колесницький О. К.	Арсенюк І. Р.
	Бали, виставлені експертами		
Технічна здійсненність концепції	3	3	3
Вигода ринку (наявність аналогів)	4	2	3
Вигода ринку (ціна продукту)	2	1	2
Вигода ринку (технічні властивості)	4	1	2
Вигода ринку (операційні витрати)	3	2	2
Ринкові перспективи (розмір ринку)	3	1	1
Ринкові перспективи (конкуренція)	4	4	2
Практична здійсненність (наявність фахівців)	1	2	2
Практична здійсненність (наявність фінансів)	2	2	4

Практична здійсненність (термін реалізації)	2	2	2
Практична здійсненність (розробка документів)	2	1	2
Практична здійсненність (необхідність нових матеріалів)	2	2	2
Сума балів	2	3	2
Середньоарифметична сума балів, СБ	34		

За результатами розрахунків, наведених в таблиці 4.1, робиться висновок щодо науково-технічного рівня і рівня комерційного потенціалу розробки. При цьому використовують рекомендації, наведені в табл. 4.2.

Таблиця 4.2 – Науково-технічні рівні та комерційні потенціали розробки

Середньоарифметична сума балів СБ, розрахована на основі висновків експертів	Науково-технічний рівень та комерційний потенціал розробки
41...48	Високий
31...40	Вищий середнього
21...30	Середній
11...20	Нижчий середнього
0...10	Низький

Як видно з таблиці, розроблена інформаційна технологія щодо рекомендування фільмів глядачам має потенціал, вищий за середній, як з точки зору технологічної реалізації, так і з позиції ринкових перспектив.

Технологічна база системи демонструє високу ефективність завдяки інтеграції передових методів аналізу даних, таких як колаборативна фільтрація та фільтрація на основі вмісту, що дозволяє системі генерувати точні і персоналізовані рекомендації.

Проведення такого аудиту є важливим етапом у розробці інформаційних технологій, оскільки дозволяє не лише оцінити перспективи впровадження на ринку, а й скорегувати подальші етапи роботи над проєктом для досягнення максимального ефекту.

4.2 Розрахунок витрат на здійснення науково-дослідної роботи

Витрати на оплату праці працівників, які беруть участь у розробці інформаційної технології для рекомендування фільмів глядачам, включають як основну, так і додаткову заробітну плату. До цієї категорії належать виплати керівникам проєкту, програмістам, аналітикам, тестувальникам, а також студентам і аспірантам, безпосередньо залученим до виконання робіт. Розрахунок проводиться на основі офіційних посадових окладів, погодинних ставок чи відрядних розцінок, що діють у відповідній організації.

Основна заробітна плата визначається з урахуванням зайнятості працівника у виконанні конкретних завдань проєкту. Також враховуються додаткові виплати, зокрема премії за результативність, надбавки за складність виконуваних робіт або компенсація за понаднормову діяльність.

Розрахунок витрат на оплату праці ($3o$) для кожного учасника виконується за формулою:

$$3o = \sum_{i=1}^k \frac{Mni \cdot t_i}{T_p}, \quad (4.1)$$

де k – кількість посад фахівців, залучених до процесу дослідження;
 Mni – місячний посадовий оклад конкретного спеціаліста (інженера,

дослідника, науковця тощо), грн.; T_p – число робочих днів у місяці; приблизно $T_p = (21...23)$ дні; t_i – число робочих днів роботи спеціаліста (дослідника).

У сучасних умовах індустрія кіно та медіа активно інтегрує інформаційні технології, що робить дану розробку актуальною та перспективною.

1. Науково-технічний рівень:

- Інноваційність: Запропонована система рекомендацій базується на сучасних алгоритмах машинного навчання з використанням гібридного підходу, який об'єднує методи колаборативної фільтрації та фільтрації на основі вмісту.
- Унікальні технічні характеристики: Система адаптована до великих обсягів даних та може забезпечувати високоякісні рекомендації навіть за умови розширення бази користувачів та контенту.
- Практична здійсненність: Обрана архітектура розробки є гнучкою та легко піддається адаптації та вдосконаленню.

2. Комерційний потенціал:

- Ринкові можливості: Запропонована система розроблена в умовах зростання популярності онлайн-платформ для перегляду фільмів і серіалів, таких як Netflix, Amazon Prime Video, Megogo, що активно використовують персоналізовані рекомендації для залучення та утримання користувачів.
- Аналіз конкурентів: Існуючі системи рекомендацій, які використовуються на ринку, мають обмежені можливості для персоналізації контенту, часто не враховуючи детальну історію переглядів, специфічні інтереси або швидкі зміни уподобань користувача.
- Цінова стратегія та переваги продукту: Упровадження такої технології може бути комерційно привабливим для кінотеатрів онлайн, потокових

платформ та соціальних мереж, де персоналізація контенту грає ключову роль.

- Розрахунок витрат на реалізацію проекту: Для реалізації проекту включатиме в себе детальний підрахунок усіх витрат, пов'язаних з оплатою праці членів команди. Значна частина цих витрат припадає на оплату праці основних фахівців, серед яких керівник проекту, розробники, аналітики та тестувальники.

Зроблені розрахунки зводимо до таблиці 4.3.

Таблиця 4.3 – Витрати на заробітну плату дослідників

Посада	Місячний оклад, грн	Оплата за робочий день, грн	Кількість днів роботи	Витрати на заробітну плату, грн.
Керівник	12,000	545	20	10,900
Розробник	10,000	455	29	13,195
Аналітик	8,500	386	25	9,650
Тестувальник	8000	341	30	10,230
Разом	43975			

Витрати на основну заробітну плату робітників (Z_p) за відповідними найменуваннями робіт розраховують за формулою:

$$Z_p = \sum_{i=1}^n C_i \cdot t_i, \quad (4.2)$$

де C_i – погодинна тарифна ставка працівник відповідного розряду, за виконання певного виду роботи, грн/год; t_i – час, витрачений працівником на виконання цієї роботи, год.

Погодинна тарифна ставка робітника відповідного розряду можна C_i

визначити за формулою:

$$Ci = \frac{M_m \cdot Ki \cdot K_c}{T_p \cdot t_{зм}}, \quad (4.3)$$

де M_m – розмір прожиткового мінімуму для працездатного особи або мінімальної заробітної плати, встановленої чинним законодавством у 2024 році, $M_m=8000$ грн; K_i – коефіцієнт міжкваліфікаційного співвідношення для визначення тарифної ставки працівника залежно від кваліфікаційного розряду; K_c – співвідношення місячних тарифних ставок робітників першого розряду з умовами праці на виробничих підприємствах до затвердженого розміру мінімальної заробітної плати та враховується для встановлення умов оплати праці.; T_p – середня кількість робочих днів в місяці, приблизно $T_p = 21 \dots 23$ дні; $t_{зм}$ – тривалість зміни, год.

Зроблені розрахунки зводимо до таблиці 4.4.

Таблиця 4.4 – Витрати на заробітну плату робітників

Найменування робіт	Трудомісткість, н-год.	Розряд роботи	Погодинна тарифна ставка	Тариф. коэф.	Величина, грн.
Інформаційна технологія щодо рекомендуван ня фільмів	70	4	50,6	1,27	3542
Аналіз сучасних засобів рекомендуван ня фільмів	70	4	50,6	1,27	3542

Продовження таблиці 4.4

Розробка алгоритмів функціонування інформаційної технології щодо рекомендуванн ^я фільмів	240	5	53,1	1,36	12744
Розробка структури інформаційної технології щодо рекомендуванн ^я фільмів	120	4	53,1	1,27	6372
Реалізація програмного забезпечення	660	4	58,2	58,2	38412
Всього	64612				

Додаткова заробітна плата: Додаткова заробітна плата розраховується як 10% від суми основної заробітної плати, що становить:

$$Зд = 0,1 \cdot (Зо + Зр) = 0,1 \cdot (43975 + 64612) = 10858 \text{ грн,} \quad (4.4)$$

Відрахування на соціальні заходи: Відрахування на соціальні заходи обчислюються за ставкою 22% від загальної суми основної та додаткової заробітної плати:

$$Нзп = 0,22 \cdot (43975 + 64612 + 10858) = 26277 \text{ грн,} \quad (4.5)$$

Витрати матеріалів M , які були задіяні під час виконання даного етапу праці, визначаються за формулою:

$$M = \sum_1^n H_i \cdot C_i \cdot K_i, \quad (4.6)$$

де H_i – загальна кількість матеріалів i -го виду, шт.; C_i – ціна матеріалів i -го виду, грн.; K_i – коефіцієнт транспортних витрат, $K_i = (1,1 \dots 1,17)$; n – кількість видів матеріалів.

Зроблені розрахунки зводимо до таблиці 4.5.

Таблиця 4.5 – Витрати на матеріали

Найменування матеріалів	Ціна за одиницю, грн.	Витрачено	Вартість витрачених комплектуючих, грн.
Папір	165	1	650
Флешка	300	1	300
CD-диск	12	1	12
Всього, з врахуванням коефіцієнта транспортних витрат			947

До складу балансової вартості програмного забезпечення включають витрати на його встановлення, які становлять додаткові 10–12% від його вартості. Балансову вартість програмного забезпечення розраховують за формулою:

$$В_{\text{прг}} = \sum_1^K C_{\text{іпрг}} \cdot C_{\text{спргі}} \cdot K_i, \quad (4.7)$$

де $C_{\text{іпрг}}$ – вартість придбання програмного забезпечення i -го типу, грн.; $C_{\text{спргі}}$ – кількість одиниць програмного забезпечення відповідного виду, шт.;

K_i – коефіцієнт, який враховує витрати на встановлення та налаштування програмного забезпечення, $K_i = (1, 1 \dots 1, 12)$; k – кількість видів програмного забезпечення.

Зроблені розрахунки зводимо до таблиці 4.6.

Таблиця 4.6 – Витрати на придбання програмного забезпечення

Найменування програмного забезпечення	Ціна за одиницю, грн.	Витрачено	Вартість програмного забезпечення, грн.
Microsoft Office	2000	1	2000
Ліцензія PyCharm	1,150	1	1,150
Google Cloud (обчислення)	1,000	1	1,000
Microsoft Windows	1400	1	1400
Всього, з врахуванням коефіцієнта інсталяції та налагодження			4216

Амортизація приміщень, комп'ютерного обладнання та приміщень, яке було задіяне під час виконання даного етапу роботи, розраховується за спрощеною методикою. У загальному вигляді амортизаційні витрати A можна обчислити за наступною формулою:

$$A = \frac{Ц_6}{T_в} \cdot \frac{t}{12}, \quad (4.8)$$

де $Ц_6$ – загальна балансова вартість всього комп'ютерного обладнання та приміщень, яке було задіяне під час виконання даного етапу роботи, грн.; t – термін використання основного фонду, місяці; $T_в$ – термін корисного

використання основного фонду, роки.

Зроблені розрахунки зводимо до таблиці 4.7.

Таблиця 4.7 – Амортизаційні відрахування за видами основних фондів

Найменування	Балансова вартість, грн.	Строк корисного використання, років	Термін використання, місяців	Сума амортизації, грн.
Ноутбук	27000	5	1	337
Принтер	6900	5	1	111
Роутер	1300	3	1	29
Всього	477			

Витрати на силову електроенергію $В_e$, якщо ця стаття має суттєве значення для виконання даного етапу роботи, розраховуються за формулою:

$$В_e = \sum \frac{W_{yt} \cdot t_i \cdot C_e \cdot K_{впi}}{ККД} = \frac{0,049 \cdot 1400 \cdot 7,5 \cdot 0,85}{0,87} = 459 \text{ грн}, \quad (4.9)$$

де W_{yt} – встановлена потужність обладнання, кВт; t_i – тривалість роботи обладнання на етапі дослідження, год.; C_e – вартість 1 кВт електроенергії, грн.; $K_{впi}$ – коефіцієнт використання потужності; ККД – коефіцієнт корисної дії обладнання.

Зроблені розрахунки зводимо до таблиці 4.8.

Таблиця 4.8 – Витрати на електроенергію

Найменування обладнання	Потужність, кВт	Тривалість годин роботи
Ноутбук	0,049	1400

До категорії «Інші витрати» входять ті витрати, які не були враховані у вищезазначених статтях.

Розрахунок витрат за статтею «Інші витрати» здійснюється у межах 50–100% від суми базової заробітної плати співробітників, залучених до виконання досліджень та робіт визначається за формулою:

$$I_v = (Z_o + Z_p) \cdot \frac{H_{iv}}{100\%} = (43975 + 64612) \cdot \frac{70}{100} = 76010,9 \text{ грн,} \quad (4.10)$$

де H_{iv} – норма нарахування за статтею «Інші витрати».

«Накладні витрати» включають витрати на управління, винахідницьку діяльність, підготовку кадрів, набір персоналу, банківські послуги, освоєння виробництва тощо.

Розрахунок таких витрат проводиться у межах 100–150% від суми основної заробітної плати дослідників і робітників за відповідною формулою:

$$V_{nzv} = (Z_o + Z_p) \cdot \frac{H_{nzv}}{100\%} = (43975 + 64612) \cdot \frac{120}{100} = 130304 \text{ грн,} \quad (4.11)$$

де H_{nzv} – норма нарахування за статтею «Накладні (загальновиробничі) витрати».

Витрати на проведення науково-дослідної роботи розраховуються як сума всіх попередніх статей витрат за формулою:

$$\begin{aligned} V_{zag} = Z_o + Z_p + Z_{dod} + Z_n + M_v + V_{prg} + A_{obl} + V_e + I_v + V_{nzv} = \\ 43975 + 64612 + 10858 + 26277 + 947 + 4216 + 477 + 459 + 76010,9 + \\ 130304 = 358135, \end{aligned} \quad (4.12)$$

Загальні витрати ЗВ необхідні для завершення науково-дослідної або

науково-технічної роботи та підготовки її результатів, визначаються за формулою:

$$ЗВ = \frac{В_{заг}}{\eta} = \frac{358135}{0,9} = 397927 \text{ грн}, \quad (4.13)$$

де η – коефіцієнт, який залежить від стадії виконання науково-дослідної роботи. Оскільки, якщо науково-технічна розробка знаходиться на етапі впровадження, то значення $\eta = 0,9$.

4.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором

У ринкових умовах узагальненим позитивним результатом, який може отримати потенційний інвестор від впровадження результатів певної науково-технічної розробки, є збільшення чистого доходу інвестора.

У цьому контексті, при створенні засобу, основу очікуваного економічного ефекту складатимуть такі показники: ΔN – збільшення кількості споживачів, яким надається відповідна інформаційна послуга в аналізовані періоди часу; N – кількість споживачів, яким надавалась відповідна інформаційна послуга у році до впровадження результатів нової науково-технічної розробки; Π_0 – вартість послуги у році до впровадження інформаційної системи; $\pm \Delta \Pi_0$ – зміна вартості послуги (зростання чи зниження) від впровадження результатів науково-технічної розробки в аналізовані періоди часу [25].

Очікуване зростання чистого доходу інвестора $\Delta \Pi_i$ у кожному році, протягом якого передбачаються позитивні результати від впровадження та комерціалізації науково-технічної розробки, розраховується за формулою:

$$\Delta \Pi = (\pm \Delta \Pi_0 \cdot N + \Pi_0 \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\rho}{100}\right), \quad (4.14)$$

де $\pm\Delta\Pi$ – це показник, який відображає зміну основного якісного параметра, що відбувається в результаті впровадження науково-технічної розробки в аналізованому році. Зазвичай цей таким показником може бути зміна ціни реалізації одиниці нової розробки в аналізованому році (відносно року до впровадження цієї розробки); $\pm\Delta\Pi_0$ може мати як додатне, так і від’ємне значення (від’ємне – при зниженні ціни відносно року до впровадження цієї розробки, додатне – при зростанні ціни); N – основний кількісний показник, який визначає величину попиту на аналогічні чи подібні розробки у році до впровадження результатів нової науково-технічної розробки; Π_0 – основний якісний показник, який визначає ціну реалізації нової науково-технічної розробки в аналізованому році; Π_6 – основний якісний показник, який визначає ціну реалізації існуючої (базової) науково-технічної розробки у році до впровадження результатів; ΔN – зміна основного кількісного показника від впровадження результатів науково-технічної розробки в аналізованому році. Зазвичай таким показником може бути зростання попиту на науково-технічну розробку в аналізованому році (відносно року до впровадження цієї розробки); λ – коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість. У 2024 році ставка податку на додану вартість становить 20%, а коефіцієнт $\lambda = 0,8333$; ρ – коефіцієнт, який враховує рентабельність інноваційного продукту (послуги). Рекомендується брати $\rho = 0,2 \dots 0,5$; ϑ – ставка податку на прибуток, який має сплачувати потенційний інвестор, у 2024 році $\vartheta = 18\%$.

Очікуваний термін життєвого циклу розробки 1 рік, тому:

$$\Delta\Pi = ((100000 - 40000) \cdot 2500 - (5000 - 2500) \cdot 40000) \cdot 0.8333 \cdot 0,3 \cdot \left(1 - \frac{18}{100}\right) = 10995600 \text{ грн}, \quad (4.15)$$

Далі розраховують приведену вартість збільшення всіх чистих прибутків ПП за формулою:

$$ПП = \sum_1^K = 1 \frac{\Delta \Pi_i}{(1+\tau)^t} = \frac{10995600}{(1+0,1)^1} = 9996000 \text{ грн}, \quad (4.16)$$

де $\Delta \Pi$ – збільшення чистого прибутку у кожному з років, протягом яких виявляються результати впровадження науково-технічної розробки, грн.; T – період часу, протягом якого очікується отримання позитивних результатів від впровадження та комерціалізації науково-технічної розробки, роки (приймаємо $T=1$ рік); τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, $\tau=0,05 \dots 0,15$; t – період часу (в роках) від моменту початку впровадження науково-технічної розробки до моменту отримання потенційним інвестором додаткових чистих прибутків у цьому році.

Далі розраховують величину початкових інвестицій PV , які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки. Для цього використовуємо формулу:

$$PV = k_{\text{інв}} \cdot ЗВ = 5 \cdot 397927 = 1989635 \text{ грн}, \quad (4.17)$$

де $k_{\text{інв}}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію. Це можуть бути витрати на підготовку приміщень, розробку технологій, навчання персоналу, маркетингові заходи тощо; зазвичай $k_{\text{інв}}=2 \dots 5$, але може бути і більшим; $ЗВ$ – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, грн.

Тоді абсолютний економічний ефект $E_{\text{абс}}$ або чистий приведений дохід для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{\text{абс}} = ПП - PV = 9996000 - 1989635 = 8006365 \text{ грн}, \quad (4.18)$$

де $ПП$ – приведена вартість зростання всіх чистих прибутків від можливого впровадження та комерціалізації науково-технічної розробки, грн.; PV – теперішня вартість початкових інвестицій, грн.

Оскільки $E_{abc} > 0$, то можемо припустити про потенційну зацікавленість інвесторів у розробці.

Для остаточного прийняття рішення з цього питання необхідно розрахувати внутрішню економічну дохідність E_v або показник внутрішньої норми дохідності вкладених інвестицій та порівняти її з так званою бар'єрною ставкою дисконтування, яка визначає ту мінімальну внутрішню економічну дохідність, нижче якої інвестиції в будь-яку науково-технічну розробку вкладати буде економічно недоцільно.

Внутрішня економічна дохідність інвестицій E_v , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки, розраховується за формулою:

$$E_v = T_{\sqrt{1}}^{\text{ж}} + \frac{E_{abc}}{PV} = 1 \sqrt{1 \frac{nx}{1!} + \frac{8006365}{1989635}} = 1,78. \quad (4.19)$$

де $T_{\text{ж}}$ – життєвий цикл розробки, роки.

Визначимо бар'єрну ставку дисконтування $\tau_{\text{мін}}$, тобто мінімальну внутрішню економічну дохідність інвестицій, нижче якої кошти у впровадження науково-технічної розробки та її комерціалізацію вкладатися не будуть.

Мінімальна внутрішня економічна дохідність вкладених інвестицій $\tau_{\text{мін}}$ визначається за формулою:

$$\tau_{\text{мін}} = d + f = 0,9 + 0,5 = 1,4. \quad (4.20)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2024 році в Україні $d = 0,9...0,12$; f – показник, що

характеризує ризикованість вкладення інвестицій; зазвичай величина $f = 0,05 \dots 0,5$, але може бути і значно вищою.

Оскільки $E_B = 1,78 > \tau_{\min} = 1,4$, то потенційний інвестор може бути зацікавлений у фінансуванні впровадження науково-технічної розробки та виведенні її на ринок, тобто в її комерціалізації.

Далі розраховуємо період окупності інвестицій T_o , які можуть бути вкладені потенційним інвестором:

$$T_o = \frac{1}{E_B} = \frac{1}{1,78} = 0,56 \text{ року.} \quad (4.21)$$

Оскільки $T_o = 0,56 < 1 \dots 3$ -х років, то це свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора цієї розробки ведення її на ринок.

4.4 Висновок до розділу 4

У розділі було проведено детальні розрахунки витрат, необхідних для створення та впровадження нової інформаційної системи. Загальна сума витрат становить 397927 гривень, що включає всі основні статті витрат, кожна з яких була ретельно проаналізована та оцінена. Також було здійснено прогнозування очікуваного чистого прибутку, який може отримати компанія в результаті впровадження даної системи, і розраховано термін окупності проєкту. Дослідження показало, що використання цієї інформаційної системи забезпечує значну економічну вигоду для споживачів у порівнянні з існуючими аналогами. На основі виконаних розрахунків доведено, що система є більш ефективною та економічно вигідною. Крім того, розробка має високий рівень конкурентоспроможності, що відкриває додаткові перспективи для її комерціалізації. Період окупності витрат був визначений і становить приблизно 0,56 року.

ВИСНОВКИ

Магістерська кваліфікаційна робота була спрямована на розробку інформаційної технології щодо рекомендування фільмів глядачам, що відповідає сучасним вимогам до персоналізації контенту. Основна мета дослідження полягала в удосконаленні процесу рекомендування фільмів шляхом створення системи, яка використовує гібридний підхід, поєднуючи методи спільної та контентної фільтрації. Для досягнення цієї мети було поставлено кілька завдань: аналіз існуючих рекомендаційних систем, вибір відповідних алгоритмів, розробка математичної моделі, реалізація програмної системи, а також тестування її ефективності.

У першому розділі роботи було здійснено аналіз сучасних рекомендаційних систем, зокрема методів колаборативної та контентної фільтрації, а також їх гібридного поєднання. Розглянуто переваги та недоліки кожного підходу і визначено, що гібридний підхід є найбільш ефективним. Також виконано постановку задачі дослідження.

У другому розділі було розроблено методи для побудови рекомендаційної системи. Створено математичну модель, яка враховує понад 2000 фільмів та щонайменше три основні параметри рейтингування: жанр, акторський склад і рейтинг. Розроблено архітектуру системи та обрано програмні засоби для її реалізації.

У третьому розділі реалізовано програмну систему, що базується на мові програмування Python. Для обробки інформації використовувався датасет The Movies Dataset, який містив понад 10 000 записів про фільми, включаючи їхні рейтинги, жанри та режисерів. Створено модулі для колаборативної та контентної фільтрації, інтегровано алгоритми для обробки користувацьких уподобань. Система була протестована на 100 профілях користувачів.

У четвертому розділі проведено економічний аналіз, який показав, що вартість розробки системи становила 397927 грн, а її окупність – менше одного року.

Під час роботи було досягнуто високих результатів: точність рекомендацій системи становила 91%, а час обробки одного запиту – 0,7 секунди.

Результати підтвердили ефективність запропонованого підходу. Система виявилася гнучкою, масштабованою та адаптованою до змін у вподобаннях користувачів.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Катеринчук В. С., Арсенюк І. Р. Підхід щодо реалізації системи рекомендування фільмів глядачам.// Scientific Collection «InterConf», (226): with the Proceedings of the 16th International Scientific and Practical Conference «Scientific Horizon in the Context of Social Crises» (November 26-28, 2024; Tokyo, Japan)/ comp. by LLC SPC InterConf».Tokyo: Otsuki Press, 2024. 404 p. – P. 349 – 353. URL:<https://archive.interconf.center/index.php/conference-proceeding/issue/view/26-28.11.2024/236>
2. Стаднік Е. В. Дослідження та аналіз алгоритмів машинного навчання для персоналізованих рекомендацій пошуку роботи. / Молодь в науці: дослідження, проблеми, перспективи (МН-2024), 15.10.23 – 20.05.24 : збірник матеріалів.// Е. В. Стаднік, І. Р. Арсенюк – Вінниця: ВНТУ, 2024.
URL:<https://conferences.vntu.edu.ua/index.php/mn2024/paper/viewFile/21503/17803>.
3. Зайчик В. О. Дослідження методів створення рекомендаційних систем для платформи огляду книг. Тези доповідей І науково-технічної конференції факультету інформаційних технологій та комп'ютерної інженерії.// В. О. Зайчик, І. Р. Арсенюк – Вінниця: ВНТУ, 2021. URL: <https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki2021/paper/view/12207/10159>.
4. Knowledge-based recommender systems Режим доступу: https://en.wikipedia.org/wiki/Knowledge-based_recommender_system.
5. U. K. Viswanathan. Optimization of User Based Collaborative Filtering./ U. K. Viswanathan. – New York: LAP LAMBERT Academic Publishing, 2019. – 325 p.
6. S., Berkovsky Optimization of User Based Collaborative Filtering, 2019.

7. Chen X. Recommender Systems Handbook / X. Chen, Y. Li, Z. Wang. — Boston: Springer-Verlag, 2021. — 920 p.
8. Viswanathan U. K. Optimization in Recommender Systems / U. K. Viswanathan. — New York: LAP LAMBERT Academic Publishing, 2019. — 370 p.
9. Recommender Systems: An Introduction. / [Jannach D., Zanker M., Felfernig A., Friedrich G.]. — Cambridge: Cambridge University Press, 2010. — 215 c.
10. P. Lops Recommender Systems Handbook / P. Lops, M. Gemmis, G. Semeraro; — Boston: Springer-Verlag, 2011. — 840 p.
11. Salem Y. Based Recommendation Systems / Y. Salem, W. Liu. — Texas: Gold and Co., 2017. — 640 p.
12. Bernal A., Jesus B. Addressing the Cold-Start Problem in Recommender Systems / A. Bernal, B. Jesus. — Seattle: ArtPublications, 2015. — 480 p.
13. Chen L., Pu P. Trends in Critique-Based Recommenders / L. Chen, P. Pu. — London: Springer, 2013. — 240 p.
14. Алгоритми рекомендацій — [Електронний ресурс]. — Режим доступу: <https://www.analyticsvidhya.com/blog/2023/03>.
15. Машинне навчання — [Електронний ресурс]. — Режим доступу: https://uk.wikipedia.org/wiki/Машинне_навчання.
16. Статистичний аналіз — [Електронний ресурс]. — Режим доступу: <https://www.simplilearn.com/what-is-statistical-analysis-article>.
17. Соціальний аналіз — [Електронний ресурс]. — Режим доступу: <https://buklib.net/books/34093>.
18. IMDb [Електронний ресурс]. — Режим доступу до ресурсу: <https://www.imdb.com>.
19. Netflix [Електронний ресурс]. — Режим доступу до ресурсу: <https://www.netflix.com/browse>.
20. Megogo [Електронний ресурс]. — Режим доступу до ресурсу:

<https://megogo.net/ua/about>.

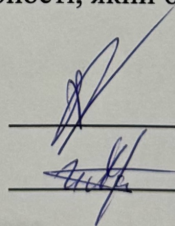
- 21.YouTube [Електронний ресурс]. – Режим доступу до ресурсу:
<https://www.youtube.com/howyoutubeworks>.
- 22.The Python Tutorial [Електронний ресурс]. – Режим доступу:
<https://docs.python.org/3/tutorial/index.html>.
- 23.Flask Framework Guide [Електронний ресурс]. – Режим доступу:
<https://flask.palletsprojects.com/en/>.
- 24.Методичні вказівки до виконання магістерських кваліфікаційних робіт для студентів спеціальності 122 «Комп'ютерні науки» [Електронний ресурс] / уклад.: А. А. Яровий, О. К. Колесницький. – Вінниця : ВНТУ,2023.– 58 с.
- 25.Web Storm – [Електронний ресурс]. – Режим доступу:
<https://www.jetbrains.com/webstorm/>.

Додаток А (обов'язковий)**Протокол перевірки кваліфікаційної роботи на наявність текстових запозичень****ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ
НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ**Назва роботи: Інформаційна технологія щодо рекомендації фільмів глядачамТип роботи: магістерська кваліфікаційна робота
(БДР, МКР)Підрозділ кафедра комп'ютерних наук, ФІТА
(кафедра, факультет)**Показники звіту подібності Turnitin**Оригінальність 98,00% Схожість 2,00%**Аналіз звіту подібності (відмітити потрібне):**

- ☒ Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
- ☐ Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її виконання автором. Роботу направити на розгляд експертної комісії кафедри.
- ☐ Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

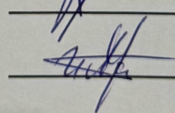
Ознайомлені з повним звітом подібності, який був згенерований системою Turnitin щодо роботи.

Автор роботи



Катеринчук В. С.

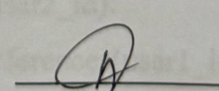
Керівник роботи



Арсенюк І. Р.

Опис прийнятого рішенняМагістерську кваліфікаційну роботу допущено до захисту

Особа, відповідальна за перевірку



Озеранський В. С.

Додаток Б (обов'язковий)**Лістинг програми****Клас Film:**

```
class Film:
    def __init__(self, title: str, year: int, genre: str, country: str, rating: float, duration:
int, age_limit: int):
        self.title = title
        self.year = year
        self.genre = genre
        self.country = country
        self.rating = rating
        self.duration = duration
        self.age_limit = age_limit

    def __str__(self):
        return f"{self.title} ({self.year}) - Genre: {self.genre}, Rating: {self.rating}"
```

Клас UserSimilarity:

```
import math

class UserSimilarity:
    def __init__(self, user_preferences):
        self.user_preferences = user_preferences

    def calculate_similarity(self, user1_id, user2_id):
        user1_preferences = self.get_user_preferences(user1_id)
```



```

user2_preferences = self.get_user_preferences(user2_id)

if not user1_preferences or not user2_preferences:
    return 0
dot_product = sum([user1_preferences.get(key, 0) * user2_preferences.get(key,
0)
                    for key in user1_preferences.keys() & user2_preferences.keys()])
magnitude_user1 = math.sqrt(sum([value ** 2 for value in
user1_preferences.values()]))
magnitude_user2 = math.sqrt(sum([value ** 2 for value in
user2_preferences.values()]))

if magnitude_user1 == 0 or magnitude_user2 == 0:
    return 0

similarity = dot_product / (magnitude_user1 * magnitude_user2)
return similarity

def get_user_preferences(self, user_id):
    preferences = {}
    for pref in self.user_preferences:
        if pref.user_id == user_id:
            preferences[pref.question_id] = pref.answer_id
    return preferences

def get_most_similar_users(self, target_user_id):
    similarities = []
    for user_id in set([pref.user_id for pref in self.user_preferences]):
        if user_id != target_user_id:

```

```

        similarity = self.calculate_similarity(target_user_id, user_id)
        similarities.append((user_id, similarity))
    similarities.sort(key=lambda x: x[1], reverse=True)
    return similarities[:5]

```

Клас CollaborativeFiltering:

```

import numpy as np
from collections import defaultdict
from sklearn.metrics.pairwise import cosine_similarity

class CollaborativeFiltering:
    def __init__(self, films, user_preferences):
        self.films = films
        self.user_preferences = user_preferences
        self.user_ratings_matrix = self.create_user_ratings_matrix()

    def create_user_ratings_matrix(self):
        user_ratings = defaultdict(lambda: defaultdict(int))

        for pref in self.user_preferences:
            user_ratings[pref.user_id][pref.question_id] = int(pref.answer_id)

        return user_ratings

    def calculate_similarity(self, user1_id, user2_id):
        user1_ratings = np.array([self.user_ratings_matrix[user1_id].get(film_id, 0) for
            film_id in self.user_ratings_matrix[user1_id]])

```

```

user2_ratings = np.array([self.user_ratings_matrix[user2_id].get(film_id, 0) for
film_id in self.user_ratings_matrix[user2_id]])

```

```

similarity = cosine_similarity([user1_ratings], [user2_ratings])[0][0]

```

```

return similarity

```

```

def get_most_similar_users(self, target_user_id):

```

```

    similarities = []

```

```

    for user_id in self.user_ratings_matrix:

```

```

        if user_id != target_user_id:

```

```

            similarity = self.calculate_similarity(target_user_id, user_id)

```

```

            similarities.append((user_id, similarity))

```

```

    similarities.sort(key=lambda x: x[1], reverse=True)

```

```

    return similarities[:5]

```

```

def recommend_films(self, target_user_id):

```

```

    similar_users = self.get_most_similar_users(target_user_id)

```

```

    recommended_films = defaultdict(float)

```

```

    for similar_user_id, _ in similar_users:

```

```

        for film_id, rating in self.user_ratings_matrix[similar_user_id].items():

```

```

            if film_id not in self.user_ratings_matrix[target_user_id]:

```

```

                recommended_films[film_id] += rating

```

```

    sorted_recommended_films = sorted(recommended_films.items(), key=lambda
x: x[1], reverse=True)

```

```

    top_films = [self.films[film_id] for film_id, _ in sorted_recommended_films[:5]]

```

```

    return top_films

```

Клас FilmService

```

class FilmService:
    def __init__(self, films):
        self.films = films

    def search_by_director(self, director_name: str):
        return [film for film in self.films if director_name.lower() in
film.director.lower()]

```

Клас: SearchByTitle

```

import pandas as pd

class Film:
    def __init__(self, title: str, year: int, genre: str, country: str, rating: float,
duration: int, age_limit: int):
        self.title = title
        self.year = year
        self.genre = genre
        self.country = country
        self.rating = rating
        self.duration = duration
        self.age_limit = age_limit

    def __str__(self):
        return f"{self.title} ({self.year}) - Genre: {self.genre}, Rating:
{self.rating}"

```

```

class MovieDataset:
    def __init__(self):
        self.df = pd.DataFrame(columns=['Title', 'Year', 'Genre', 'Country',
'Reating', 'Duration', 'Age Limit'])

    def add_film(self, film: Film):
        self.df = self.df.append({
            'Title': film.title,
            'Year': film.year,
            'Genre': film.genre,
            'Country': film.country,
            'Rating': film.rating,
            'Duration': film.duration,
            'Age Limit': film.age_limit
        }, ignore_index=True)

    def search_by_title(self, title: str):
        return self.df[self.df['Title'].str.contains(title, case=False, na=False)]

films = [
    Film("Film 1", 2021, "Action", "USA", 8.5, 120, 18),
    Film("The Great Adventure", 2022, "Adventure", "UK", 7.4, 150, 12),
    Film("Action Movie 2", 2023, "Action", "USA", 8.0, 130, 16),
    Film("The Comedy Show", 2024, "Comedy", "UK", 8.2, 95, 13)
]

MovieDataset
movie_dataset = MovieDataset()

```

```

for film in films:
    movie_dataset.add_film(film)

searched_movies = movie_dataset.search_by_title("action")

print(searched_movies)

```

Клас ContentBasedFiltering:

```

class ContentBasedFiltering:
    def __init__(self, films):
        self.films = films

    def filter_by_genre(self, genre: str):
        return [film for film in self.films if genre.lower() in film.genre.lower()]

    def filter_by_country(self, country: str):
        return [film for film in self.films if country.lower() in film.country.lower()]

    def filter_by_rating(self, min_rating: float, max_rating: float):
        return [film for film in self.films if min_rating <= film.rating <= max_rating]

    def filter_by_age_limit(self, age_limit: int):
        return [film for film in self.films if film.age_limit >= age_limit]

    def filter_by_duration(self, min_duration: int, max_duration: int):
        return [film for film in self.films if min_duration <= film.duration <=
max_duration]

```

```
def filter_films(self, genre=None, country=None, min_rating=None,
max_rating=None, age_limit=None, min_duration=None, max_duration=None):
    filtered_films = self.films

    if genre:
        filtered_films = self.filter_by_genre(genre)
    if country:
        filtered_films = self.filter_by_country(country)
    if min_rating is not None and max_rating is not None:
        filtered_films = self.filter_by_rating(min_rating, max_rating)
    if age_limit:
        filtered_films = self.filter_by_age_limit(age_limit)
    if min_duration is not None and max_duration is not None:
        filtered_films = self.filter_by_duration(min_duration, max_duration)

    return filtered_films
```

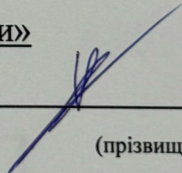
Додаток В (обов'язковий)

ІЛЮСТРАТИВНА ЧАСТИНА

«ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ЩОДО РЕКОМЕНДУВАННЯ
ФІЛЬМІВ ГЛЯДАЧАМ»

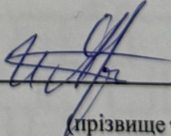
Виконав: студент 2-го курсу,
групи КН-23м

Спеціальності 122 – «Комп'ютерні
науки»


Катеринчук В. С.

(прізвище та ініціали)

Керівник: д.т.н., доцент. каф. КН


Арсенюк І. Р.

(прізвище та ініціали)

« 05 » 12 2024 р.

Вінниця ВНТУ - 2024 рік

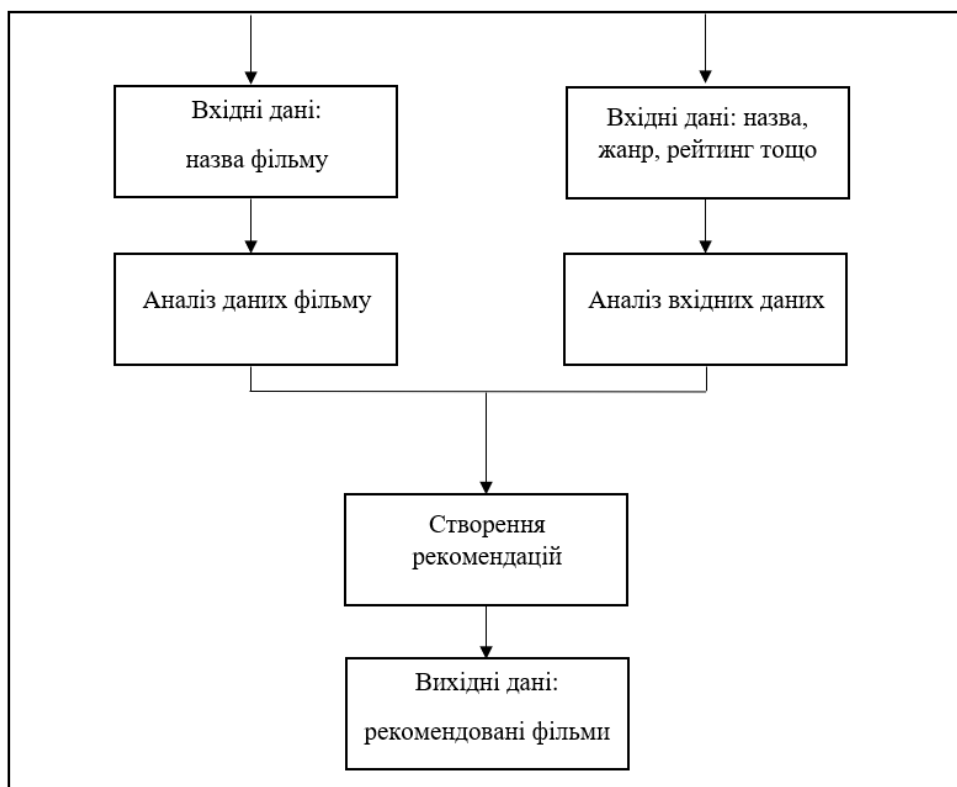


Рисунок В.1 – Діаграма сценаріїв роботи технології аналізу та рекомендації фільмів.

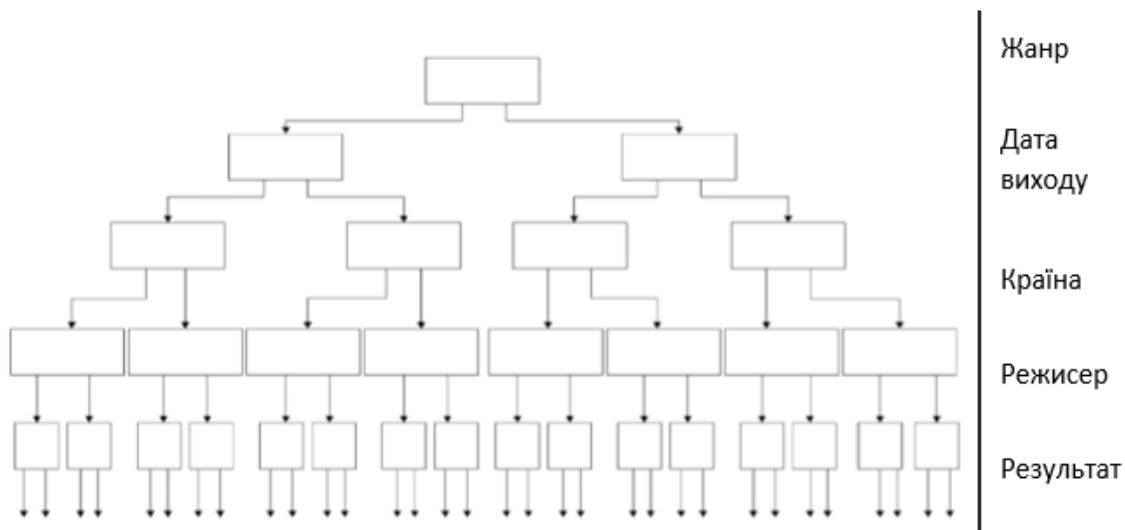


Рисунок В.2 – Структурна схема дерева рішень інформаційної технології підбору фільмів

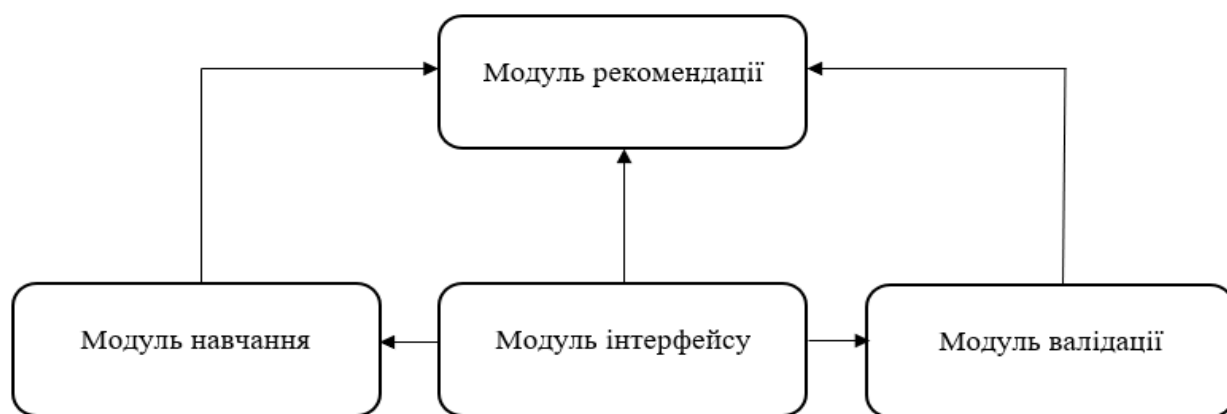


Рисунок В.3 – Загальна структура інформаційної системи щодо
рекомендування фільмів глядачам



Рисунок В.4 – Структура інформаційної технології щодо рекомендування фільмів глядачам

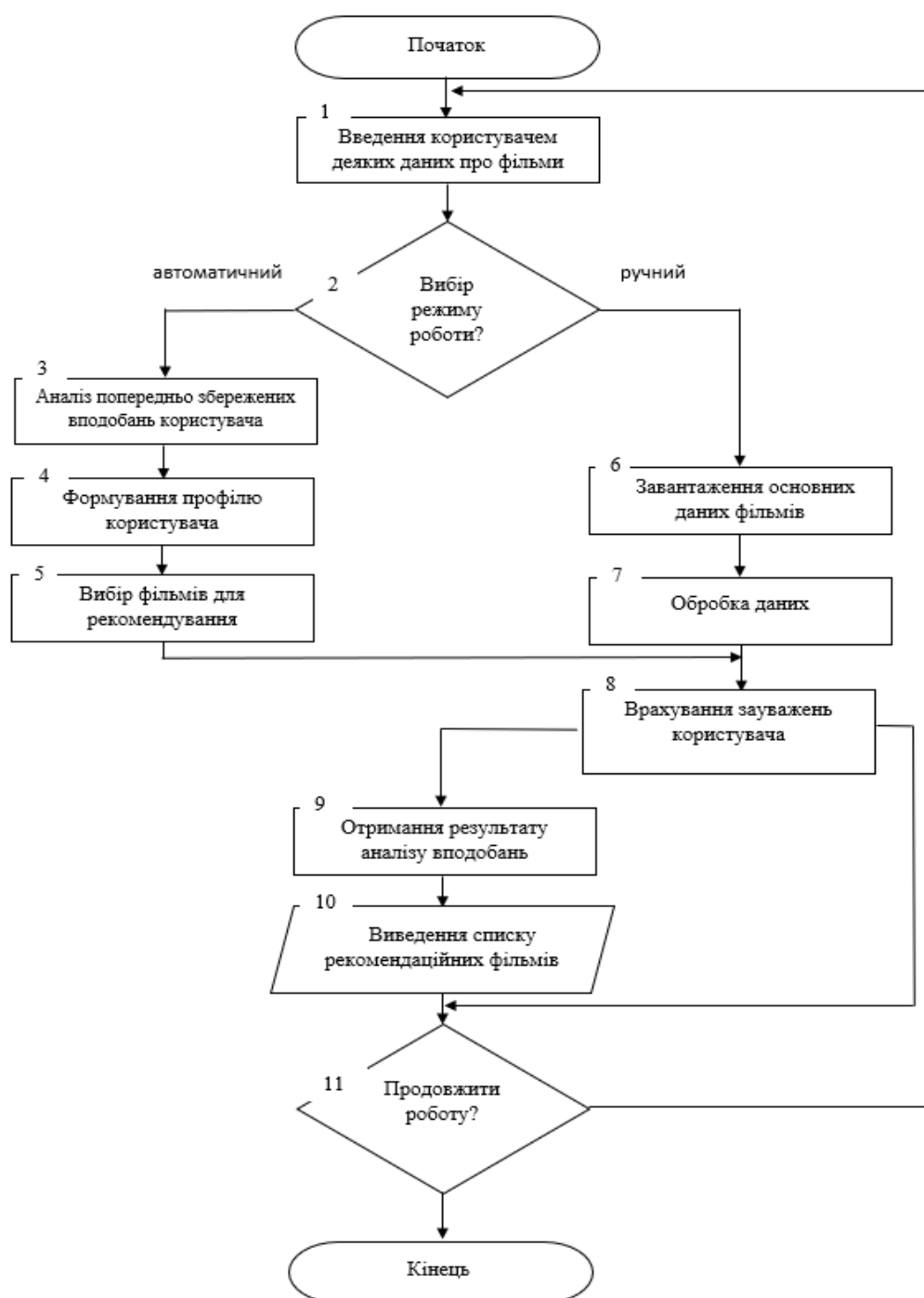


Рисунок В.5 – Загальний алгоритм роботи інформаційної технології щодо рекомендації фільмів глядачам

Інформаційна технологія щодо рекомендування фільмів глядачам

Введіть назву фільму для пошуку:

Дюна: Частина друга

Пошук

Виберіть жанри:

Введіть акторів:

Введіть улюблених режисерів:

Зберегти уподобання



Дюна: Частина друга
(2024)

Рисунок В.6 – Рекомендований фільм для користувача

Додаток Г (довідниковий)

Інструкція користувача

Крок 1. Після запуску програми відкривається вікно головного інтерфейсу (рисунок Г.1).

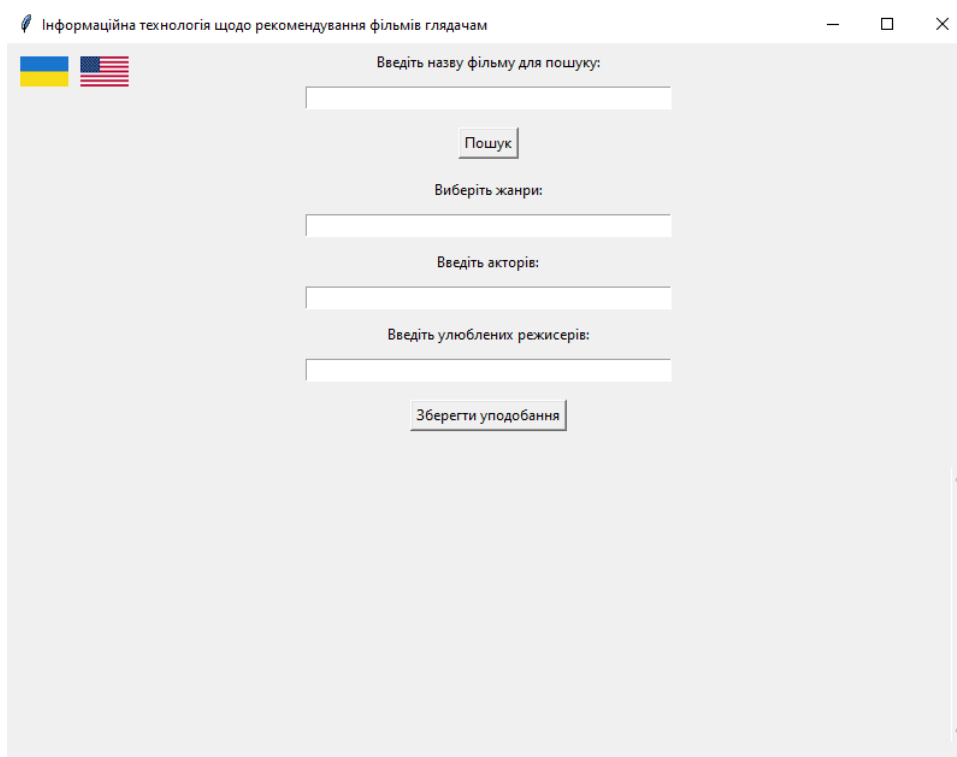


Рисунок Г.1 – Головний інтерфейс розробленої програми

Крок 2. Аби почати роботу програми, необхідно обрати одну із запропонованих опцій. При виборі «Введіть назву фільму для пошуку» користувач вводить назву фільму. Після введення користувачем необхідно натиснути кнопку «Пошук»

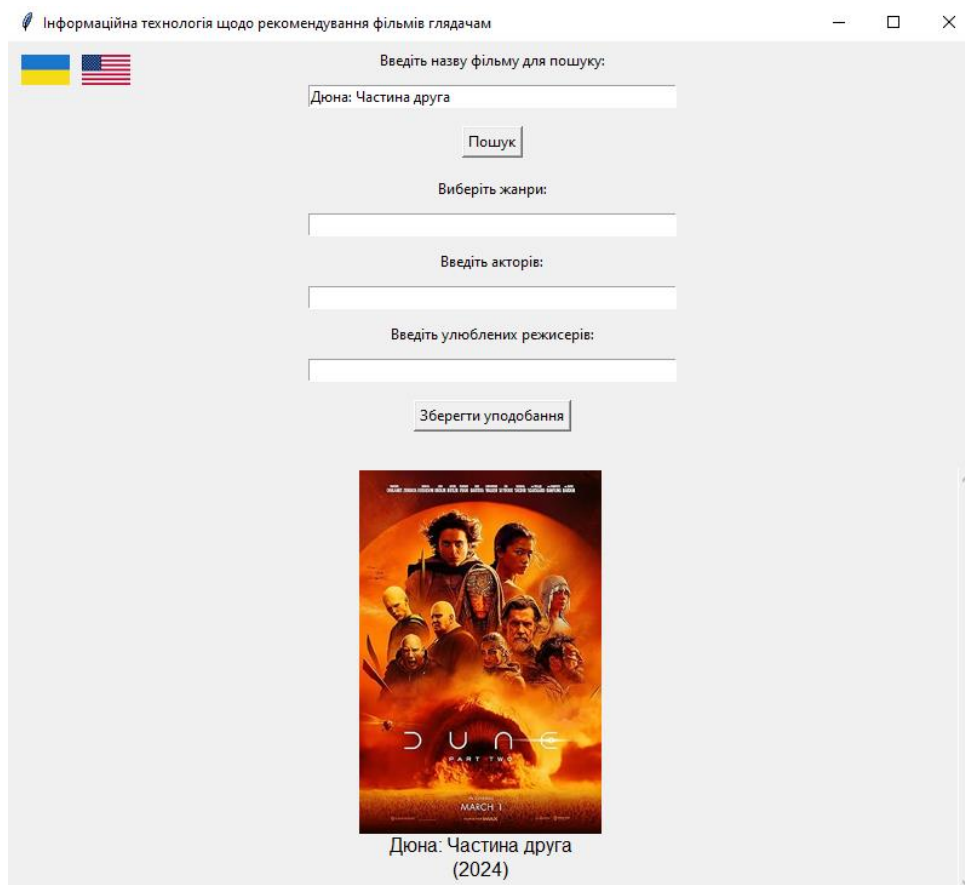


Рисунок Г.2 – Рекомендований фільм для користувача

Крок 3. Користувач може виконати пошук фільмів за ім'ям актора та ключовим словом у відповідні поля та натиснути кнопку "Пошук". Також доступна зміна мови програми. Це продемонстровано на рисунках Г.3 та Г.4.

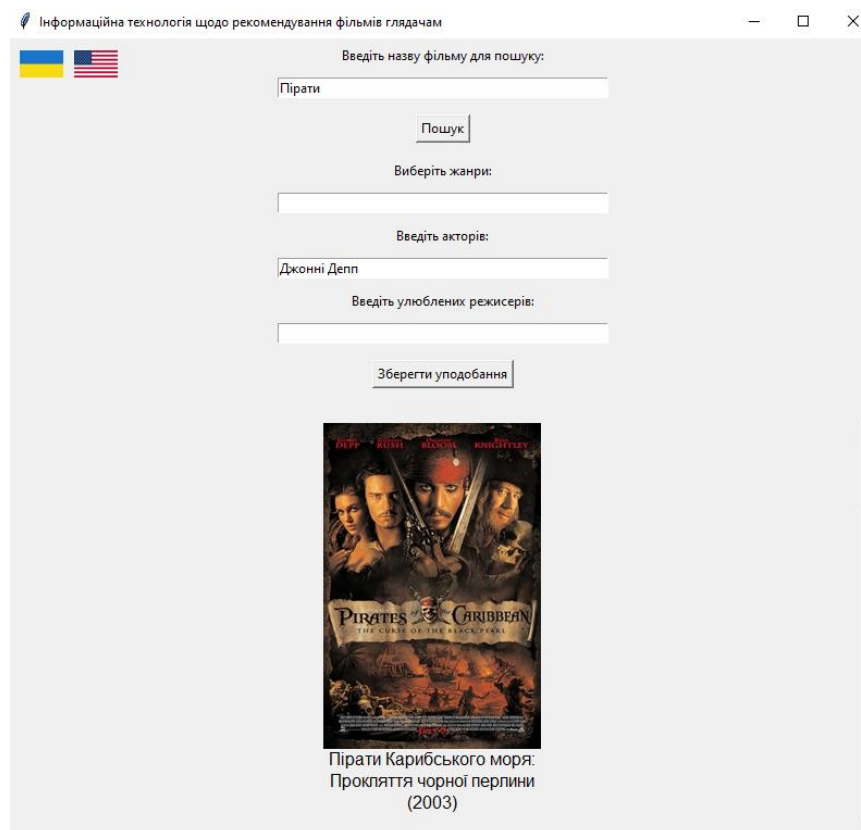


Рисунок Г.3 – Рекомендовані фільми для користувача за акторами та ключовим СЛОВОМ

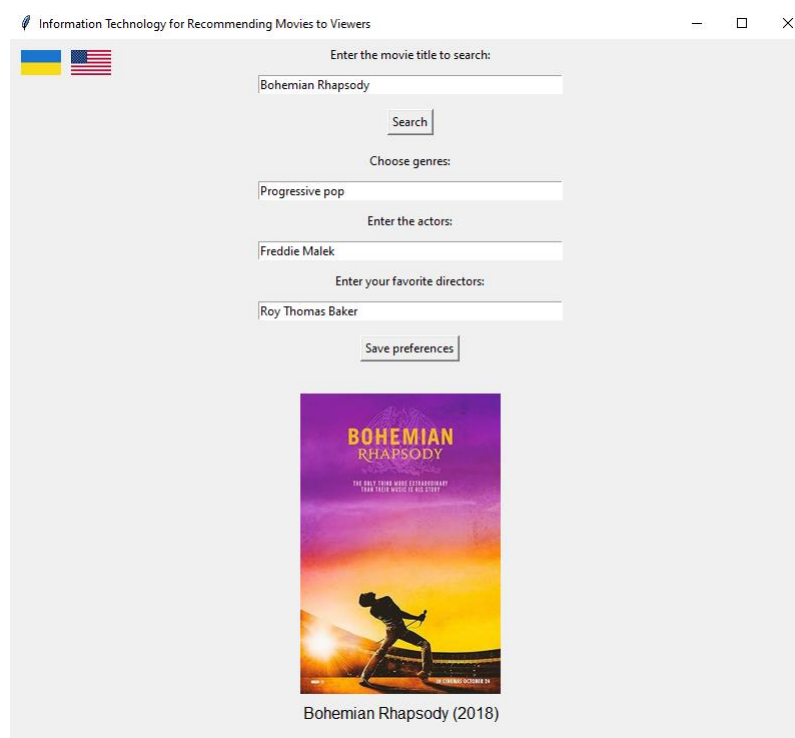


Рисунок Г.4 – Рекомендований фільм для користувача з використанням англomовного інтерфейсу