Вінницький національний технічний університет
Факультет комп'ютерних систем і автоматики
Кафедра системного аналізу та інформаційних технологій

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ АНАЛІЗУ ТА РЕКОМЕНДУВАННЯ КІНОФІЛЬМІВ ДЛЯ ГЛЯДАЧІВ

Пояснювальна записка до магістерської кваліфікаційної роботи

Виконав: студент 2 курсу, групи 2ІСТ-19м
спеціальності 126 – «Інформаційні системи та технології»
Корпало А.В.

Керівник: доцент, к.т.н. Козачко О.М. _____

Рецензент: доцент, к.т.н. Бойко О.Р. _____

Вінниця ВНТУ – 2020 року

Вінницький національний технічний університет
Факультет комп'ютерних систем і автоматики
Кафедра системного аналізу та інформаційних технологій

Освітньо-кваліфікаційний рівень магістр
Спеціальність 126 - Інформаційні системи та технології

ЗАТВЕРДЖУЮ
Завідувач кафедри САІТ
_____ д.т.н., проф. В. Б. Мокін
"__" _____ 2020 р.

ЗАВДАННЯ
на магістерську кваліфікаційну роботу студенту
Корпало Андрію Володимировичу

1. Тема роботи: «Інформаційна технологія аналізу та рекомендування кінофільмів для глядачів»,
керівник роботи: Козачко О. М.., доцент, к.т.н,
затверджені наказом закладу вищої освіти від "__"_____2020 року №__
2. Строк подання студентом роботи_____
3. Вихідні дані до роботи:
   - датасет «The Movies Dataset»;
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):
   - огляд і аналіз існуючих рекомендаційних систем;
   - розробка інформаційної технології аналізу та рекомендацій кінофільмів;
   - реалізація інформаційної технології аналізу та рекомендацій кінофільмів.
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень):
   - структурна схема інформаційної технології;
   - UML діаграма діяльності основних сценаріїв роботи інформаційної технології;
   - UML діаграма діяльності модулю спільної фільтрації інформаційної технології;
   - UML діаграма діяльності другого сценарію роботи модулю фільтрації на основі вмісту інформаційної технології;
   - UML діаграма діяльності третього сценарію роботи модулю фільтрації на основі вмісту інформаційної технології;
   - UML діаграма діяльності модулю фільтрації на основі вмісту інформаційної технології;
   - Головне вікно програми

- Результат роботи головного сценарію інформаційної технології

6. Консультанти розділів МКР

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
4	Руда Л.П., к.е.н., доц. каф. ЕПВМ		

7. Дата видачі завдання_____

КАЛЕНДАРНИЙ ПЛАН

з/п	Назва етапів МКР	Термін виконання етапів роботи	Примітка
	Аналіз предметної області	09.2020	
	Розробка інформаційної технології	09.2020	
	Реалізація інформаційної технології	10.2020	
	Економічна частина	10.2020	
	Розробка інструкції користувача	11.2020	
	Оформлення матеріалів до захисту МКР	11.2020	

Студент _____ Корпало А. В.

Керівник роботи _____ Козачко О. М.

Рецензент _____ Бойко О.Р.

РЕФЕРАТ

Магістерська кваліфікаційна робота: 106 стор., 6 таблиць, 53 рис., 25 джерел.

Об'єкт дослідження – процес створення інформаційної технології аналізу та рекомендації кінофільмів.

Мета роботи – збільшення точності рекомендацій кінофільмів та прогнозованої оцінки за рахунок використання інформаційних технологій.

Здійснено порівняльний аналіз існуючих рекомендаційних систем. Проаналізовано основні методи розробки рекомендаційних систем. Розроблено гібридний метод рекомендації. Програмно реалізовано розроблену інформаційну технологію аналізу та рекомендаціїї кінофільмів для глядачів. Виконано порівняння точності розробленої технології з аналогами.

Подальшого розвитку набула інформаційна технологія аналізу та рекомендації кінофільмів для глядачів, яка на відміну від існуючих, заснована на удосконаленому методі гібридної фільтрації, що забезпечує підвищення точності рекомендацій та прогнозованої оцінки.

Галузь застосування – сервіси, що працюють з рейтинговими системами.

ІНФОРМАЦІЙНА АНАЛІЗУ ТА РЕКОМЕНДАЦІЙ КІНОФІЛЬМІВ, ІНТЕЛЕКТУАЛЬНА ТЕХНОЛОГІЯ, РЕКОМЕНДАЦІЙНА СИСТЕМА, ПРОГНОЗУВАННЯ ОЦІНОК.

ABSTRACT

Master's qualification work: 106 pages, 6 tables, 53 figures, 25sources.

The object of research – the process of creating information technology analysis and recommendation of films.

The purpose of the work is to increase the accuracy of film recommendations and projected assessment through the use of information technology.

A comparative analysis of existing recommendation systems is performed. The main methods of developing recommendation systems are analyzed. The hybrid method of the recommendation is developed. The developed information technology of the analysis and the recommendation of films for spectators is programmatically realized. The accuracy of the developed technology is compared with analogues.

Further development of information technology analysis and recommendations of films for viewers, which, unlike existing counterparts, offers a combination of content-based filtering methods and joint filtering of films, which improves the accuracy of recommendations and predictive evaluation.

Scope of application - services that work with rating systems. INFORMATION ANALYSIS AND RECOMMENDATIONS OF MOVIES, INTELLECTUAL TECHNOLOGY, RECOMMENDATION SYSTEM, EVALUATION FORECASTING.

ЗМІСТ

ВСТУП

Швидкий розвиток збору даних призвів до нової ери інформації. Дані використовуються для створення більш ефективних систем, причому одним із типів таких систем є система рекомендацій. Рекомендаційні системи – це тип системи фільтрації інформації, оскільки вони покращують якість результатів пошуку та забезпечують найбільш релевантні елементи пошукового запиту та пов’язані з історією пошуку користувача. Вони використовуються для оцінки рейтингу або для того, щоб видати користувачеві товар якому він надасть перевагу [1].

В даний час існує дві основні стратегії побудови систем рекомендацій: спільна фільтрація та фільтрація на основі вмісту. І гібридні підходи, які певною мірою використовують обидві стратегії [2, 3]. Зазвичай, більші сервіси, такі як Netflix, YouTube або Amazon, які мають велику базу клієнтів та каталоги товарів, використовують спільну фільтрацію або гібридний підхід хоча вони також можуть використовувати системи рекомендацій на основі вмісту.

Спільна фільтрація в системах рекомендацій базується на попередній взаємодії між користувачами та елементами, що використовуються для створення нових рекомендацій. Такі взаємодії підтримуються як матриці оцінок "елемента користувача". Основна ідея полягає в тому, що цих початкових оцінок достатньо для пошуку подібних користувачів та/або подібних (близьких) елементів та вироблення рекомендацій на їх основі [2, 3].

Методи фільтрації на основі вмісту базуються на описі елементів та профілях користувацьких переваг на основі попередньої взаємодії користувача з елементами. Цей елемент взаємодії зберігається в матриці оцінок, на основі якої створюються вподобання користувачів. Основна ідея полягає в тому, щоб запропонувати користувачеві найбільш відповідні елементи для його особистих уподобань [2, 3].

Майже кожна велика технологічна компанія використовує системи рекомендацій так чи інакше: Amazon використовує їх, щоб пропонувати товари клієнтам, YouTube використовує його, щоб вирішити, яке відео дивитися під час

автоматичного відтворення, а Facebook використовує це, щоб рекомендувати сторінки, за якими люди хочуть стежити. Крім того, успіх таких компаній, як Netflix та Spotify, багато в чому залежить від ефективності їхньої рекомендаційної системи.

Отже, розробка інформаційної технології аналізу та рекомендування кінофільмів для глядачів є актуальною.

Метою даної кваліфікаційної роботи є збільшення точності рекомендацій кінофільмів та прогнозованої оцінки для глядачів за рахунок використання інформаційних технологій.

Мета роботи ставить такі **завдання дослідження**:

– оглянути існуючі стратегії створення рекомендаційних систем;

– проаналізувати способи застосування оглянутих стратегій для підвищення точності рекомендації кінофільмів;

– розробити інформаційну технологію аналізу та рекомендації кінофільмів для глядачів;

– програмно реалізувати інформаційну технологію аналізу та рекомендації кінофільмів для глядачів;

– порівняти точність передбачення розробленої технології з аналогами.

Об'єктом дослідження є процес створення інформаційної технології аналізу та рекомендації кінофільмів для глядачів.

Предметом дослідження є інформаційна технологія аналізу та рекомендації кінофільмів для глядачів.

Методи дослідження. Досягнення мети даної роботи базується на використані методів фільтрації інформації, що стосуються створення інформаційної технології аналізу та рекомендації кінофільмів для глядачів.

Наукова новизна одержаних результатів. Подальшого розвитку набула інформаційна технологія аналізу та рекомендації кінофільмів для глядачів, яка на відміну від існуючих, заснована на удосконаленому методі гібридної фільтрації, що забезпечує підвищення точності рекомендацій та прогнозованої оцінки.

Практичне значення одержаних результатів. Дослідження, виконані в роботі, дозволили розробити інформаційну технологію аналізу та рекомендації кінофільмів для глядачів, що надає рекомендації кінофільмів та прогнозує їх оцінку з більшою точністю ніж існуючі аналоги. Окремі частини системи, з незначними змінами, можуть бути використані для рекомендації, будь яких, речей в рейтингових системах.

Апробація результатів роботи. Результати роботи були представлені на XV міжнародної конференції "Контроль і управління в складних системах (КУСС-2020)", м. Вінниця, 8-10 жовтня 2020 р. [1].

Публікації. Тези за темою магістерської роботи були опубліковані у матеріалах XV міжнародної конференції "Контроль і управління в складних системах (КУСС-2020)" [1].

1 ОГЛЯД І АНАЛІЗ ІСНУЮЧИХ РЕКОМЕНДАЦІЙНИХ СИСТЕМ

1.1 Поняття рекомендаційної системи

Рекомендаційні системи – це підмножина систем фільтрації інформації, це програми та служби, які намагаються вгадати, які об’єкти (фільми, музика, новини, товари, веб-сайти тощо) будуть користувачеві найбільш цікаві, використовуючи певну інформацію (профіль користувача або інформацію про обєкт). Ці програми вдосконалили методи взаємодії між системою та користувачем, оскільки замість надання статичної інформації користувач отримує інтерактивні можливості [4].

Системи рекомендацій використовуються в різних сферах і часто використовуються для створення списків відтворення відео- та музичних систем, таких як Netflix, YouTube та Spotify, рекомендацій щодо продуктів для таких служб, як Amazon, або рекомендацій щодо вмісту для платформ соціальних мереж, таких як Facebook та Twitter [5, 6]. Переважно ці системи використовуються для комерційних цілей. Рекомендовані системи є хорошою заміною алгоритмам пошуку оскільки вони допомагають користувачам знаходити елементи, які вони не знайшли б якби шукали самі. Це насправді дуже важлива функція для інтернет-магазинів і це один з небагатьох способів добре працювати. У цьому випадку рекомендації не є звичайним додатковим варіантом платформи, вони забезпечують зручну навігацію по веб-ресурсу. Якщо в електронному каталозі понад 20 000 найменувань, навігація по ним являється непомірно сладною, якщо ж товарів мільйони це стає практично неможливо.

Існує багато конкретних прикладів використання різних типів систем рекомендацій. Рекомендації у відомого інтернет-гіпермаркету Amazon зображено на рисунку 1.1.

Рисунок 1.1 – Рекомендації книг в інтернет-магазині Amazon

На практиці, не тільки інтернет-магазини використовують цю технологію. Соціальні мережі використовують їх також. На рисунку 1.2 зображено рекомендовані користувачі у соціальній мережі Instagram.

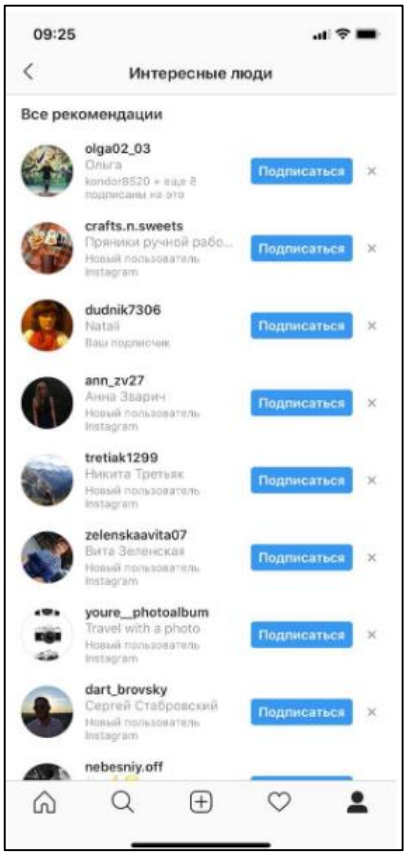


Рисунок 1.2 – Рекомендації користувачів в соціальні мережі Instagram

Крім того, подібні технології можна легко побачити на різних соціальних платформах, порталах, присвячених літературі, подорожам, новинним ресурсам, словом – майже скрізь. Ця техніка насправді дуже популярна.

Інший гарний приклад виокристання систем рекомендацій - веб-ресурс «Кинопоиск». На рисунку 1.3 зображено рекомендовані ресурсом фільми.

Рисунок 1.3 – Рекомендації фільмів у веб-ресурсі «Кинопоиск»

В даний час існує три основні та найпоширеніші стратегії створення рекомендаційних систем [2, 3]:

– content-basedfiltering (фільтрація на основі вмісту);

– collaborative filtering (спільна фільтрація);

– гібридний підхід - певною мірою використовує дві попередні стратегії.

Далі будуть детально розглянуті принципи дії цих стратегій і за допомогою них проаналізована можливість вирішення дослідницької роботи.

1.2 Рекомендаційні системи на основі спільної фільтрації

Спільна фільтрація - це триетапний процес, який починається зі збору інформації про користувача, потім будується матриця для обчислення асоціацій і, нарешті, дає дуже правдоподібну рекомендацію. ЇЇ головне припущення полягає в

наступному: хто однаково оцінював деякі обєкти в минулому, дає таку ж оцінку на схожі обєкти в майбутньому [7]. Наприклад, за допомогою спільної фільтрації музичний сервіс зможе передбачити музику яка сподобається користувачеві, на основі неповної інформації щодо його улюблених треків (лайків і дизлайків). Прогнози для кожного користувача робляться окремо, хоча використана інформація збирається від кількох учасників.

Спільна фільтрація поділяється на два основні методи:

– memorybased (на основі сусідства) - існує два основних типи цього методу: на основі користувачів, залучених у пошук, і відповідно до елементів, що складають певну категорію;

– modelbased - цей підхід надає рекомендації шляхом вимірювання параметрів статистичних моделей для оцінок користувачів, заснованих на таких методах як байєсівський метод мереж, кластеризації, семантичні моделі, такі як одиничне розкладання, імовірнісний семантичний аналіз, розподіл Діріхле і процес прийняття рішення Маркова відповідно до моделі. Моделі розробляються з використанням аналізу даних, алгоритмів машинного навчання для пошуку закономірностей на основі навчання даних. Цей підхід є більш комплексним і дає більш точні прогнози, тим самим допомагаючи виявити приховані фактори, що пояснюють спостережувані результати. Цей підхід має ряд переваг. Він обробляє розріджені матриці краще, ніж підходи, що базуються на сусідствах, що, у свою чергу, допомагає масштабувати великі масиви даних [7].

Розглянемо підхід, що базується на сусідстві більш докладно. Існує два основних методи «вибору сусідів»:

– «користувач-користувач»,

– «елемент-елемент».

Для отримання нових рекомендацій для активного користувача метод "користувач-користувач" намагається знайти користувачів із найбільш подібним "профілем взаємодії"(так звані найближці сусіди) щоб запропонувати найпопулярніші продукти серед цієї групи користувачів. Ідея, на якій заснований такий метод - шукати користувачів, смак яких схожий на уподобання нашого

цільового користувача, якнайкраще [8]. Якщо раніше "Користувач 1" та "Користувач 2" мали однакові рейтинги для багатьох фільмів, ми вважаємо, що вони мають однаковий смак, а за рейтингами певних фільмів які оцінив "Користувач 1", ми можемо дати оцінку фільмам які ще не оцінив "Користувач 2". Наприклад, якщо Жан-П'єру подобаються такі фільми як "Повернення джедаїв" і "Імперія завдає удару у відповідь", а Джейсон переглянув і високо оцінив фільм "Повернення джедаїв", ми обовязково повинні запропонувати Джейсону фільм "Імперія завдає удару у відповідь" як такий який йому може сподобатись (рис. 1.4). Вподобання користувачів фіксуються в матриці "елемент-користувач " (рис. 1.5), на основі якої системи знаходять найближчих сусідів.

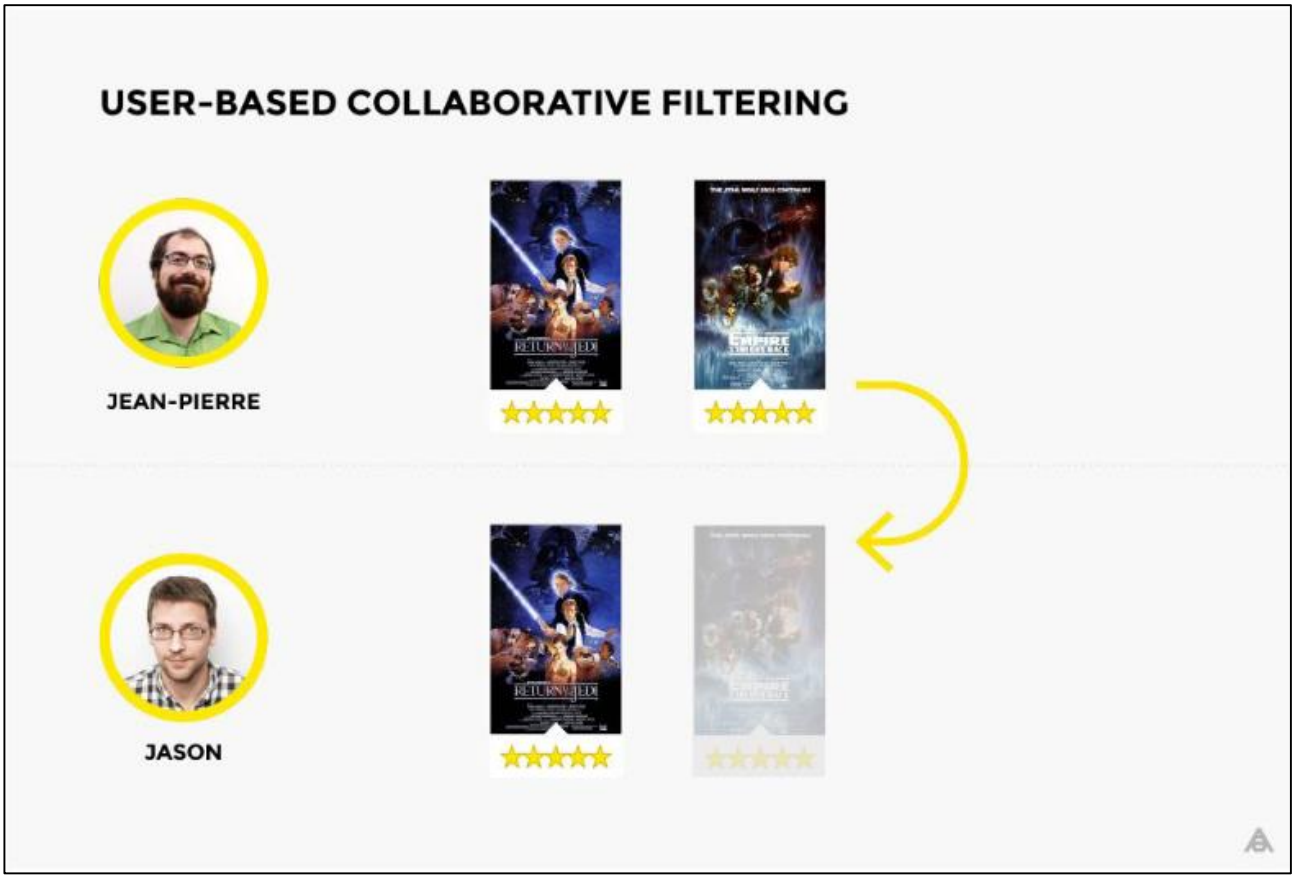


Рисунок 1.4 – Спільна фільтрація «користувач-користувач»

У такій матриці кожен рядок підходить для користувача, а кожен стовпець - для елемента. Тобто кожен рядок матриці є вектором користувацьких переваг для відображених елементів (якщо користувач ще не оцінив певний елемент,

відповідний елемент матриці залишається порожнім). Далі обчислюється міра подібності (відповідно до обраної матриці) між активним користувачем та іншими користувачами, яка полягає в наступному: Потрібно враховувати двох користувачів з однаковим балом на одному наборі однакових елементів. На основі підрахованих метрик подібності створюються групи користувачів, яким пропонують обєкти які серед них найпопулярніші, вибираючи лише ті, які цільовий користувач ще не оцінив.

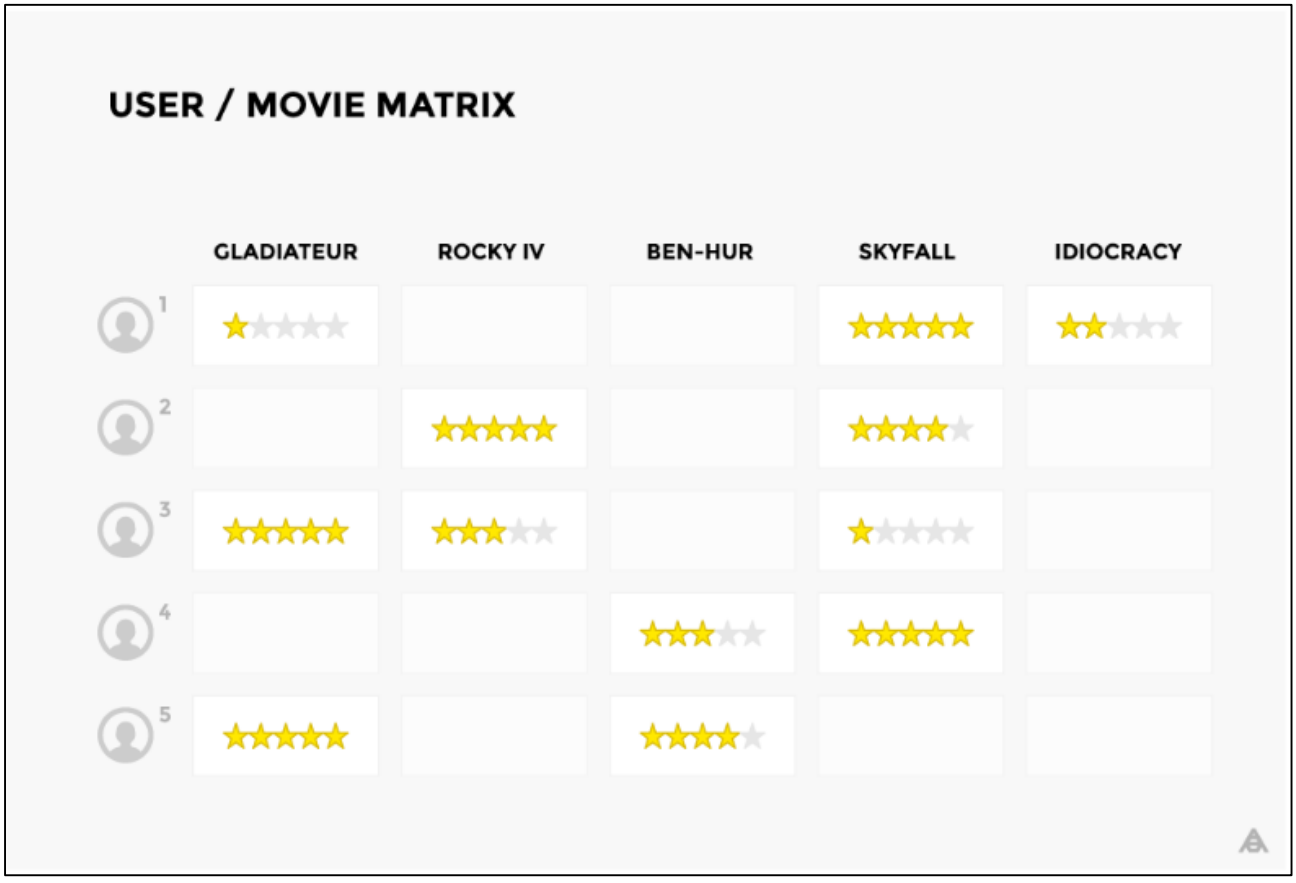


Рисунок 1.5 – Матриця «елемент-користувач»

Метод "товар-товар" полягає в тому, що користувачу система повинена робити рекомендації, подібні до предметів, які йому сподобалися в минулому. Тобто обчислюється схожість між елементами, а не користувачами. У цьому випадку два елементи вважаються ідентичними, якщо більшість користувачів взаємодіють з обома ними подібними способами [9]. Наприклад, Жан-П'єр і Джейсон любили фільми "Повернення джедаїв" та "Імперія завдає удару у

відповідь". Можна зробити висновок, що більшість користувачів, які оцінили перший фільм, повинні віддавати перевагу другому фільму. Отже, варто було запропонувати фільм "Імперія завдає удару у відповідь" Ларрі, який високо оцінив фільм "Повернення джедаїв" (рис. 1.6).

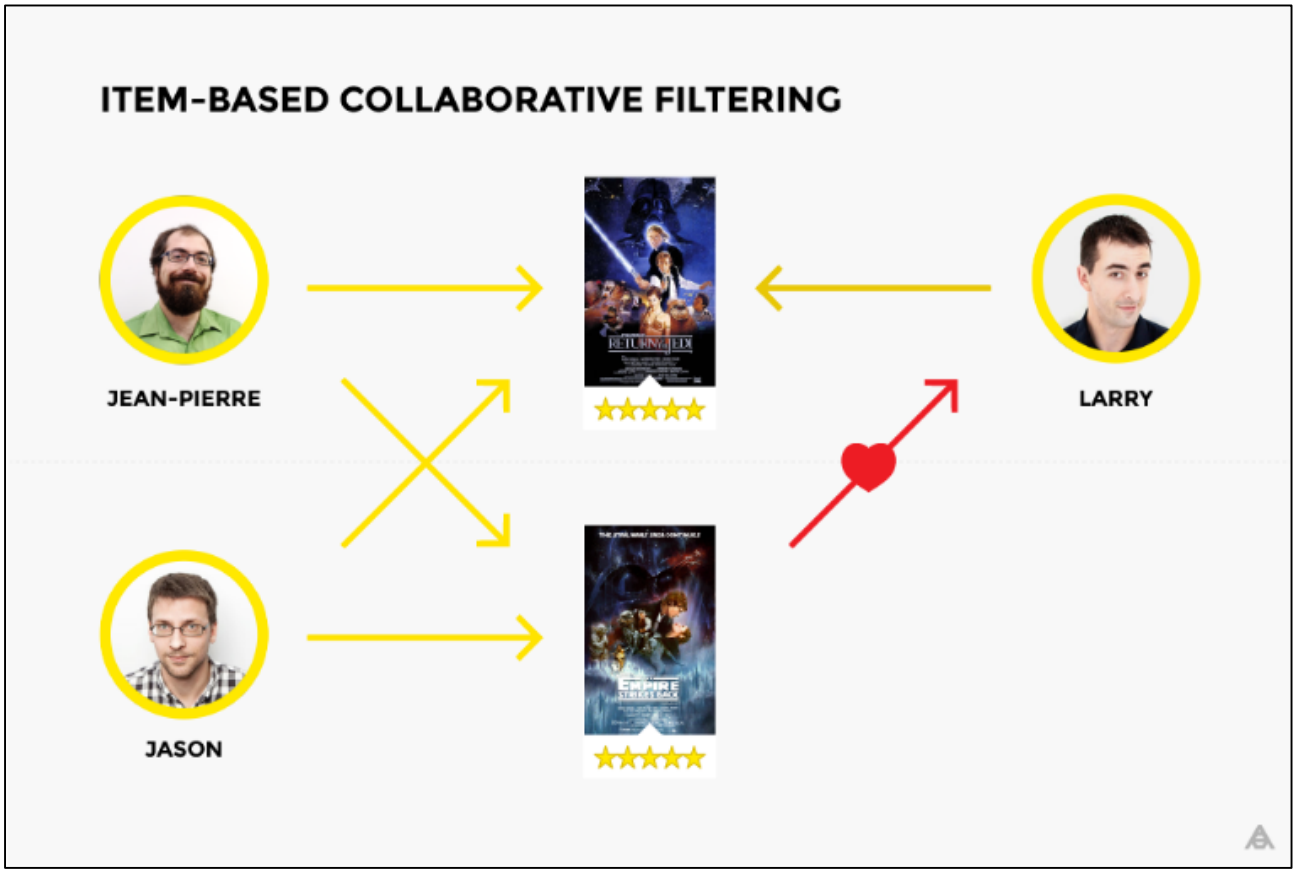


Рисунок 1.6 – Спільна фільтрація «елемент-елемент»

Даний підхід використовує ту саму матрицю, яку і попередній (рисунок 1.5) При цьому враховуються стовпці, що представляють вектор взаємодії елемента з кожним користувачем. Щоб дати рекомендацію цільовому користувачеві, елементи, які йому найбільше подобаються, представлені як вектори взаємодії з кожним користувачем. Потім розраховується ступінь подібності між цими елементами та всіма іншими наступним кроком виділяються найбільшподібні які і потрапляють у рекомендації.

Метод "користувач-користувач" базується на пошуку подібних користувачів на основі взаємодії продуктів. Якщо кожен користувач взаємодіє з невеликою

кількістю продуктів, метод стає досить чутливим до будь-якої збереженої взаємодії. З іншого боку, якщо остаточна рекомендація базується лише на захищених взаємодіях користувачів зі схожими інтересами, це дозволяє отримувати більш індивідуальний результат (менша похибка).

На пошуку обєктів які схожі між собою на основі дій користувачів над ними заснований метод "продукт-продукт". Загалом, якщо з продуктом взаємодіють декілька користувачів, пошук сусідів набагато менш чутливий до окремих взаємодій. Подібним чином взаємодії деяких кристувачів (навіть суттєво відмінні від вибраних) беруть участь у рекомендаціях, що робить цей метод менш особистим (більше помилок). Отже, цей підхід менш індивідуальни, ніж підхід "користувач-користувач", але він значно надійніший. Під час аналізу спільної фільтрації було виявлено кілька важливих недоліків:

– розрідженість користувачів - в загальному комерційні рекомендаційні системи беруть за основу товари кількість яких оцінюється в тисячах, тоді як більшість користувачів не оцінюють такі кількість товарів. Як результат, матриця "обєкт-користувач" занадто велика і розріджена, що спричиняє особливі проблеми при вирахувані рекомендацій;

– маштабованість - з великим набором користувачів у системі виникає проблема масштабованості. Наприклад, коли 10 мільйонів клієнтів і один мільйон товарів, складність алгоритму спільної фільтрації дуже велика, а сам алгоритм занадто складний для розрахунку;

– холодний старт - нові елементи або користувачі є основною проблемою для систем рекомендацій заснованих на спільній фільтрації. Такі елементи ще не мають оцінок користувачів, тому вони не будуть включені до рекомендацій, якщо матриця "елемент-користувач" не буде заповнена певною кількістю записів. Однак проблема з рекомендацією нового користувача схожа: користувач ще не має матриці переваг, і тому його не можна віднести до жодної групи [10, 11].

1.3 Рекомендаційні системи на основі вмісту

Методи фільтрації на основі вмісту базуються на описі елементів та профілях користувацьких уподобань. Дані методи найліпше підходять для ситуацій, коли є інформація про об’єкт (ім’я, місцезнаходження, опис тощо), але не про користувача. Орієнтовані на вміст рекомендації відносяться до рекомендацій як до специфічної для користувача проблемі класифікації та створюють категорію вибору користувача на основі характеристик товару.

Система використовує ключові слова для опису обєкту, а профіль користувача використовує для того щоб визначити тип елементу що подобається цьому користувачу. Говорячи простіше, ці алгоритми намагаються рекомендувати предмети, подібні до того, що сподобалось користувачеві раніше або які він вивчає в даний час. Різні обєкти-кандидати порівнюються з обєктами, попередньо оціненими користувачем, і рекомендуються найбільш відповідні обєкти. Цей підхід корениться у пошуку інформації та вивченні фільтрації інформації.

Важливою проблемою фільтрації на основі вмісту є те, чи здатна система дізнатися вподобання користувачів за допомогою дій користувача над джерелом вмісту та використовувати їх для інших типів вмісту. Наприклад, рекомендація для статей з новинами корисна, але буде ще більш корисна, коли ви зможете порекомендувати музику, відео, товари тощо на основі переглянутих новин. Для подолання цієї проблеми більшість систем рекомендацій на основі вмісту використовують гібридні системи, що містять елементи з обох методів.

Цей підхід використовує описи або атрибути елементів, за допомогою яких може рекомендувати користувачу подібні елементи. У цьому випадку залежності будуються на основі попереднього вибору користувачем, що робить цей метод надійним, та допомагає уникнути проблеми холодного запуску [2].

Припустимо, що певний користувач дивиться певний жанр фільмів, тоді йому буде рекомендовано фільми відповідного жанру. Назва, рік випуску, режисер, актори, коротче кажучи опис фільму, також дає модливість визначити схожий за вмістом фільм. Зображення цього підходу можна побачити на рисунку 1.7.

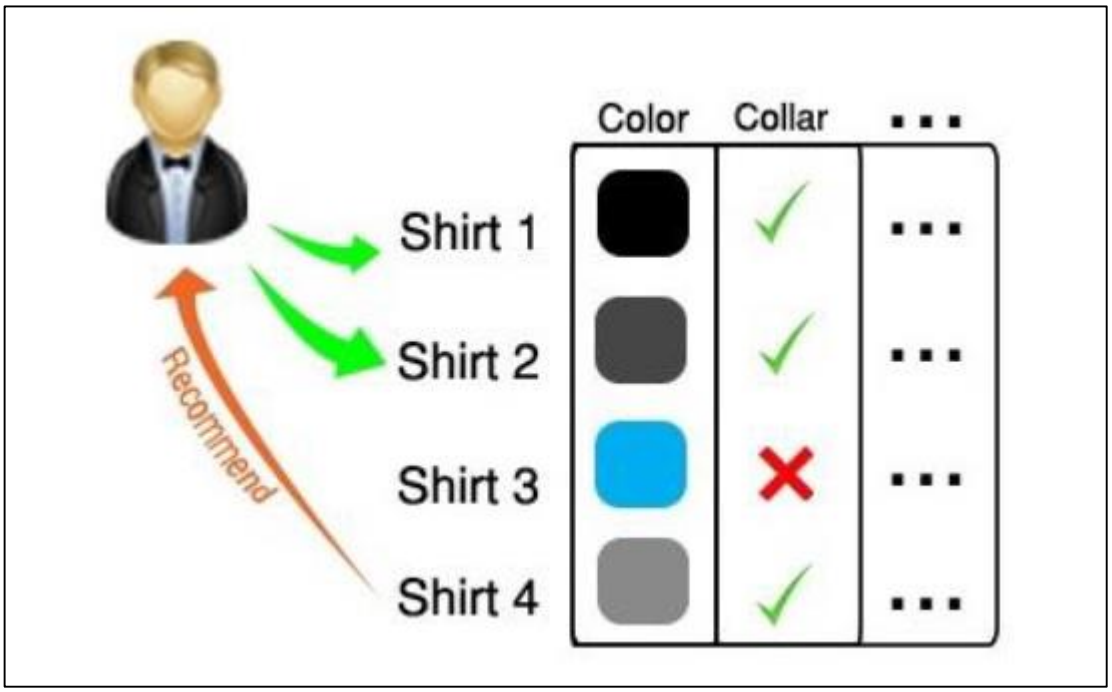


Рисунок 1.7 – Зображення підходу зі створенням профілю користувача і профілю елемента

При такому підході вміст товару оцінюється заздалегідь відповідно до уподобань користувача (профілю користувача) а значення атрибута елемента - це неявні функції, які будуть використовуватися для опису елемента. Потім предмет оцінюється за допомогою обох профілів і може бути рекомендований [2].

Щоб рекомендувати його, потрібно створити профілі користувача та елемента. Для цього існує два набори вхідних даних:

– таблиця уподобань користувача (rating table) - відображає взаємозв’язок між елементом та користувачем;

– профіль користувача (item profile) - відображає взаємозв’язок обєкт-атрибут, набір значень атрибутів елемента.

Пізніше на основі цих даних буде створено профіль користувача (user profile) який зображує взаємозвязок користувач-атрибу тобто ми маємо набір атрибутів певного типу елемента, що є найбільш важливим для користувача [3].

Тепер, коли у нас є профілі користувача та елементів, ми можем прогнозувати оцінку для елемента відповідно до уподобань цього користувача.

Цей підхід має кілька переваг перед спільною фільтрацією:

– незалежність користувача - спільна фільтрація вимагає від інших користувачів визначити оцінки щоб визначити схожість між ним, і тільки після цього вона зможе визначити рекомендацію. Натомість заснований на вмісті метод повинен аналізувати лише елементи та профілі користувачів для рекомендації;

– прозорість - метод спільної фільтрації визначає рекомендації, на основі користувачів які мають той самий смак, що і ви, але метод, заснований на вмісті, рекомендує елементи лише відповідно до їх опису;

– відсутність холодного старту - на відміну від спільної фільтрації, ви можете рекомендувати нові елементи, перш ніж їх оцінить велика кількість користувачів.

Крім переваг, цей підхід має ряд важливих недоліків:

– обмеджений аналіз контенту - якщо система не містить достатньої кількості інформації про елемент щоб розрізнити його з іншими, рекомендація буде неточною;

– надспеціалізація - метод заснований на вмісті надає обмежений ступінь новизни, тому що він змушений відповідати особливостям профілю та елементів.

– новий користувач - якщо система має не достатньо інформації про користувача щоб створити його профіль, рекомендація може бути дана неправильно.

1.4 Рекомендаційні системи на основі знань

Системи рекомендацій на основі знань (експертні рекомендаційні системи) - це особливий тип систем рекомендацій, що базується на чітких знаннях про асортимент товарів, уподобаннях користувачів та критеріях рекомендацій (тобто, який товар слід рекомендувати і в якому контексті) [12].

Ці системи використовуються у сценаріях, коли базові підходи, такі як спільна фільтрація та фільтрація на основі вмісту, не можуть бути реалізовані. Системи рекомендацій, засновані на знаннях, добре підходять для складних областей, де предмети не часто купуються, наприклад, квартири та машини [13].

Іншими прикладами предметних областей, що мають відношення до систем рекомендацій на основі знань, є фінансові послуги та туризм. Системи, що базуються на рейтингу, не працюють добре через низький рейтинг, який існує в цих сферах. Крім того, у районах зі складними предметами клієнти хочуть чітко заявити про свої уподобання (наприклад, "Максимальна ціна автомобіля-X"). У цьому контексті система рекомендацій повинна враховувати обмеження: наприклад, слід рекомендувати лише фінансові послуги, які підтримують інвестиційний період, визначений клієнтом. Такі підходи, як спільна фільтрація та фільтрація на основі вмісту, не підтримують жодного аспекту. Ще однією перевагою систем рекомендацій, заснованих на знаннях, є відсутність проблем холодного старту.

Для створення такої системи рекомендацій використовується низка ключових ідей:

– "діалог" - системи рекомендацій на основі знань, зазвичай є діалоговими, тобто користувацькі уподобання і переваги виявляються в рамках циклу зворотного зв'язку [14]. Основною причиною діалогічного характеру цих рекомендаційних систем є складність предметної області, яку часто неможливо адаптувати до всіх вимог користувача одночасно. Більше того, спочатку уподобання не відомі, але їх можна визначити на сесії рекомендацій;

– "обмеження" - звязко з користувачем може бути побудований через відповіді на питання, які обмежують певні характеристики оюєктів. [15] Прикладом такого запитання є "Якому типу автомобілів ти надаєш перевагу: легкові чи вантажні?" На технічному рівні такі сценарії рекомендацій можуть бути реалізовані на основі встановлених користувачем обмежень при виконанні пошукового запиту (пошуковий фільтр). У цьому випадку система рекомендацій є доповненням до визначених користувачем обмежень, які на основі фіксованих асоціативних правил шукають відповідні альтернативи. Такі доповнення потрібні, коли розташування елемента є складним і загальна його фільтрація (фільтр пошуку) втрачає деякі варіанти;

– "критика" - відгуки користувачів зазвичай надаються в контексті пункту "критичний коментар", які вказують запити на зміну, що стосуються

елемента, який ррекомендується користувачеві в даний час. [16] Потім критика використовується для рекомендації наступного "кандидата". Наприклад критика щодо рекомендації автомобіля: "Я хотів би мати такий самий автомобіль але з меншою ціною". Це приклад "критичної оцінки", що представляє запит на зміну одного атрибута елемента. Найбільшим недоліком цього підходу є вузьке місце знань, яке випливає з необхідності чітко визначити рекомендації [12]. Зрештою, створення такої системи рекомендацій вимагає участі експертів у даній темі, які вибудують систему асоціативних правил, на яких ґрунтуватимуться рекомендації. Метод пізнання цього є найдорожчим з точки зору фінансів, людських ресурсів та часу. Більше того, при такому підході кожна така система рекомендацій буде унікальною і не застосовуватиметься до будь-якої іншої галузі без повторного створення нових асоціативних правил.

1.5 Огляд існуючих рекомендаційних систем

Розглянемо існуючі рекомендаційні системи.

– Amazon. Сервіс електронної комерції який перший використав системи рекомендацій. Amazon рекомендує книги та інші продукти на основі товарів придбаних покупцями, чеків, встановлених рейтингів та залишених коментарів. Система збирає всю можливу інформація, хоча більшість ще ніяк не використовує, і забезпечує 35% продажів.

Інтернет-магазин використовує рекомендації на основі вмісту. Коли відвідувач вирішить придбати товар, Amazon буде використовувати даний продукт, щоб рекомендувати відвідувачеві інші товари, придбані іншими користувачами (за допомогою матриці покупки наступного придбаного товару на основі схожості з останньою покупкою). Amazon запатентувала цей метод, який назвала Item-To-Item Collaborative Filtering (спільна фільтрація від елементу до елементу). На рисунку 1.8 зображено рекомендації надані інтернет-магазином Amazon [17].

Рисунок 1.8 – Рекомендації товарів в інтернет-магазині Amazon

– Ebay. Торговий гігант вже багато років експериментує із інструментами машинного навчання та методами штучного інтелекту, що використовуються майже на всіх рівнях великого бізнесу компанії: вибір продуктів на домашній сторінці, спеціальні пропозиції, правильний рейтинг пошукових систем, реклама. Алгоритм рекомендацій може приносити додаткові продажі щомісяця близько 1 мільярда доларів США.

Система рекомендацій аналізує товари, які зараз продаються на веб-сайті, та товари, які найбільше подобаються користувачам. І під час придбання система гарантує, що товари не будуть повторюватись, а придбані товари не перекриватимуться. Щоб використовувати цієї функції, використовують розпізнавання зображень. Також важливо, щоб рекомендація була пов’язана з поточною покупкою для збільшення інтересу, і щоб покупець добавив і рекомендований товар до корзини. На рисунку 1.9 зображено домашню сторінку сайту «Ebay» [18].

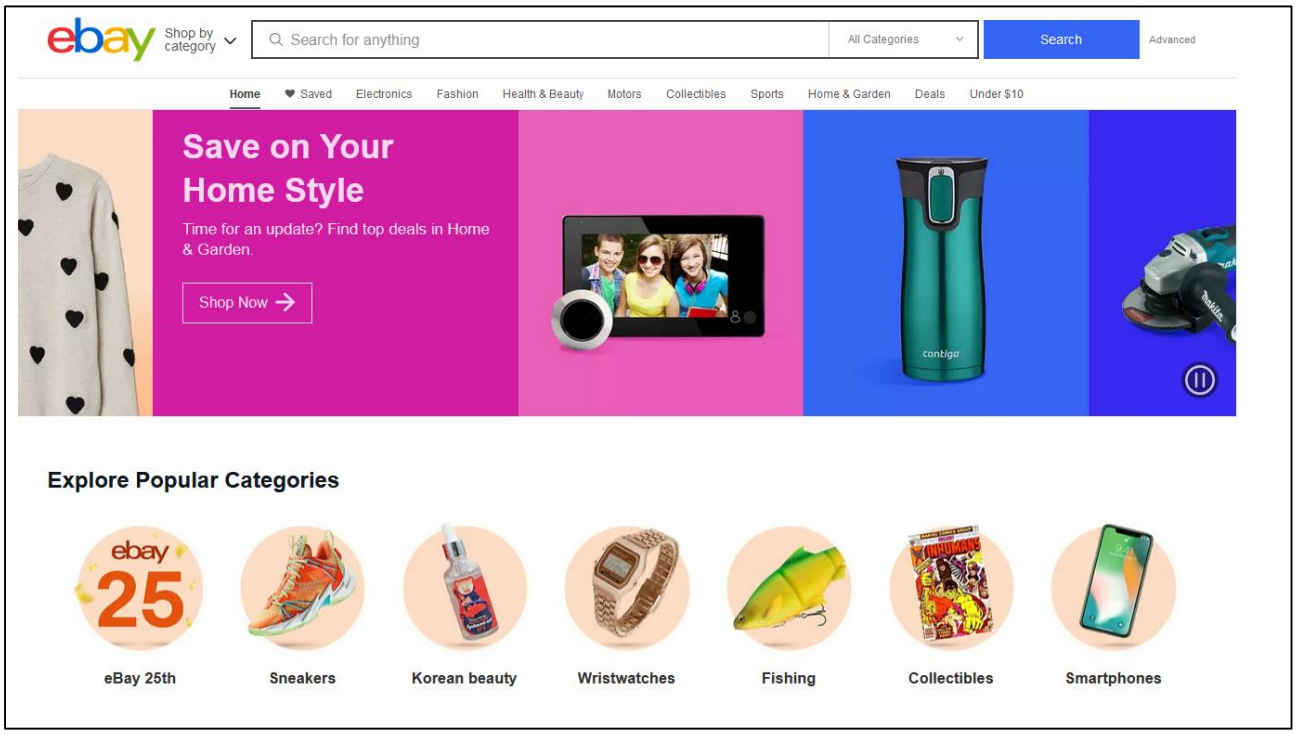


Рисунок 1.9 – Домашня сторінка сайту «Ebay»

– Netflix. Онлайн сервіс потокового відео використовує технології, засновані на штучному інтелекті та машинному навчанні, та рекомендує трейлери фільмів та серіалів відповідно до вподобань користувачів.

У зростаючій цифровій бібліотеці загальним методом для вирішення проблеми вибору вмісту глядачами є системи рекомендацій, які враховують досвід перегляду та уподобання конкретних людей. Netflix проаналізував "кожен клік" своїх 100 мільйонів користувачів, щоб досягти унікальної деталізації рекомендацій.

Зараз Netflix впроваджує машинне навчання та штучний інтелект для створення персоналізованих трейлерів. Ідея полягає не в тому, щоб вибирати найцікавіші сцени з фільмів та серіалів, як звичайні трейлери. Йдеться про їх персоналізацію: наприклад, глядачі, які люблять романтичні сцени, будуть бачити відповідні кадри в трейлерах навіть до не романтичних фільмів. На рисунку 1.10 зображено сторінку рекомендацій [19].

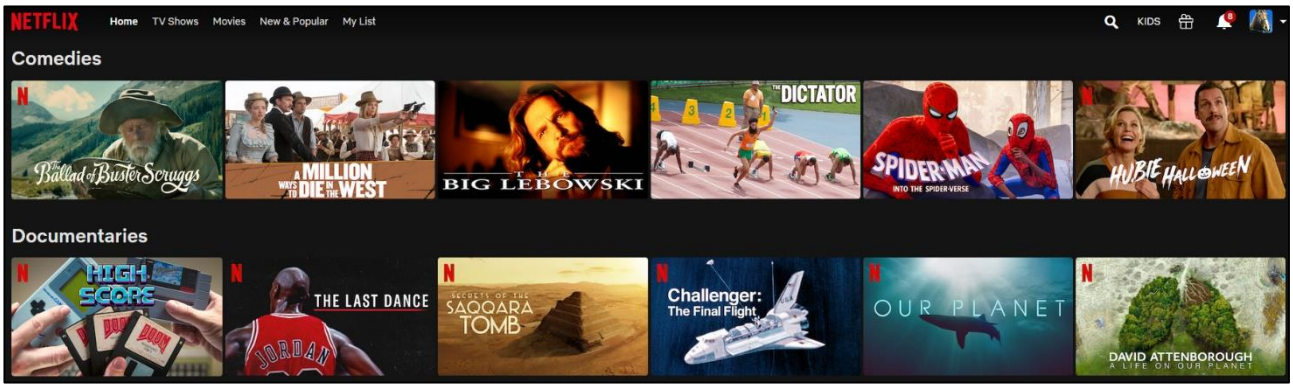


Рисунок 1.10 – Сторінка рекомендацій сервісу «Netflix»

– Megogo. Один із найбільших у Східній Європі відеосервісів для онлайн-перегляду відео та ТБ із загальною кількістю глядачів понад 50 мільйонів. Станом на 2020 рік, онлайн-бібліотека фільмів має 77 000 медіа-товарів, включаючи художні та документальні фільми, мультфільми, серіали, телешоу, спортивні та культурні програми, новини та музичні відео.

Джерелом рекомендацій є WatchStat, внутрішня система рекомендацій на основі вмісту вона надає персоналізовані рекомендації для всіх користувачів, які переглянули принаймні 2 відео. На рисунку 1.11 зображену головну сторінку сайту Megogo [20].

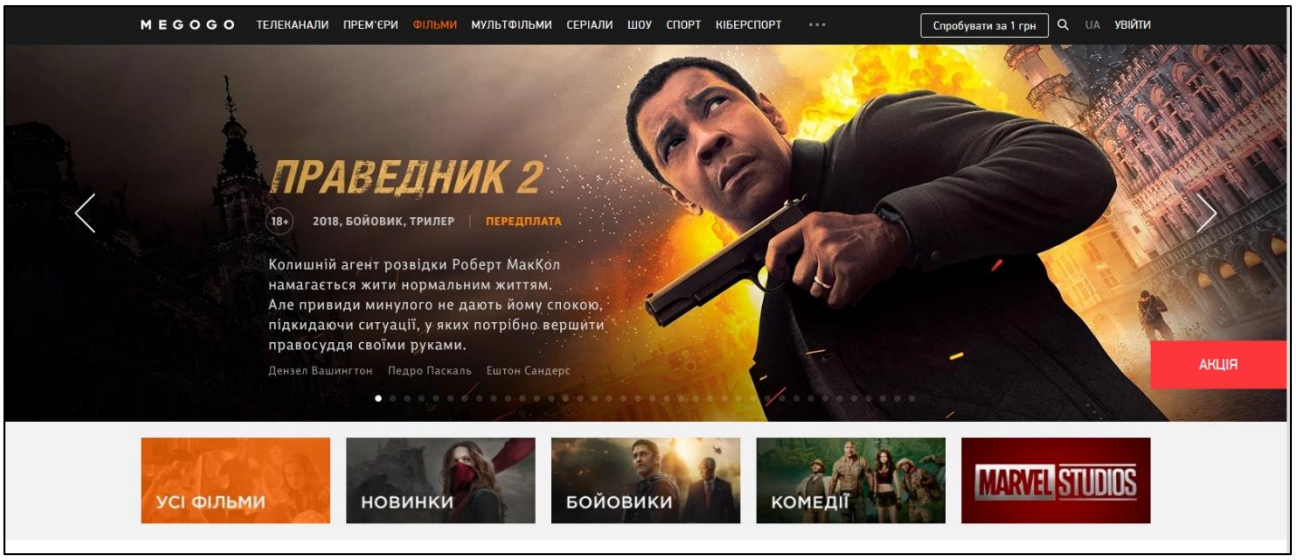


Рисунок 1.11 – Головна сторінка сайту «Megogo»

– Deezer. Надає послуги потокового передавання музики з різних джерел, таких як EMI, Sony, Universal Music Group та Warner Music Group.

Система штучного інтелекту служби може поєднувати певні пісні з різними емоціями та діяльністю. Щоб визначити настрій пісні, команда розглянула звуковий сигнал та текст пісні. Для початку вони подавали аудіосигнали в нейронну мережу, а також моделі, які відтворювали мовне значення слів. Для навчання системи визначення настрою пісні використовується база даних Million Song DataSet (MSD), що представляє колекцію метаданих понад мільйон сучасних пісень. Зокрема, використовуйте набір даних Last.fm, який надає понад 500 000 унікальних тегів для ідентифікації пісень. Багато з цих тегів пов'язані з емоціями, і понад 14 000 з них використовуються для визначення негативного чи позитивного сенсу слів.

База даних Million Songs містить лише метадані пісні, а не саму музику, тому розробники повинні використовувати назву пісні, ім’я виконавця та альбом як ідентифікатор, щоб поєднати цю інформацію зі своїм каталогом Deezer. Близько 60% отриманого набору даних використовується для навчання системи, а решта треків використовується для перевірки та подальшого тестування системи. Системи штучного інтелекту краще виявляють спокій або життєву силу пісень, ніж традиційні методи, які не використовують штучний інтелект і роблять те саме. Дослідники вважають, що ефективність можна покращити, аналізуючи та порівнюючи музику та пісні разом. На рисунку 1.12 зображено головну сторінку сервісу «Deezer» [21].

– LinkedIn - це соціальна мережа для бізнесу. Пропонує користувачеві рекомендації щодо знайомих людей, відповідно до його фаху, робочі місця та компанії. Для надання рекомендацій використовується спільна система фільтрації, заснована на технології Apache Hadoop. На рисунку 1.13 зображена сторінка користувача в «LinkedIn» [22].

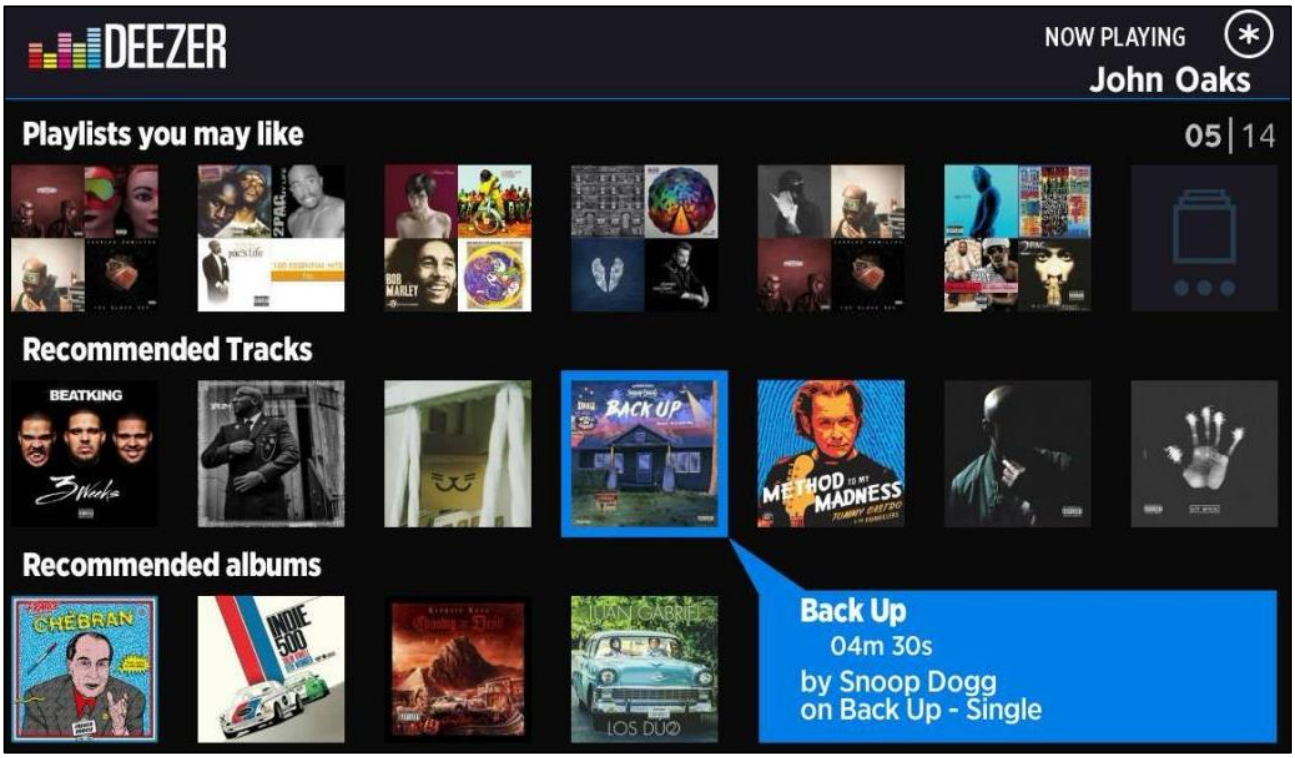


Рисунок 1.12 – Головна сторінка сервісу «Deezer»

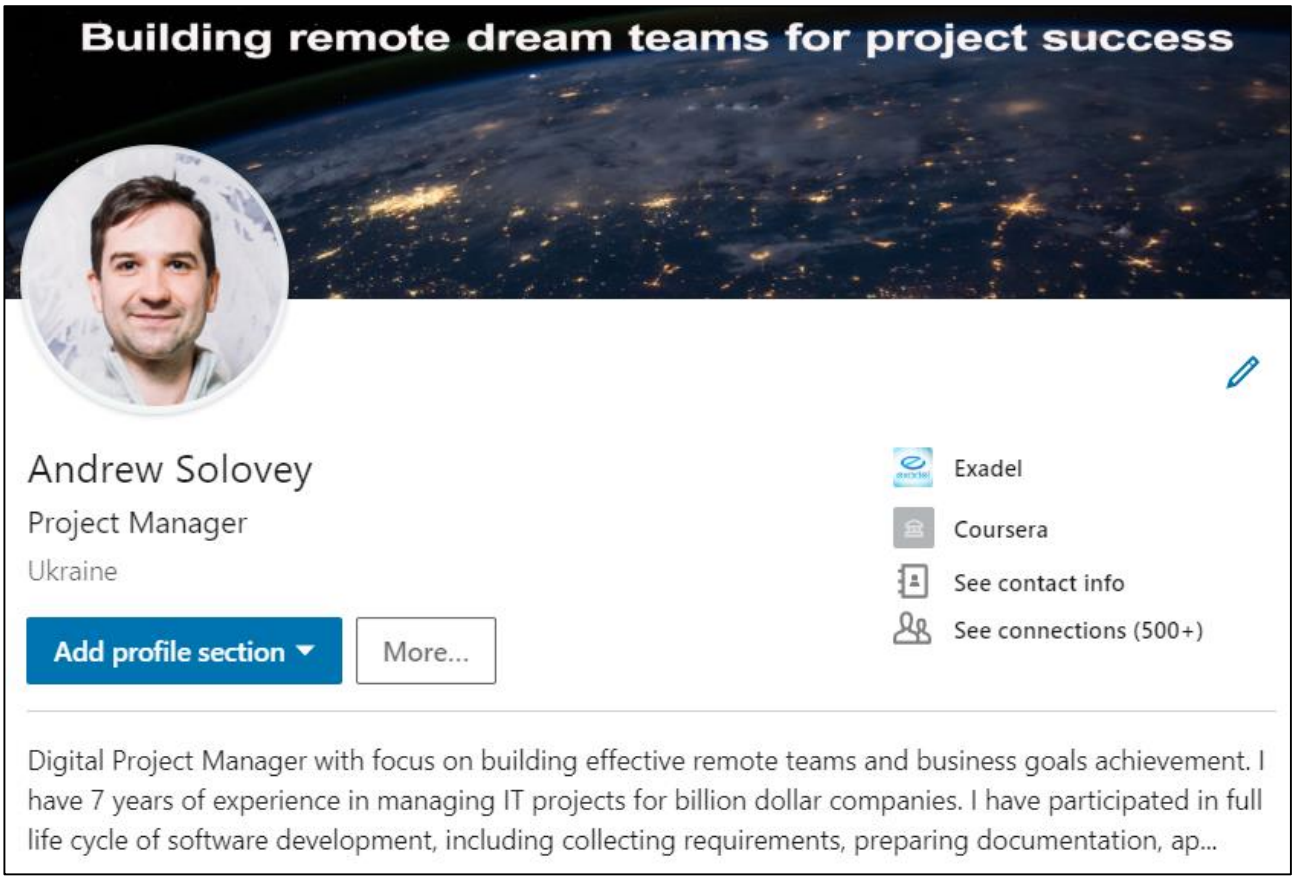


Рисунок 1.13 - Сторінка користувача в «LinkedIn»

– YouTube. Щохвилини на YouTube додається щонайменше 300 годин нового відеоконтенту різного вмісту. Близько 2 мільярдів користувачів щомісяця відвідують платформу і переглядають більше однієї години відео. Для обробки великих обсягів інформації компанія активно використовує функції штучного інтелекту для спрощення певних операцій. На рисунку 1.14 зображено сторінку з рекомендованими відео.

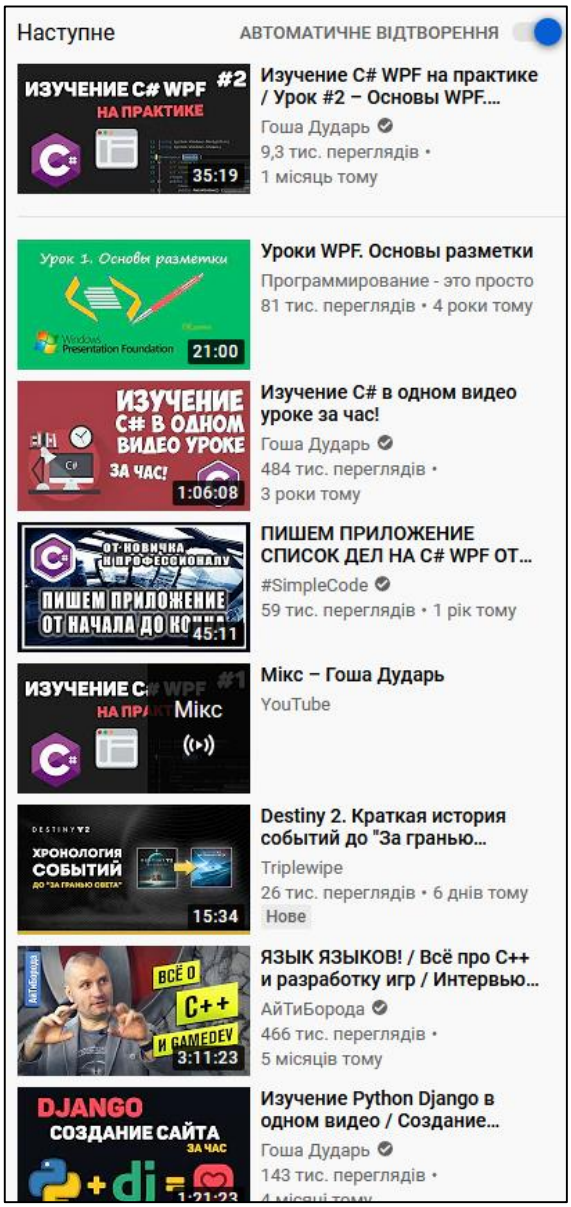


Рисунок 1.14 - Рекомендованими відео на «YouTube»

YouTube має найдосконалішу систему рекомендацій на основі штучного інтелекту. Вона сильно відрізняється від механізму, що використовується на

Netflix, Hulu або Spotify. Рекомендаційна система YouTube постійно обробляє оновлення вмісту та формує рекомендації в режимі реального часу.

Розроблена система складається з двох частин. Перший - це нейронна мережа, метою якої є відбір кандидатів на основі історії перегляду користувачів. Це дозволяє широку персоналізацію та подальшу фільтрацію вмісту за ідентифікаторами, такими як перегляд відео, демографічні показники, пошукові запити.

Друга частина - це рейтингова мережа, яка використовує "опис користувачів та відео", щоб надати оцінки кожному відео. За допомогою дворівневого підходу система може обробляти мільйони відео та рекомендувати змістовний та корисний вміст. Час перегляду - головний показник корисності рекомендацій. Система дуже ефективна платформи, рекламодавців, та користувачів [23].

1.6 Висновки

В даному розділі було сформовано поняття рекомендаційних систем. Розглянуто три основні методи побудови рекомендаціних систем такі як:

– content-basedfiltering (фільтрація на основі вмісту);
– collaborative filtering (спільна фільтрація);
– гібридний підхід - певною мірою використовує дві попередні стратегії.

та визначені їх переваги та недоліки.

Також було розглянуто та проаналізовано існуючі системи рекомендацій.

На основі проаналізованих даних було обрано гібридну стратегію розробки інформаційної технології, як таку яка найкраще підходить для досягнення поставлених цілей.

2 РОЗРОБКА ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ АНАЛІЗУ ТА РЕКОМЕНДУВАННЯ КІНОФІЛЬМІВ ДЛЯ ГЛЯДАЧІВ

2.1 Обгрунтування вибору стратегії створення рекомендаційної системи

На основі інформації розглянутої в першому розділі, найкращою стратегією для створення інформаційної технології аналізу та рекомендації кінофільмів буде гібридна стратегія. Гібридна стратегія поєднує два основні методи побудови рекомендаційних систем фільтрацію на основі вмісту та спільну фільтрацію.

Обрана стратегія дозволить отримати переваги обох вище вказаних методів, а саме:

– незалежність користувача;
– прозорість;
– персональний результат;
– маштабованість.

Також комбінація двох методів дозволить уникнути таких проблем, як:

– холодний старт;
– проблема нового користувача;
– розрідженість даних;
– обмежений аналіз контенту.

Отже, для як стратегію розробки інофрмаційної системи аналізу та рекомендації кінофільмів було обрано гібридну стратегію.

2.2 Аналіз сценаріїв роботи інформаційної технології

Перед початком розробки інформаційної технології аналізу та рекомендації кінофільмів проведемо аналіз можливих сценераціїв її роботи.

Інформаційна технологія має три можливі сценарії роботи.

За першим сценарієм користувач хоче отримати рекомендації лише на основі свого списку рейтингового списку. Для цього йому не потрібно вводити ніяких

додаткових даних, для рекомендації буде використовуватись лише дані системи, з якою працює користувач.

Другий сценарій базується на ідеї, що користувач переглянувший фільм який йому дуже сподобався хоче отримати рекомендацію фільмів, які схожі на нього. Для цього йому потрібно буде ввести назву фільму на основі якого він хоче отримати рекомендацію, далі система видасть рекомендацію проаналізувавши дані цільового фільму.

Третій сценарій дозволить цільовому користувачу конретизувати фільми, які йому будуть рекомендовані. Наприклад, користувач хоче отримати рекомендовані фільми які відносяться до конкретного жанру, або фільми у яких знялися конкретні актори, або які зняв конкретний режисер, або все разом. Для цього йому потрібно ввести ці дані, система їх проаналізує і видасть рекомендацію на їх основі. На рисунку 2.1 зображено діаграму діяльності можливих сценаріїв роботи інтелектуальної системи аналізу та рекомендації кінофільмів.

З рисунку 2.1 видно, що перший сценарій роботи інтелектуальної технології має такі кроки:

Крок 1: Початок роботи;

Крок 2: Створення рекомендації;

Крок 3: Вихідні дані;

Крок 4: Кінець роботи.

Перший сценарій найпростіший, так як не потребує ніяких додаткових даних від корисутувача.

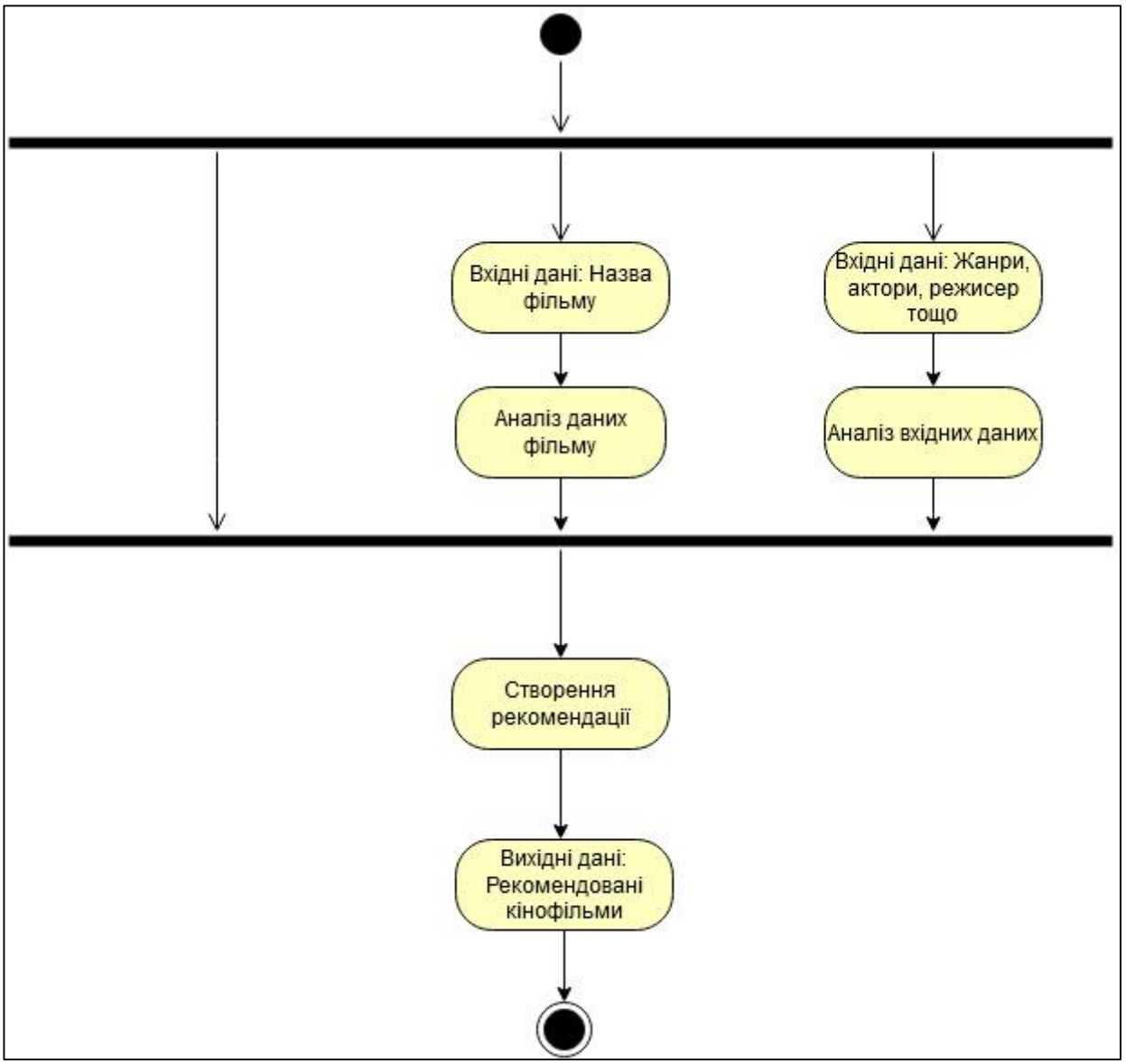


Рисунок 2.1 – Діаграми діяльності можливих сценаріїв роботи інтелектуальної технології аналізу та рекомендації кінофільмів

Другий сценарій – має такі кроки:

Крок 1: Початок роботи;

Крок 2: Вхідні дані(назва філмьму);

Крок 3: Аналіз даних цільового фільму;

Крок 4: Створення рекомендації;

Крок 5: Кінець роботи.

Даний сценарій потребує від користувача введення назви фільму на основі якого він хоче отримати рекомендацію.

Третій сценарій – має такі кроки:

Крок 1: Початок роботи;

Крок 2: Вхідні дані(Жанри, актори. режисер, тощо);

Крок 3: Аналіз вхідних даних;

Крок 4: Створення рекомендації;

Крок 5: Кінець роботи.

Третій сценарій має схожий з другим алгоритм роботи, але в даному сценарії користувачу потрібно ввести конретні дані, які описують фільм такі як (жанри, актори, режисери тощо).

Отже, розробка інтелектуальної технології аналізу та рекомендації кінофільмів базується на трьох основних сценаріях роботи.

2.3 Розробка загальної структури інтелектуальної технології

Розробимо загальну структуру інформаційної технології аналізу та рекомендації кінофільмів.

Так як головною стратегією розробки було обрано гібридну стратегію, буде доречно розділити роботу інформаційної технологію на два модулі, які будуть працювати послідовно. Перший модуль працюватиме на основі методу спільної фільтрації і відповідає за обчислення схожості користувачів та прогнозування оцінки, яку може поставити користувач рекомендованому кінофільму. Другий модуль відповідатиме за метод фільтрації на основі вмісту саме він буде аналізувати вхідні дані і на основі них обирати ті фільми, які схожі найбільше. На рисунку 2.2 зображено загальну структуру інформаційної технології аналізу та рекомендації кінофільмів.

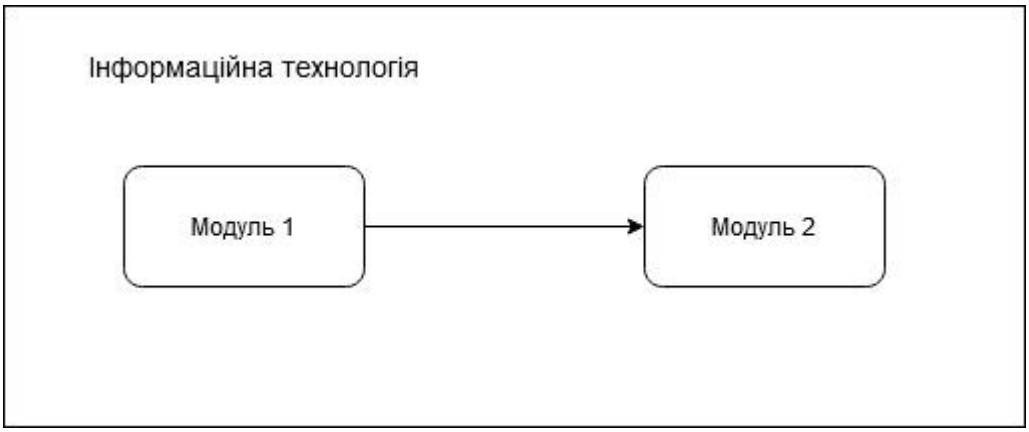


Рисунок 2.2 – Загальна структура інформаційної технології аналізу та рекомендації кінофільмів

Отже, розроблена структура інформаційної технології аналізу та рекомендації кінофільмів складається з двох модулів, які працюють послідовно.

2.4 Розробка модулю спільної фільтрації

Розробимо перший модуль інформаційної технології аналізу та рекомендації кінофільмів, який працюватиме на основі методу спільної фільтрації.

Даний модуль відповідає за обчислення схожості між користувачами та прогнозування оцінки рекомендованому кінофільму.

Спочатку модуль обчислює схожість між цільовим кристувачем і всіма іншими користувачами використовуючи формулу:

$$S = 2 - \sqrt{\frac{\sum_i | y_i - x_i |}{m}}, \quad (2.1)$$

де S – схожість між двома користувачами,

і – номер фільму,

у – оцінка фільму користувача,

x – оцінка фільму цільового користувача,

m – кількість фільмів.

За формулою (2.1) максимальна схожість між користувачами буде дорівнювати 2 (в пятибальній системі оцінвання). Це значення отримується наступним шляхом, послідовно порівнюючи оцінки, які ставив кінофільмам цільовий користувач і користувач з яким порівнюємо, знаходимо суму різниць між ними і ділимо на кількість фільмів, які були порівнянні, потім вираховуємо квадратний корінь отриманого значення і віднімаємо його від двійки. Наприклад, якщо користувач 1 оцінив фільм в 5 балів, а користувач оцінив цей самий фільм в 3 бали значить схожість між ними буде дорівнювати:

$$S = 2 - \sqrt{\frac{|5-3|}{1}} = 0.586.$$

Чим менше заначення S для користувача тим більше він схожий на цільового.

Далі після того, як для усіх користувачів в системі було оцінено значення схожості, на основі цих даних можна прогнозувати оцінку для кінофільмів. Та спочатку потрібно відфільтрувати користувачів, які мають схожість меншу ніж порогове значення, так як вони не мають впливати на прогнозовану оцінку. За замовчування порогове значення буде дорівнювати - 1.2, даний коефіцієнт може змінюватись чим він більший тим більш точніше буде прогнозоване значення, та занабто велике значення даного коефіцієна може видати не достатньо актуальне значення, так як в прогнозування буде враховувати не достатня кількість користувачів.

Для передбачення оцінки використовується формула:

$$P = \sum_i \frac{S_i}{\sum S_i} * x_i, \qquad (2.2)$$

де P – рекомендована оцінка,

i – номер користувача,

S – значення схожості користувача,

x – оцінка фільму користувача.

Формула (2.2) прогнозує оцінку на конкретний фільм для конкретного користувача, для цього ми обираємо усіх кристувачів, які оцінювали цільовий фільм і які мають схожість більше порогового значення, і сумуємо їх вплив на оцінку. Вплив на оцінку вираховується наступним чином, проходячи по користувачах беремо значення їхньої схожості ділимо його його на суму схожості усіх користувачів, які беруть участь у прогнозувані, і множемо на їх оцінку цільовому фільму. Таким чином чим більше користувач схожий з цільовим користувачом тим більший вплив він має на оцінку.

Далі модуль формує список фільмів з їх прогнозованими оцінками для цільового користувача.

На рисунку 2.3 зображено діаграму діяльності першого модулю інформаційної технології аналізу та рекомендації кінофільмів.

Алгоритм роботи першого модулю такий:

Крок 1: Початок роботи;

Крок 2: Вхідні дані;

Крок 3: Обчислення схожості користувачів;

Крок 4: Прогнозування оцінки для фільмів;

Крок 5: Вихідні дані;

Крок 6: Кінець роботи.

Рисунок 2.3 – Діаграма діяльності першого модулю інформаційної технології аналізу та рекомендації кінофільмів

Отже, розроблений перший модуль інформаційної технології аналізу та рекомендації кінофільмів відповідає за аналіз схожості усіх користувачів з цільовим користувачам та за прогнозування оцінок кінофільмам.

2.5 Розробка модулю фільтрації на основі вмісту

Розробимо другий модуль інформаційної технології аналізу та рекомендації кінофільмів, який реалізовуватиме метод фільтрації на основі вмісту.

Даний модуль відповідає за аналіз інформації про кінофільм та знаходженю найбільш схожих.

Як було визначено і підрозділі 2.2 є три варіанта вхідних даних для цього методу:

- Без вхідних даних;
- Назва фільму;
- Інформація про фільм (жанри, актори, режисер тощо).

У першому варіанті даний модуль не буде відпрацьовувати, так як вхідних даних для нього немає.

Розгянемо другий варіант. У ньому модуль в якості вхідних даних отримує назву фільму. В цьому випадку він повинен сам знайти дані про фільму, і на основі них сформувати рекомендацію, тому першим кроком в алгоритмі роботи модулю для даного сценарію буде пошук інформації про фільм. Далі з цієї інформації визначаються три головні актори, три головні жанри, три ключових слова, які описують даний фільм, та режисер. Потім ідентичним чином обробляються усі фільми, що є в системі. Далі за допомогою косинуса подібності (коефіцієнт подібності двох не нульових векторів у предгільбертовому просторі, який обчислюється як косинус кута між ними.) визначаємо, які фільми найбільше схожі на цільовий та виводимо їх. На рисунку 2.4 зображено діаграму діяльності другого варіанту роботи модулю фільтрації на основі вмісту.

Рисунок 2.4 - Діаграма діяльності другого варіанту роботи модулю фільтрації на основі вмісту

Третій варіант роботи модулю в якості вхідних даних отримує детальну інформацію про фільм таку як жанри, актори, режисери тощо. На відміну від

другого варіанту, який розглянутий вище, даному варіанту не потрібно шукати інформацію він одразу може до визначення рекомендації тобто до третього кроку алгоритму роботи другого варіанту модулю фільтрації на основі вмісту. На рисунку 2.5 зображено діаграму трього варіанту роботи модулю фільтрації на основі вмісту.

Рисунок 2.5 – Діаграма діяльності трього варіанту роботи модулю фільтрації на основі вмісту

Отже, розроблений другий модуль інтелектуальної технології аналізу та рекомендації кінофільмів має два основні варіанти роботи. На рисунку 2.6

зображено діаграму діяльності загального алгоритму роботи модулю фільтрації на основі вмісту.

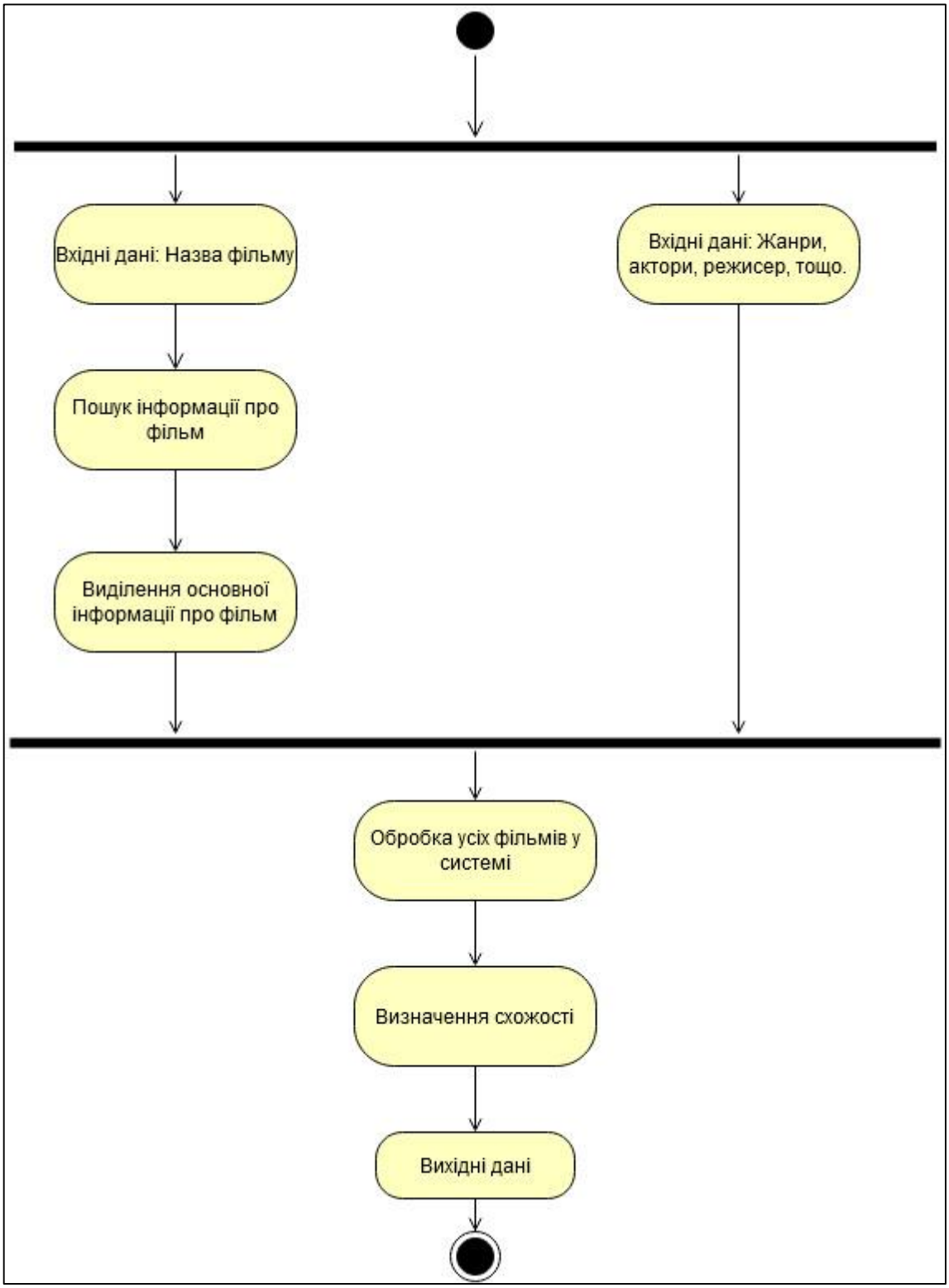


Рисунок 2.6 - Діаграма діяльності загального алгоритму роботи модулю фільтрації на основі вмісту

2.6 Висновки

В даному розділі було розроблено інформаційну систему аналізу та рекомендації кінофільмів для глядачів.

На основі проаналізованих даних першого розділу було обрано гібридну стратегію розробки інформаційної технологію, як таку яка найкраще підходить для отримання поставлених цілей.

Було розроблено загальну структуру інформаційної технології аналізу та рекомендації кінофільмів для глядачів, яка скаладається з двох модулів, які працюють послідовно.

Було розроблено два модулі інформаційної технології аналізу та рекомендації кінофільмів для глядачів, модуль спільної фільтрації та модуль фільтрації на основі вмісту.

3 РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ АНАЛІЗУ ТА РЕКОМЕНДУВАННЯ КІНОФІЛЬМІВ ДЛЯ ГЛЯДАЧІВ

Реалізуємо програмно розроблену у попередньому розділі інформаційну технологію аналізу та рекомендації кінофільмів

3.1 Вибір мови програмування

Для програмної реалізації інформаційної технології аналізу та рекомендації кінофільмів було обрано мову програмування Python.

Python - інтерпретована об'єктно-орієнтована мова програмування високого рівня зі строгим динамічним набором тексту. Інтерпретатор можна використовувати в інтерактивному режимі, що дозволяє спробувати мовні функції, написати шаблони програм або протестувати функції розробника знизу вгору. Як настільний калькулятор, це також дуже зручно. Python дозволяє писати дуже компактні та зручні для читання програми. Програми, написані на Python, як правило, набагато коротші, ніж еквіваленти на C або C ++ з наступних причин:

– Типи даних високого рівня дозволяють виразити складну роботу в простій інструкції;

– Використовує відступ замість фігурних дужок для виконання інструкцій групування;

– Не потрібно оголошувати зміни;

Python - це сучасна мова, яка в порівнянні з іншими (включаючи PHP та Java) забезпечує швидкі можливості розробки, що робить розробників більш економічними. Сама мова має інтуїтивно зрозумілу граматику та хорошу бібліотеку класів, що дозволяє зосередитись на головному завданні, замість того, щоб писати багато різних абстракцій. У базі даних мовної бібліотеки вже існує велика кількість необхідних абстракцій, вони перевірені та стабільні. Це призводить до високої якості продукції та короткого часу розробки.

У більшості випадків швидкість коду, написаного на Python, залежить не тільки від якості коду, але і від завданнь які він вирішує. Python добре працює в однопотокових програмах (швидкість пошуку швидша, ніж і інших популярних мовах, що інтерпретуються: PHP, Perl, Ruby тощо), але, звичайно, це не швидше, ніж скомпільована мова C. Однак рішенням цих проблем буде спеціально створений Cython. Для складних розрахунків, які вимагають високої швидкості та високого опору навантаженню, ми можемо використовувати підключення Python + NumPy або перенести код у модулі C для досягнення необхідної швидкості. Для багатопотокових програм ми використовуємо лише один багатопроцесорний модуль, що робить багатопотокові програми швидшими порівняно з іншими мовами сценаріїв.

Python має чудову документацію, яка може зменшити самостійне вивчення, розробку та підтримку проектів [24].

3.2 Аналіз датасету

Для реалізації та тестування інтелектуальної технології аналізу та рекомендації кінофільмів було обрано датасет «The Movies Dataset». Дослідимо та нормалізуємо його.

Даний датасет містить 7 файлів, а саме:

а) credits.csv;

б) keywords.csv;

в) links.csv;

г) links_small.csv;

д) movies_metadata.csv;

е) ratings.csv;

ж) ratings_small.csv.

Файли «links.csv» і «links_small.csv», містять інформацію, яка дозволяє поєднувати однакові фільми з різних ресурсів, таких як «Rotten tomatoes», «TMDB» і «IMDB».

Файл «credits.csv», «keywords.csv», «movies_metadata.csv» містять дані про фільми, а саме:

а) credits.csv:

1) id – ідентифікатор фільму;

2) cast – дані про акторський склад фільму;

3) crew – дані про команду, яка працювала над фільмом (режисер, продюсер, оператор і т.д.).

б) keywords.csv:

1) id – ідентифікатор фільму;

2) keywords – ключові слова, які описують фільм.

в) movies_metadata.csv:

1) id – ідентифікатор фільму;

2) adult – чи має фільм дорослий рейтинг;

3) belongs_to_collection – чи належить фільм до якого небудь циклу;

4) budget – бюджет фільму;

5) genres – жанри фільму;

6) homepage – посилання на домашню сторінку фільму;

7) imdb_id – ідентифікатор фільму на сайті «IMDB»;

8) original_language – оригінальна мова якою був знятий фільм;

9) original_title – назва рідною мовою фільму;

10) overview – короткий опис фільму;

11) popularity – популярність фільму;

12) poster_path – посилання на постер фільму;

13) production_companies – компанія, яка створила фільм;

14) production_countries – країна в якій був створений фільм;

15) release_date – дата релізу фільму;

16) revenue – дохід фільму (скільки грошей він заробив у прокаті);

17) runtime – тривалість фільму;

18) spoken_languages – мова якою розмовляють герої фільму;

19) status – статус фільму (в прокаті, закінчився прокат, ще знімається);

20) tagline – теги фільму;
21) title – назва фільму;
22) vote_average – середня оцінку фільму;
23) vote_count – к-сть користувачів, які оцінили фільм.

Файли «ratings.csv» і «ratings_small.csv» містять такі дані:

а) userId – ідентифікатор користувача;
б) movieId – ідентифікатор фільму;
в) rating – оцінка фільму, яку поставив користувач.

Перед тим, як почати аналіз даних, об’єднаємо файли «credits.csv», «keywords.csv» та «movies_metadata.csv» в один датафрейм, для зручності подальшого маніпулювання ними. На рисунку 3.1 зображено фрагмент коду, який конвертує дані файлів у датафрейми і об’єднує їх в один.

```
credits = pd.read_csv('../input/the-movies-dataset/credits.csv')
moviesMetaData = pd.read_csv('../input/the-movies-dataset/movies_metadata.csv',low_memory=False)
keywords = pd.read_csv('../input/the-movies-dataset/keywords.csv')

mainList= pd.merge(moviesMetaData, credits, on='id')
mainList= pd.merge(mainList,keywords, on='id')
```

Рисунок 3.1 – Фрагмент коду, який об’єднує дані у датафрейм

На рисунку 3.2 зображено фрагмент коду, який зображує heatmap і гістограму коефіцієнту заповнення.

```
def missingDF(data):
    missing_df = data.isnull().sum(axis=0).reset_index()
    missing_df.columns = ['column_name', 'missing_count']
    missing_df['filling_factor'] = (mainList.shape[0]
                                    - missing_df['missing_count']) / mainList.shape[0] * 100
    missing_df.sort_values('filling_factor').reset_index(drop = True)

    missing_df = missing_df.sort_values('filling_factor').reset_index(drop = True)
    y_axis = missing_df['filling_factor']
    x_label = missing_df['column_name']
    x_axis = missing_df.index

    fig = plt.figure(figsize=(11, 4))
    plt.xticks(rotation=80, fontsize = 14)
    plt.yticks(fontsize = 13)

    plt.xticks(x_axis, x_label,family='fantasy', fontsize = 14 )
    plt.ylabel('Filling factor (%)', family='fantasy', fontsize = 16)
    plt.bar(x_axis, y_axis);

    return missing_df

sn.heatmap(mainList.isnull(), yticklabels = False, cbar = False, cmap = 'viridis')
missingDF(mainList)
```

Рисунок 3.2 – Фрагмент коду, який зображує heatmap і гістограму коефіцієнту заповнення

Спочатку проаналізуємо фактор заповнення даних. На рисунку 3.3 зображено «heatmap» коефіцієнту заповнення.

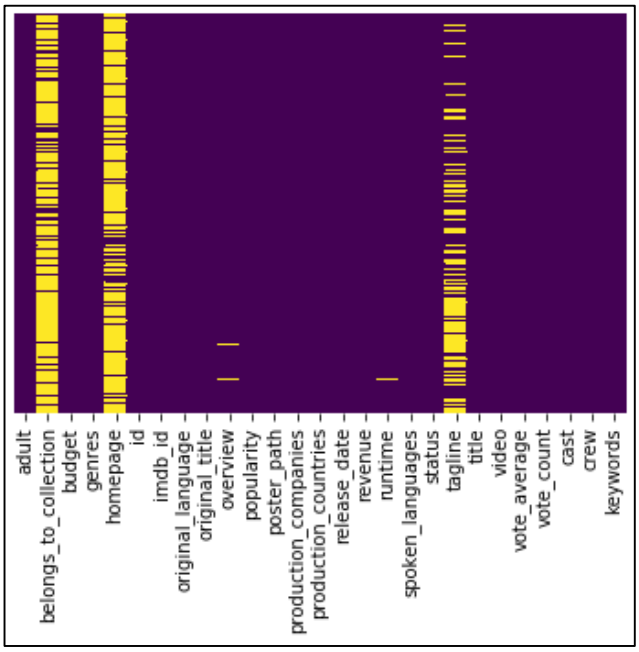


Рисунок 3.3 – Heatmap фактору заповнення

З рисунку 3.3 видно, що такі атрибути, як «belongs_to_collection», «homepage», «overview», «runtime» та «tagline» заповнено не повністю.

На рисунку 3.4 зображена гістограма фактору заповнення.

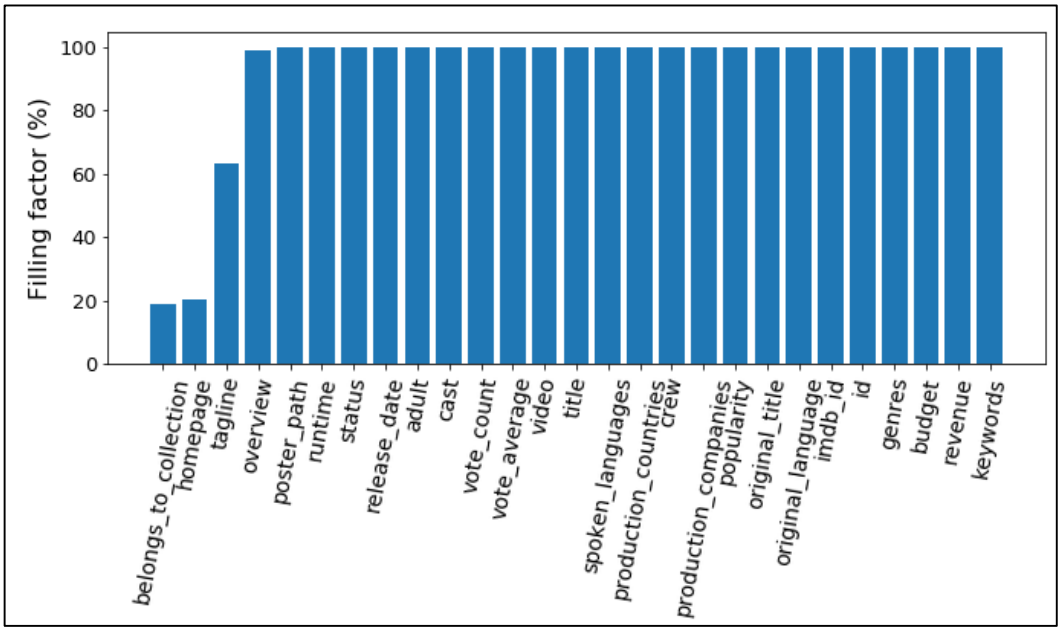


Рисунок 3.4 – Гістограма фактору заповнення

З гістограми, яка зображена на рисунку 3.4 можна оцінити фактор заповнення у відсотках. Атрибути «belongs_to_collection», «homepage» мають фактор заповнення менше 90%, що робить їх використання у даній магістерській роботі неможливим. Атрибути «overview» і «runtime» мають фактор заповнення більше 99% тому використання їх у магістерській роботі є прийнятним. Усі інші атрибути мають фактор заповнення 100%.

Проаналізуємо, яка кількість користувачів оцінювала фільми по роках. На рисунку зображено фрагмент коду, який аналізує дані кількості користувачів, які оцінили фільми.

На рисунку 3.5 наведено фрагмент коду, який аналізує залежність кількості користувачів, які оцінили фільми, до років виходу фільмів. На рисунку 3.6 зображено самий графік.

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

mainList[(mainList['years'] < 2019) & (mainList['years'] >= 1950)].groupby(by = 'years').mean()['vote_count'].plot()
```

Рисунок 3.5 – Фрагмент коду, який аналізує залежність кількості користувачів, які оцінили фільми, до років виходу фільмів

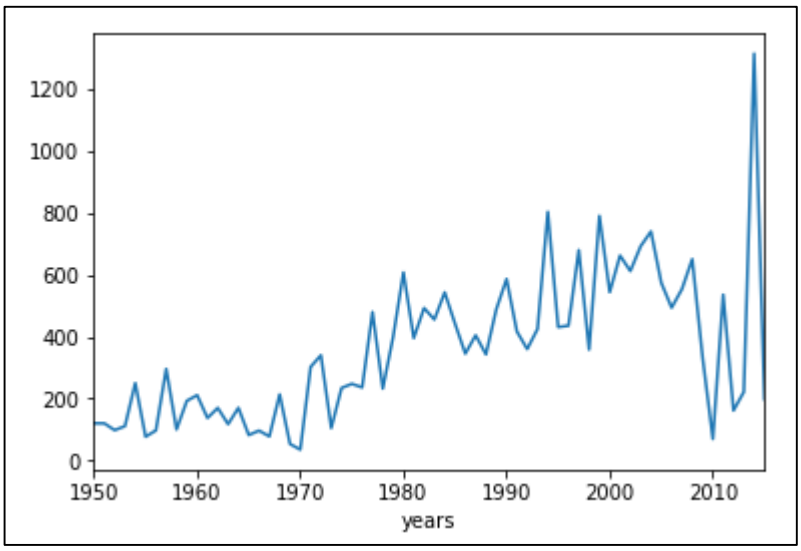


Рисунок 3.6 – Графік залежності кількості користувачів, які оцінили фільм до року виходу фільму

З графіку, який зображений на рисунку 3.6 можна зробити висновок, що фільми, які вийшли в останні 20 років були оцінені найбільшою кількістю користувачів.

Побудуємо графік кореляції між критеріями. На рисунку 3.7 зображено фрагмент коду, який будує теплову мапу кореляційної матриці. На рисунку 3.8 зображено heatmap кореляційної матриці.

```
mainList['budget'] = mainList['budget'].astype(float)
mainList['popularity'] = mainList['popularity'].astype(float)
sn.heatmap(mainList.corr(), annot=True)
```

Рисунок 3.7 – Фрагмент коду, який будує теплову мапу кореляційної матриці

Рисунок 3.8 – Heatmap кореляційної матриці

Виходячи з вищенаведеної теплової карти, можна сказати, що більшість функцій знаходяться у високій кореляції один з одним.

Проаналізуємо популярність жанрів. На рисунку 3.9 зображено фрагмент коду, який будує WordCloud і гістограму даних. На рисунку 3.10 зображено WordCloud і гістограму популярності жанрів кінофільмів.

```
def createWordCloud(data):
    wordcloud = WordCloud(max_font_size=100)

    wordcloud.generate_from_frequencies(data)

    plt.figure(figsize=[10.1,10.1])
    plt.imshow(wordcloud, interpolation="bilinear")
    plt.axis("off")

    plt.show()
    genres_count = pd.Series(data)
    genres_count.sort_values(ascending = False).head(20).plot(kind = 'bar', grid='True')
```

Рисунок 3.9 – Фрагмент коду, який будує WordCloud і гістограму даних

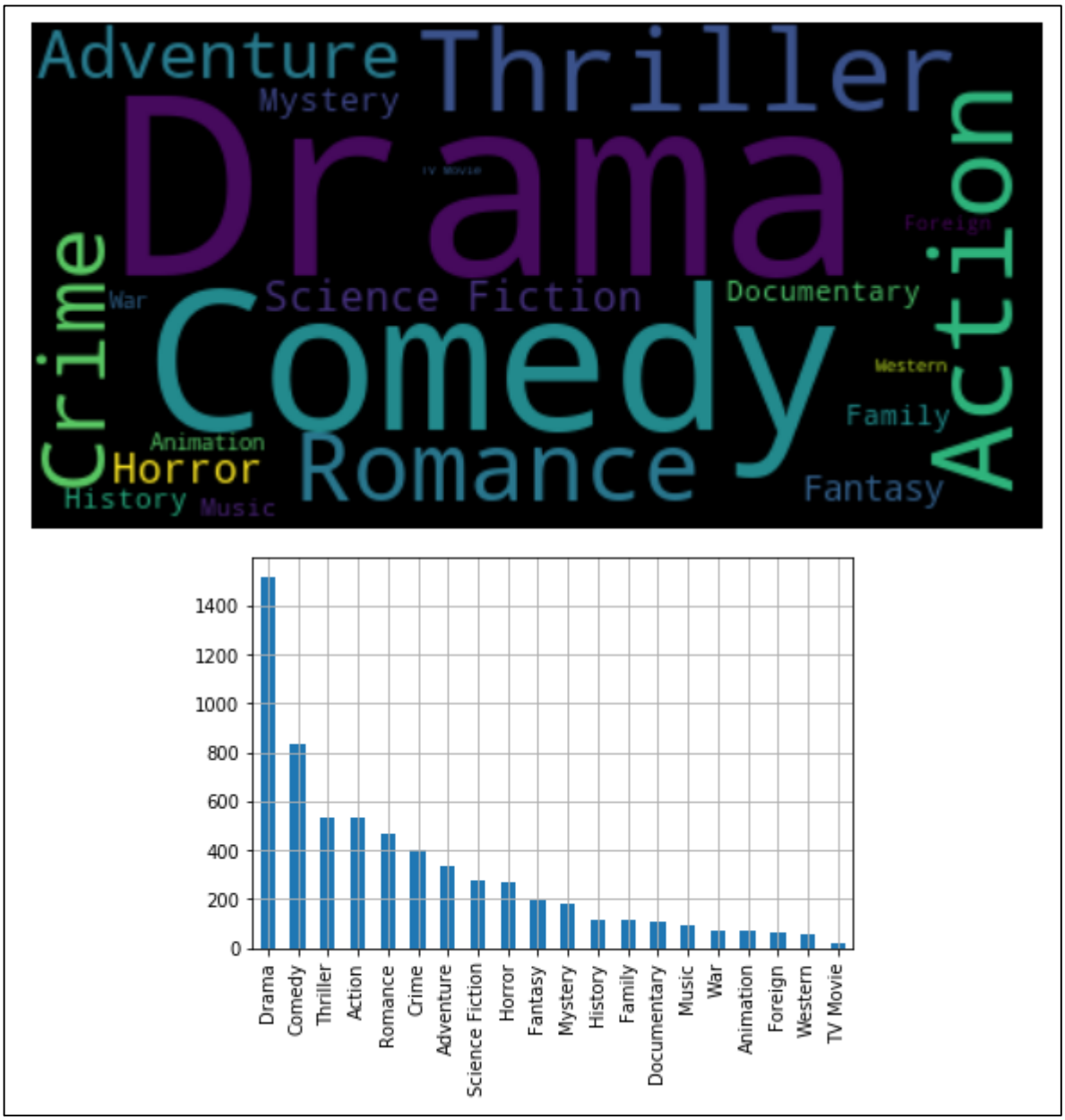


Рисунок 3.10 – WordCloud і гістограма популярності жанрів кінофільмів

З рисунку 3.10 видно, що найбільш популярним жанром кінофільму є драма, вона майже вдвічі популярніше наступного по популярності жанру - комедії.

Було проаналізовано найбільш популярних акторів. На рисунку 3.11 зображено WordCloud і гістограму популярності кіноакторів.

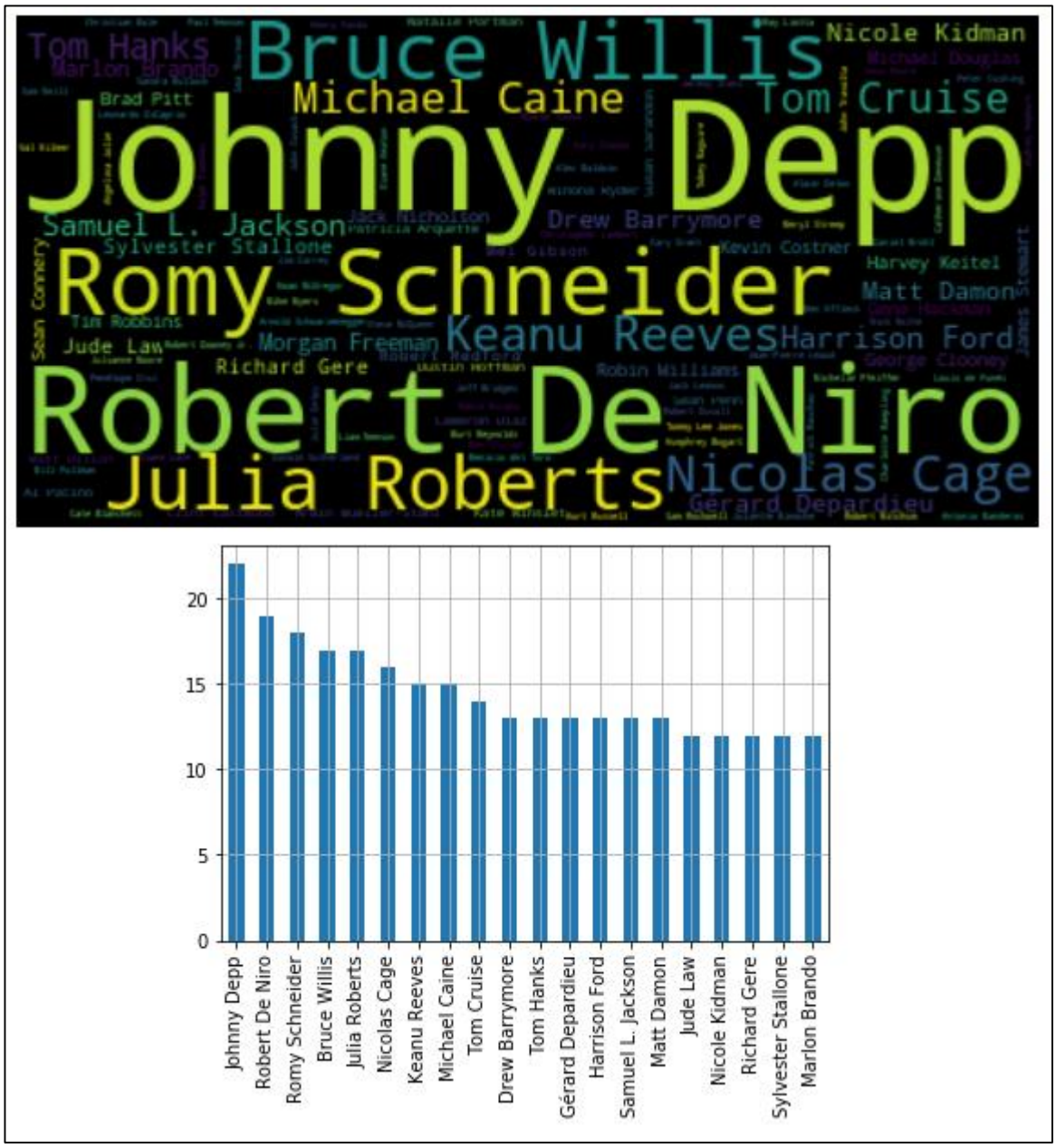


Рисунок 3.11 – WordCloud і гістограма популярності кіноакторів

З рисунку 3.11 видно, що найбільш популярним актором є Джоні Депп, наступним по популярності є Роберт де Ніро, хоча різниця між ними всього кілька фільмів.

Було проаналізовано найбільш популярних кінорежисерів. На рисунку 3.12 зображено WordCloud і гістограму популярності кінорежисерів.

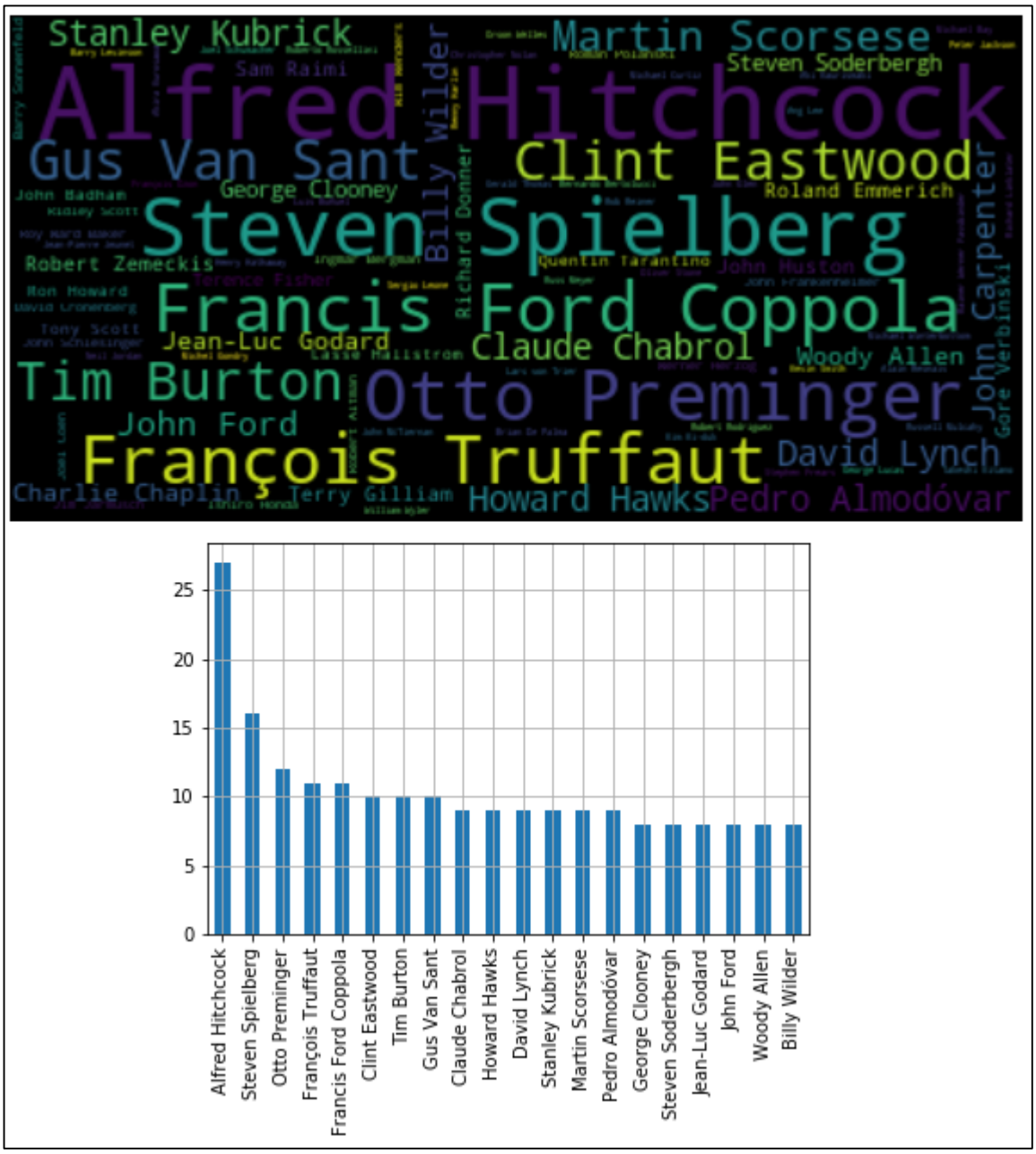


Рисунок 3.12 – WordCloud і гістограма популярності кінорежисерів

З рисунку 3.12 видно, що найбільш популярним кінорежисером є Альфред Хічкок, він майже вдвічі випереджає наступного по популярності кінорежисера Стівена Спілберга.

Було проаналізовано ключові слова, які найчастіше використовуються для опису кінофільмів. На рисунку 3.13 зображено WordCloud і гістограму ключових слів, які найчастіше використовуються для опису кінофільмів.

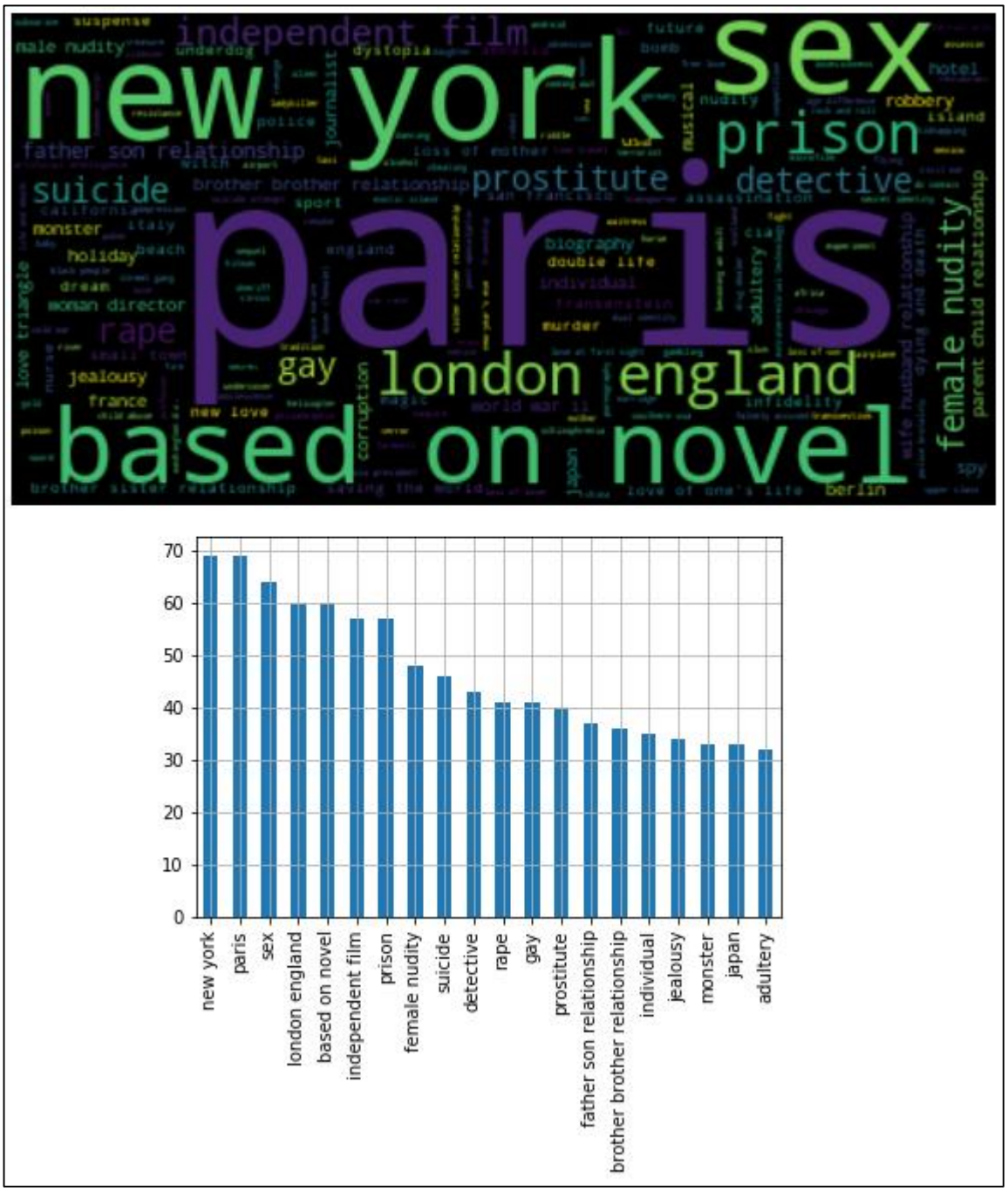


Рисунок 3.13 – Зображено WordCloud і гістограму ключових слів, які найчастіше використовуються для опису кінофільмів

З рисунку 3.13 видно, що найбільш популярними ключовими словами є Нью-йорк і Париж.

Проаналізувавши дані та беручи до уваги розроблений у підрозділі 2.5 модуль оберемо такі ключові дані для побудови рекомендації, а саме

– актори;

– режисер;

– ключові слова;

– жанри.

Нормалізуємо вищенаведені дані, і проаналізуємо їх коефіцієнт заповнення.

На рисунку 3.14 зображено фрагмент коду, який нормалізує вищенаведені дані.

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

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

def combineKeywords(x):
    return ' '.join(x['keywords']) + ' ' + ' '.join(x['cast']) + ' ' + x['director'] + ' ' + ' '.join(x['genres'])

mainList['myKeywords'] = mainList.apply(combineKeywords, axis=1)

mainList['myKeywords'].head(5)
```

Рисунок 3.14 – Фрагмент коду, який нормалізує дані

На рисунку 3.15 зображено теплову мапу і гістограму коефіцієнту заповнення нормалізованих даних.

З рисунку 3.15 видно, що коефіцієнт заповнення усіх нормалізованих даних сягає 100%.

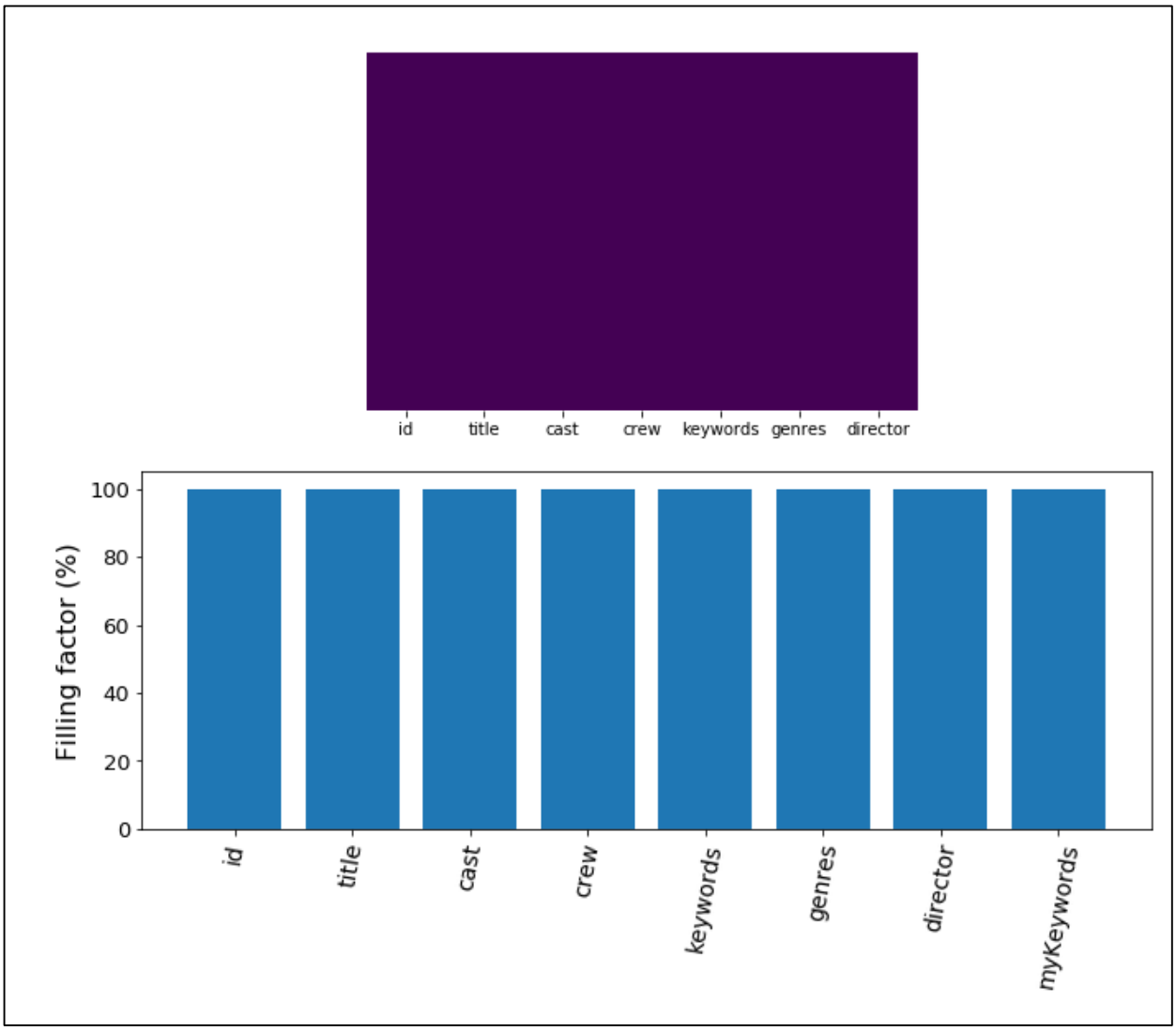


Рисунок 3.15 – Теплова мапа і гістограма коефіцієнту заповнення нормалізованих даних

3.3 Реалізація модулю спільної фільтрації

Реалізуємо розроблений у підрозділі 2.4 модуль спільної фільтрації за допомогою мови програмування Python.

Спочатку створимо датафрейм з даними рейтингу користувачів. Для цього за допомогою функції «read_csv» бібліотеки Pandas прочитаємо файл «ratings_small.csv». На рисунку 3.16 зображено код перетворення даних в датафрейм.

```
ratings = pd.read_csv('../input/the-movies-dataset/ratings_small.csv')
```

Рисунок 3.16 – Фрагмент коду створення датафрейму

Далі створимо функцію, яка буде обраховувати схожість користувачів за формулою (2.1), яка була розроблена у підрозділі 2.4. На рисунку 3.17 зображено код даної функції.

```
def GetUserSimilarity(df, targetUserDic):
    userRatingSum = 0.0
    numberMovies = 0
    for row in df.itertuples(index=False, name='Pandas'):
        if targetUserDic.get(row.movieId) is not None:
            userRatingSum += abs(float(row.rating) - float(targetUserDic[row.movieId]))
            numberMovies += 1

    if (userRatingSum != 0.0 and numberMovies != 0):
        df['similarity'] = 2 - np.sqrt(userRatingSum/numberMovies)
    else:
        df['similarity'] = 0

    return df
```

Рисунок 3.17 – Код функції, яка обраховує схожість користувачів

Функція «GetUserSimilarity» приймає два аргументи, датафрейм з оцінками користувача, з яким йде порівняння цільовго користувача, та масив даних «Dictionary» в якому ключ відповідає ідентифікатору фільма, а значення – оцінки фільму, яку поставив цільовий користувач. Далі функція обраховує значення схожості за формулою (2.1) та повертає датафрейм з новим стовбцем даних «similarity» в якому і зберігається це значення.

Для того, щоб обрахувати занчення схожості цільового користувача зі всіма користувачами в системі, використаємо код який зображено на рисунку 3.18.

```
UsersSimilarity = ratings.groupby(['userId'], group_keys=False, as_index=False).apply(GetUserSimilarity, targetUserDic)
```

Рисунок 3.18 – Фрагмент коду який обраховує схожість користувачів

Код який зображено на рисунку 3.18 працює наступним чином. Усі дані які були отримані після результату виконання коду, який зображено на рисунку 3.16, ми групуємо за атрибутом даних «userId», потім за допомогою функуції «apply» до кожної групи даних застосовуємо функцію «GetUserSimilarity», яка описана вище. В результаті отримуємо датафрейм із значеннями схожості користувачів з цільовим користувачем.

Наступним кроком буде реалізація функції, яка буде прогнозувати оцінку рекомендованому фільму для цільового користувача. На рисунку 3.19 зображено код даної функції.

```
def GetMovieRatingForecast(movieGroup):
    predictRating = 0.0
    nMovie = 0
    sumSimilarity = movieGroup.similarity.sum()
    for row in movieGroup.itertuples(index=False, name='Pandas'):
        if sumSimilarity != 0:
            predictRating += ((row.similarity * 100) / (sumSimilarity * 100)) * float(row.rating)
            nMovie += 1

    cols = ['rating', 'nusers']
    return pd.Series((predictRating, nMovie), index=cols)
```

Рисунок 3.19 – Код функції для прогнозування оцінки

Функція «GetMovieRatingForecast» код якої зображено на рисунку 3.19, реалізує прогнозування оцінки фільму для цільового користувача, використовуючи формулу (2.3) з підрозділу 2.4. Дана функція отримує один аргумент, групу користувачів із значеннями схожості та їх оцінками фільму для якого відбувається прогнозування та повертає прогнозовану оцінку і кількість користувачів на основі яких відбувалось прогнозування. На рисунку 3.20 зображено код використання функції «GetMovieRatingForecast».

```
MovieRatingForecast = UsersSimilarity[UsersSimilarity.similarity > 1]
                        .groupby(['movieId'], as_index=False)
                        .apply(GetMovieRatingForecast)
```

Рисунок 3.20 – Код застосування функції «GetMovieRatingForecast»

Групуючи дані, які ми отримали після виконання коду, який зображено на рисунку 3.18, по атрибуту «movieId» за допомогою функції «apply» ми виконуєм функцію «GetMovieRatingForecast» для кожної групи. Таким чином ми отримуємо датафрейм з ідентифікатором фільму прогнозованою оцінкою та кількостю користувачів на основі даних яких відбувалось пронозування.

На рисунку 3.21 зображено повний код реалізації розробленого у підрозділі 2.4 модулю спільної фільтрації.

```
targetUserData = ratings[ratings.userId == userId]
targetUserDic = dict(zip(targetUserData.movieId, targetUserData.rating))

UsersSimilarity = ratings.groupby(['userId'], group_keys=False, as_index=False)
                         .apply(GetUserSimilarity, targetUserDic)

UsersSimilarity = UsersSimilarity.drop(UsersSimilarity[UsersSimilarity.similarity < 1.2].index)

MovieRatingForecast = UsersSimilarity[UsersSimilarity.similarity > 1]
                        .groupby(['movieId'], as_index=False)
                        .apply(GetMovieRatingForecast)
```

Рисунок 3.21 – Код реалізації модулю спільної фільтрації

3.4 Реалізація модулю фільтрації на основі вмісту

Реалізуємо розроблений у підрозділі 2.5 модуль фільтрації на основі вмісту за допомогою мови програмування Python.

Як було визначено в підрозілі 2.5 даний модуль має два основних варіанта роботи, реалізуємо перший з них, коли користувач вводить назву фільму.

В даному випадку нам потрібно визначити основні дані про фільм, а саме три головних жанри, трьох головних акторів, три головні ключові слова, які описують фільм та режисера. На рисунку 3.22 зображено функцію за допомогою якої визначаються головні жанри, актори та ключові слова.

```
def getFirstThree(x):
    if isinstance(x, list):
        names = [i['name'] for i in x]

        if len(names) > 3:
            names = names[:3]
        return names

    return []
```

Рисунок 3.22 – Код для визнаяення головних ознак кінофільму

Метод «getFirstThree» повертає три перші значення з масиву який йому передається в параметрах. Так як в датасеті «The Movies Dataset» основні ознаки фільму вже відсортовані по важливості ми може просто застосувати дану функцію і отримати ту інформацію, яка нам потрібна.

Для отримання режисера реалізуємо окрему функцію, так як в датасеті дані про команду, яка знімала фільм зберігаються разом. На рисунку 3.23 зображено код функція, яка знаходить режисера фільму.

```
def getDirector(x):
    for i in x:
        if i['job'] == 'Director':
            return i['name']
    return np.nan
```

Рисунок 3.23 – Код функція, яка знаходить режисера

Функція «getDirector» знаходить ім’я режисера по ключому слову «Director» та повертає його.

На рисунку 3.24 зображено застосування двох вище описаних функцій.

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

features = ['cast', 'keywords', 'genres']
for feature in features:
    mainList[feature] = mainList[feature].apply(getFirstThree)
```

Рисунок 3.24 – Код застосування функцій «getDirector» та «getFirstThree»

Для реалізації другого варіанту роботи модулю фільтрації на основі вмісту використаєм коди який зображено на рисунку 3.25.

```
new_row =
{
    'id':'0',
    'title':'Current',
    'cast':cast,
    'keywords':keywords,
    'genres':genres,
    'rating':0.0,
    'nusers':0,
    'director':director}
mainList = mainList.append(new_row, ignore_index=True)
```

Рисунок 3.25 – Код додання іноформації про фільм до датафрейму

Код зображений на рисунку 3.25 додає до датафрейму дані, які ввів користувача, таким чимом ми можеме використовувати ці дані, як фільм, тим самим відпадає потреба в реалізації спеціального методу який буде обробляти даний випадок. В якості назви фільму ми використовуєм слово «Current», щоб в майбутньому можна було однозначно ідентифікувати додані дані.

Щоб використати метод косинуса подібності для знаходження схожих фільмів нам потрібно передставити дані фільмів у вигляді вектору. Для цього за допомогою методу «deletingSpaces», код якого зображено на рисунку 3.26, перетворимо підготовлені вище дані в один рядок, та задопомогою методу «CountVectorizer» бібліотеки «sklearn» представимо їх у вигляді вектору.

```
def deletingSpaces(x):
    if isinstance(x, list):
        return [str.lower(i.replace(" ", "")) for i in x]
    else:

        if isinstance(x, str):
            return str.lower(x.replace(" ", ""))
        else:
            return ''
```

Рисунок 3.26 – Код реалізації методу «deletingSpaces»

Далі за допомогою методу «cosine_similarity» бібліотеки «sklearn», який обраховує косинус подібності, знаходимо даний коефіцієнт для усіх фільмів що є в системі.

На рисунку 3.27 зображено код реалізації модулю фільтрації на основі вмісту.

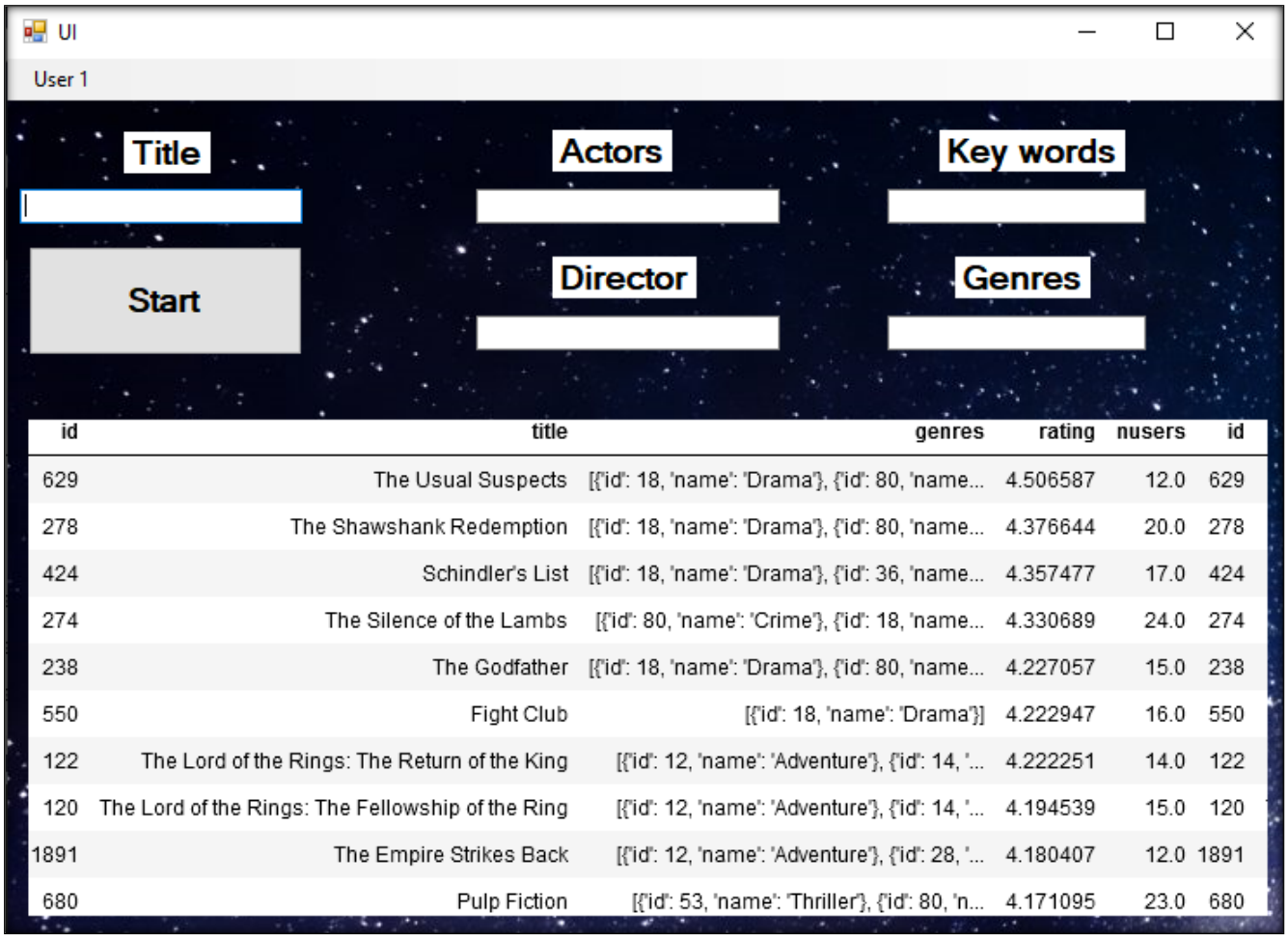
```
mainList = mainList[['id', 'title', 'cast', 'crew', 'keywords', 'genres', 'rating', 'nusers']]

features = ['cast', 'crew', 'keywords', 'genres']
for feature in features:
    mainList[feature] = mainList[feature].apply(literal_eval)

mainList['director'] = mainList['crew'].apply(getDirector)
del mainList['crew']

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

if title == '' and (genres != [''] or director != '' or cast != [''] or keywords != ['']):
    new_row = {'id':'0', 'title':'Current', 'cast':cast, 'keywords':keywords, 'genres':genres, 'rating':0.0, 'nusers':0, 'director':director}
    mainList = mainList.append(new_row, ignore_index=True)

features = ['cast', 'keywords', 'director']

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

mainList['myKeywords'] = mainList.apply(combineKeywords, axis=1)

count = CountVectorizer(stop_words='english')
count_matrix = count.fit_transform(mainList['myKeywords'])
cosine_sim = cosine_similarity(count_matrix, count_matrix)

mainList = mainList.reset_index()
indices = pd.Series(mainList.index, index=mainList['title'])

if title != '':
    idx = indices[title]
else:
    idx = indices['Current']

sim_scores = list(enumerate(cosine_sim[idx]))

sim_scores = sorted(sim_scores, key=lambda x: x[0], reverse=False)

sim_value = [x[1] for x in sim_scores]

mainList['cosine_sim'] = sim_value

return mainList
```

Рисунок 3.27 – Код реалізації модулю фільтрації на основі вмісту

3.5 Тестування роботи реалізованої технології

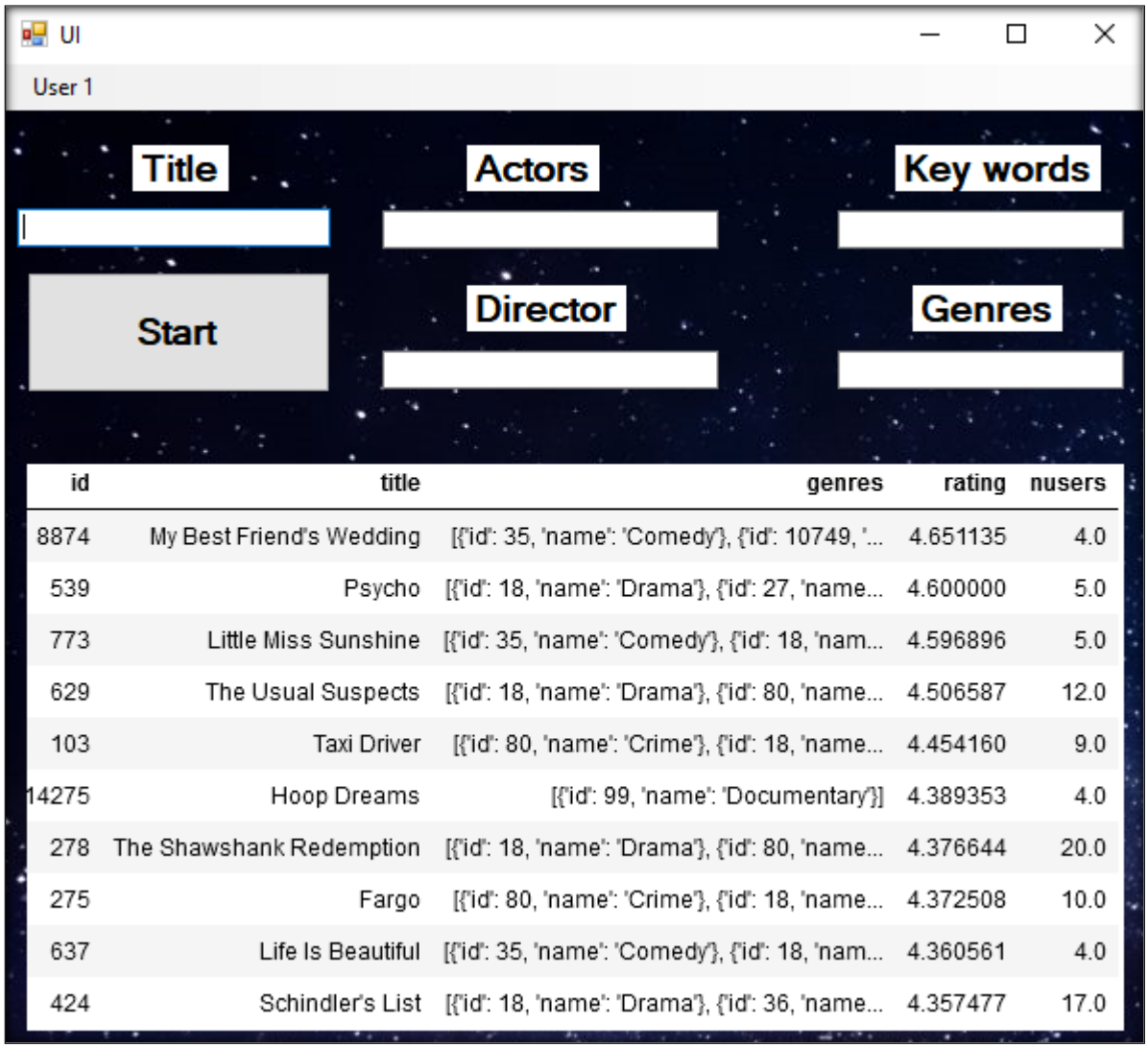
Проведемо тестування реалізованої інтелектуальної технології аналізу та рекомендації кінофільмів. Для цього створимо метод «GetMovieRecommendation» який буде приймати дані від користувача та повертати список рекомендованих фільмів.

Спочатку перевіримо перший сценрій роботи інтелектуальної технології, коли користувач не передає ніяких додаткових даних. На рисунках 3.28 та 3.29 зображено результати роботи даного сценарію для різних користувачів



id	title	genres	rating	nusers	id
629	The Usual Suspects	[{'id': 18, 'name': 'Drama'}, {'id': 80, 'name...	4.506587	12.0	629
278	The Shawshank Redemption	[{'id': 18, 'name': 'Drama'}, {'id': 80, 'name...	4.376644	20.0	278
424	Schindler's List	[{'id': 18, 'name': 'Drama'}, {'id': 36, 'name...	4.357477	17.0	424
274	The Silence of the Lambs	[{'id': 80, 'name': 'Crime'}, {'id': 18, 'name...	4.330689	24.0	274
238	The Godfather	[{'id': 18, 'name': 'Drama'}, {'id': 80, 'name...	4.227057	15.0	238
550	Fight Club	[{'id': 18, 'name': 'Drama'}]	4.222947	16.0	550
122	The Lord of the Rings: The Return of the King	[{'id': 12, 'name': 'Adventure'}, {'id': 14, '...	4.222251	14.0	122
120	The Lord of the Rings: The Fellowship of the Ring	[{'id': 12, 'name': 'Adventure'}, {'id': 14, '...	4.194539	15.0	120
1891	The Empire Strikes Back	[{'id': 12, 'name': 'Adventure'}, {'id': 28, '...	4.180407	12.0	1891
680	Pulp Fiction	[{'id': 53, 'name': 'Thriller'}, {'id': 80, 'n...	4.171095	23.0	680

Рисунок 3.28 – Рекомендовані фільми для користувача з ідентифікатором 1

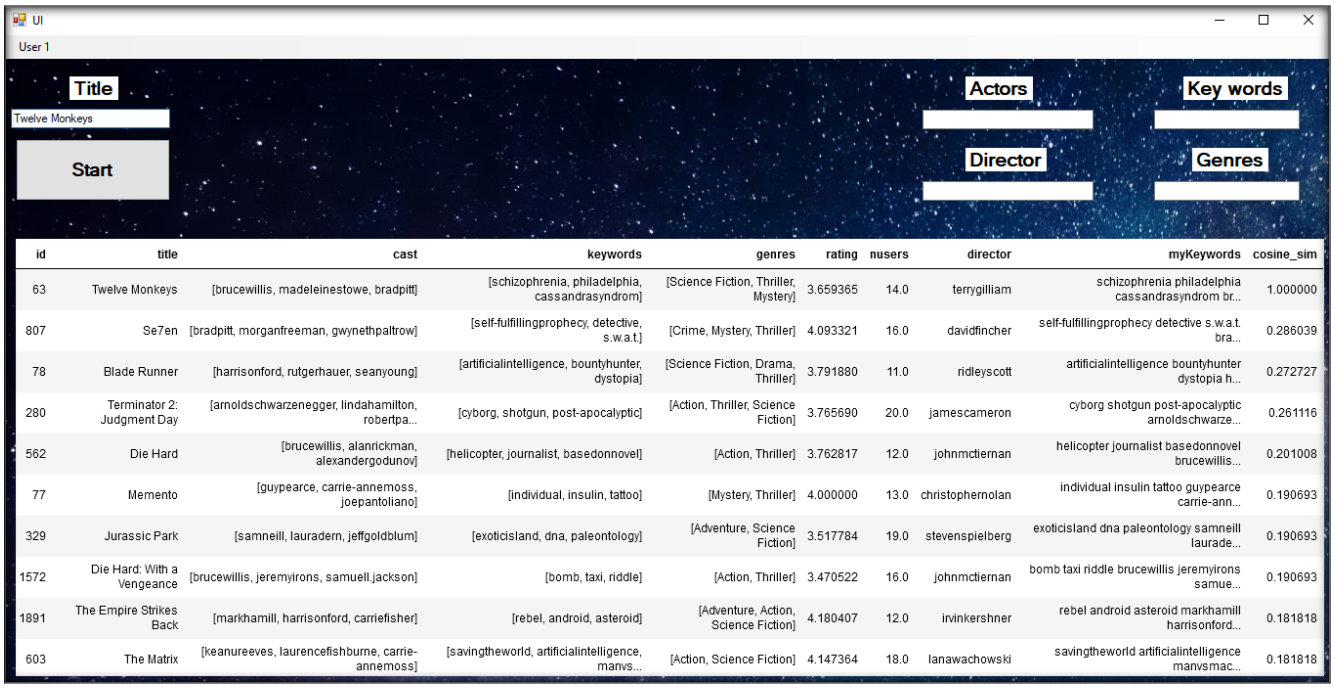


id	title	genres	rating	nusers
8874	My Best Friend's Wedding	[{'id': 35, 'name': 'Comedy'}, {'id': 10749, '...	4.651135	4.0
539	Psycho	[{'id': 18, 'name': 'Drama'}, {'id': 27, 'name...	4.600000	5.0
773	Little Miss Sunshine	[{'id': 35, 'name': 'Comedy'}, {'id': 18, 'nam...	4.596896	5.0
629	The Usual Suspects	[{'id': 18, 'name': 'Drama'}, {'id': 80, 'name...	4.506587	12.0
103	Taxi Driver	[{'id': 80, 'name': 'Crime'}, {'id': 18, 'name...	4.454160	9.0
14275	Hoop Dreams	[{'id': 99, 'name': 'Documentary'}]	4.389353	4.0
278	The Shawshank Redemption	[{'id': 18, 'name': 'Drama'}, {'id': 80, 'name...	4.376644	20.0
275	Fargo	[{'id': 80, 'name': 'Crime'}, {'id': 18, 'name...	4.372508	10.0
637	Life Is Beautiful	[{'id': 35, 'name': 'Comedy'}, {'id': 18, 'nam...	4.360561	4.0
424	Schindler's List	[{'id': 18, 'name': 'Drama'}, {'id': 36, 'name...	4.357477	17.0

Рисунок 3.29 - Рекомендовані фільми для користувача з ідентифікатором 5

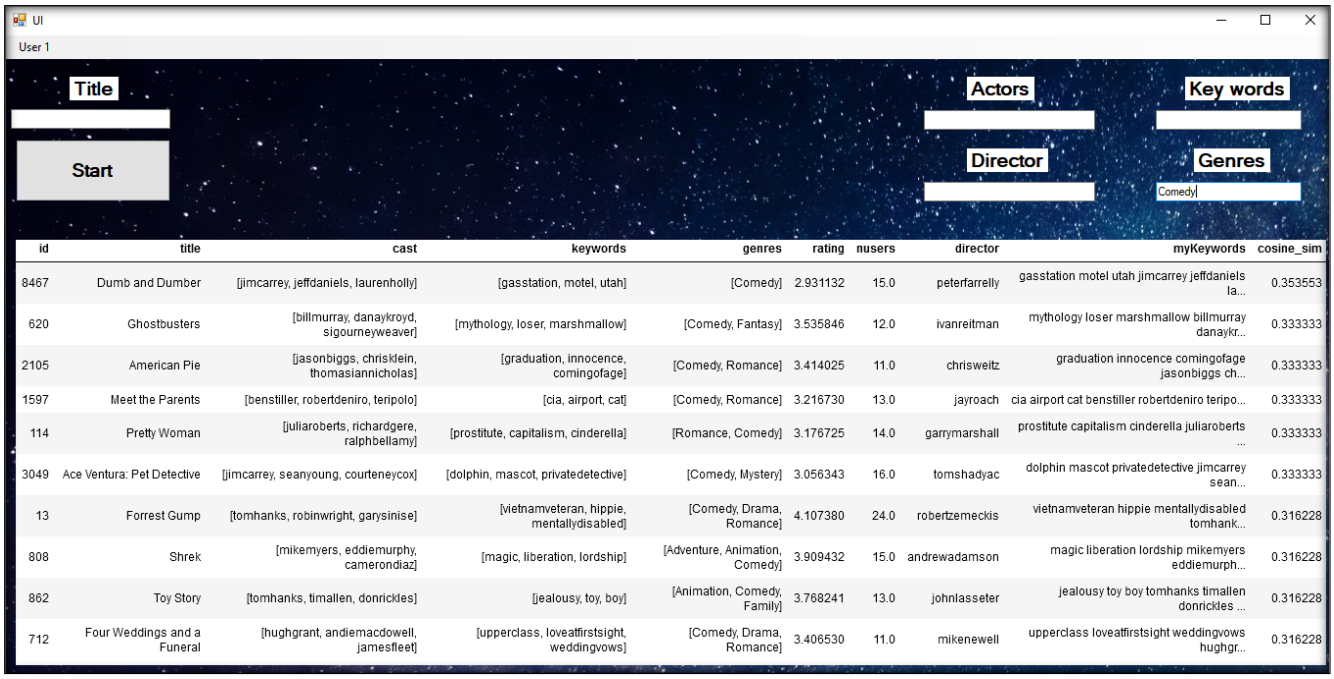
Як видно з рисунків 3.28 та 3.29 перший сценарій роботи інтелектуальної технології аналізу та рекомендації кінофільмів працює правильно.

Далі перевіримо другий та третій сценарій роботи інтелектуальної системи. На рисунках 3.30 та 3.31 зображно результати роботи другого та третього сценарію відповідно.



id	title	cast	keywords	genres	rating	nusers	director	myKeywords	cosine_sim
63	Twelve Monkeys	[brucewillis, madeleinestowe, bradpitt]	[schizophrenia, philadelphia, cassandrasyndrom]	[Science Fiction, Thriller, Mystery]	3.659365	14.0	terrygilliam	schizophrenia philadelphia cassandrasyndrom br...	1.000000
807	Se7en	[bradpitt, morganfreeman, gwynethpaltrow]	[self-fulfillingprophecy, detective, s.w.a.t.]	[Crime, Mystery, Thriller]	4.093321	16.0	davidfincher	self-fulfillingprophecy detective s.w.a.t. bra...	0.286039
78	Blade Runner	[harrisonford, rutgerhauer, seanyoung]	[artificialintelligence, bountyhunter, dystopia]	[Science Fiction, Drama, Thriller]	3.791880	11.0	ridleyscott	artificialintelligence bountyhunter dystopia h...	0.272727
280	Terminator 2: Judgment Day	[arnoldschwarzenegger, lindahamilton, robertpa...	[cyborg, shotgun, post-apocalyptic]	[Action, Thriller, Science Fiction]	3.765690	20.0	jamescameron	cyborg shotgun post-apocalyptic arnoldschwarze...	0.261116
562	Die Hard	[brucewillis, alanrickman, alexandergodunov]	[helicopter, journalist, basedonnovel]	[Action, Thriller]	3.762817	12.0	johnmctiernan	helicopter journalist basedonnovel brucewillis...	0.201008
77	Memento	[guypearce, carrie-annemoss, joepantoliano]	[individual, insulin, tattoo]	[Mystery, Thriller]	4.000000	13.0	christophernolan	individual insulin tattoo guypearce carrie-ann...	0.190693
329	Jurassic Park	[samneill, lauradern, jeffgoldblum]	[exoticisland, dna, paleontology]	[Adventure, Science Fiction]	3.517784	19.0	stevenspielberg	exoticisland dna paleontology samneill laurade...	0.190693
1572	Die Hard: With a Vengeance	[brucewillis, jeremyirons, samuell.jackson]	[bomb, taxi, riddle]	[Action, Thriller]	3.470522	16.0	johnmctiernan	bomb taxi riddle brucewillis jeremyirons samue...	0.190693
1891	The Empire Strikes Back	[markhamill, harrisonford, carriefisher]	[rebel, android, asteroid]	[Adventure, Action, Science Fiction]	4.180407	12.0	irvinkershner	rebel android asteroid markhamill harrisonford...	0.181818
603	The Matrix	[keanureeves, laurencefishburne, carrie-annemoss]	[savingtheworld, artificialintelligence, manvs...	[Action, Science Fiction]	4.147364	18.0	lanawachowski	savingtheworld artificialintelligence manvsmac...	0.181818

Рисунок 3.30 – Рекомендовані фільми за другим сценарієм роботи



id	title	cast	keywords	genres	rating	nusers	director	myKeywords	cosine_sim
8467	Dumb and Dumber	[jimcarrey, jeffdaniels, laurenholly]	[gasstation, motel, utah]	[Comedy]	2.931132	15.0	peterfarrelly	gasstation motel utah jimcarrey jeffdaniels la...	0.353553
620	Ghostbusters	[billmurray, danaykroyd, sigourneyweaver]	[mythology, loser, marshmallow]	[Comedy, Fantasy]	3.535846	12.0	ivanreitman	mythology loser marshmallow billmurray danaykr...	0.333333
2105	American Pie	[jasonbiggs, chrisklein, thomasiannicholas]	[graduation, innocence, comingofage]	[Comedy, Romance]	3.414025	11.0	chrisweitz	graduation innocence comingofage jasonbiggs ch...	0.333333
1597	Meet the Parents	[benstiller, robertdeniro, teripolo]	[cia, airport, cat]	[Comedy, Romance]	3.216730	13.0	jayroach	cia airport cat benstiller robertdeniro teripo...	0.333333
114	Pretty Woman	[juliaroberts, richardgere, ralphbellamy]	[prostitute, capitalism, cinderella]	[Romance, Comedy]	3.176725	14.0	garrymarshall	prostitute capitalism cinderella juliaroberts ...	0.333333
3049	Ace Ventura: Pet Detective	[jimcarrey, seanyoung, courteneycox]	[dolphin, mascot, privatedetective]	[Comedy, Mystery]	3.056343	16.0	tomshadyac	dolphin mascot privatedetective jimcarrey sean...	0.333333
13	Forrest Gump	[tomhanks, robinwright, garysinise]	[vietnamveteran, hippie, mentallydisabled]	[Comedy, Drama, Romance]	4.107380	24.0	robertzemeckis	vietnamveteran hippie mentallydisabled tomhank...	0.316228
808	Shrek	[mikemyers, eddiemurphy, camerondiaz]	[magic, liberation, lordship]	[Adventure, Animation, Comedy]	3.909432	15.0	andrewadamson	magic liberation lordship mikemyers eddiemurph...	0.316228
862	Toy Story	[tomhanks, timallen, donrickles]	[jealousy, toy, boy]	[Animation, Comedy, Family]	3.768241	13.0	johnlasseter	jealousy toy boy tomhanks timallen donrickles ...	0.316228
712	Four Weddings and a Funeral	[hughgrant, andiemacdowell, jamesfleet]	[upperclass, loveatfirstsight, weddingvows]	[Comedy, Drama, Romance]	3.406530	11.0	mikenewell	upperclass loveatfirstsight weddingvows hughgr...	0.316228

Рисунок 3.31 – Рекомендовані фільми за третім сценарієм роботи

На рисунку 3.30 зображено результат роботи розробленої інтелектуальної технології за другим сценарієм, коли користувач хоче отримати в рекомендацію фільми, що схожі на фільм «Twelve Monkeys». Як видно з рисунку найбільшу схожість має фільм «Twelve Monkeys», що логічно так як він і є цільовим фільмом.

Також видно, що усі рекомендовані фільми мають щось спільне з цільовим фільмов акторів, жанри, ключові слова або режисера.

На рисуку 3.31 зображено результат роботи розробленої інтелектуальної технології за третім сценарієм, коли користувач хоче отримати в рекомендацію фільми, що відповідають якомусь критерію в даному випадку, фільми що мають жанр «Comedy». Як видно з рисунку усі фільми відповідають цьому критерію і в жанрах містять цільовий жанр.

З результатів тестування можна зробити висновок, що система працює так як планувалось.

Визначимо точність прогнозованої оцінки. Для цього розділимо дані, а саме оцінки цільового користувача, які він ставив фільмам, на дві частини. На основі першої частини визначимо рекомендаці та спрогнозуємо оцінки на фільми другої частини та порівняємо отримані результати з початковими. Для порівняння визначимо середню абсолютну похибку (mean absolute error - MAE), та середньоквадратичну похибку (root-mean-square error - RMSE) розробленої технології та рекомендаційної системи веб-сайту «Fantlab».

На рисунку 3.32 зображено ствопчасту діаграму мінімальної, максимальної та середньої MAE розробленої технології та рекомендаційної системи веб-сайту «Fantlab».

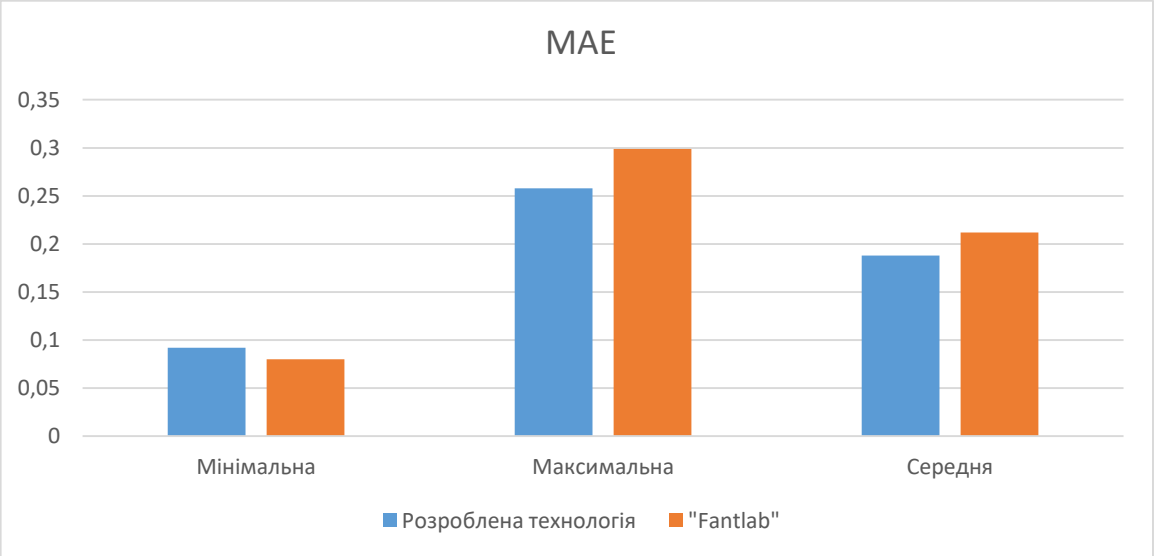


Рисунок 3.32 – Стовпчаста діаграма MAE

На рисунку 3.33 зображено ствопчасту діаграму мінімальної, максимальної та середньої RMSE розробленої технології та рекомендаційної системи веб-сайту «Fantlab».

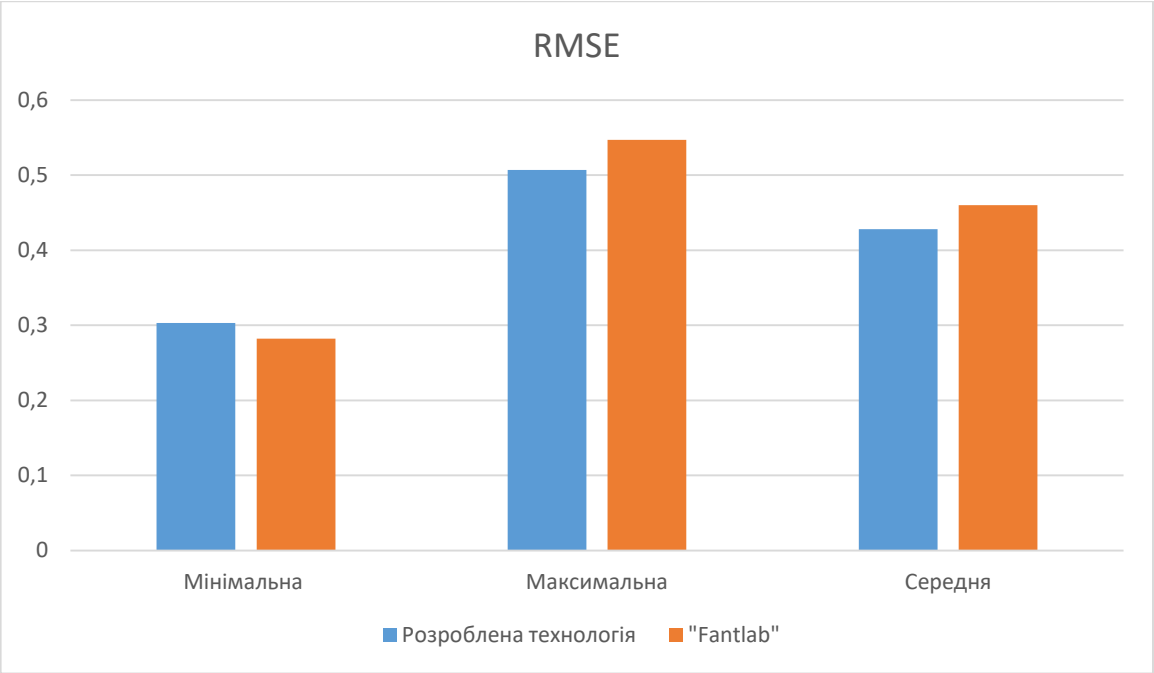


Рисунок 3.33 - Стовпчаста діаграма RMSE

В результаті тестування були отримані такі результати. Мінімальні похибки розробленої технології більші за мінімальні похибки рекомендаційної системи веб-сайту «Fantlab» приблизно на 15%. Однак, максимальні та середні похибки реалізованої системи менші за аналог приблизно на 14% та 12% відповідно.

Отже, з вище наведених даних можна зробити висновок, що розроблена інформаційна технологія в середньому працює краще за аналог приблизно на 12%, але рекомендаційна система веб-сайту «Fantlab» в окремих випадках може видавати кращий результат.

3.6 Висновки

В даному розділі було реалізовано інтелектуальну технологію аналізу та рекомендації кінофільмів для глядачів.

Для реалізації було обрану мову програмування Python, як таку, яка найкраще підходить для досягнення поставлених завдань.

Було реалізовано два основних модулі, модуль спільної фільтрації та модуль фільтрації на основі вмісту.

Було проведено тестування реалізованої системи. За результатами тестування було визначено, що розроблена технологія в середньому надає точніші рекомендації та прогнозує оцінку з похибками MAE = 0,188, RMSE = 0,428, що приблизно на 12% точніше ніж в аналогів.

4 ЕКОНОМІЧНА ЧАСТИНА

4.1 Оцінювання комерційного потенціалу розробки «інформаційна технологія аналізу та рекомендування кінофільмів для глядачів» (або технологічний аудит розробки)

Проведення оцінювання комерційного потенціалу розробки є метою технологічного аудиту. Для проведення технологічного аудиту залучені три незалежних експерти. Оцінювання комерційного потенціалу розробки здійснюється за 12-тю категоріями, наведені в таблиці 4.1

Таблиця 4.1 – Рекомендовані критерії оцінювання комерційного потенціалу розробки та їх можлива бальна оцінка

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Кри-терій	0	1	2	3	4
Технічна здійсненність концепції:					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність продукту в реальних умовах
Ринкові переваги (недоліки):					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів

Продовження таблиці 4.1

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Кр.	0	1	2	3	4
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практична здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витрачати значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Результати оцінювання комерційного потенціалу розробки потрібно звести в таблицю за зразком таблиці 4.2. Рівні комерційного потенціалу розробки – таблиці 4.3.

Таблиця 4.2 – Результати оцінювання комерційного потенціалу розробки

Критерії	Експерти		
	Експерт 1	Експерт 2	Експерт 3
	Бали, виставлені експертами:		
1	2	3	4
2	3	3	2
3	2	2	3
4	3	3	2
5	3	3	3
6	3	4	4
7	3	3	3
8	2	4	2
9	2	3	4
10	2	2	3
11	4	4	2
12	2	2	3
Сума балів	$СБ_1$=31	$СБ_2$=35	$СБ_3$=35
Середньоарифметична сума балів $\overline{СБ}$	$\overline{СБ} = \frac{\sum_{1}^{3} СБ_i}{3} = 33$		

Таблиця 4.3 – Рівні комерційного потенціалу розробки

Середньоарифметична сума балів $\overline{СБ}$, розрахована на основі висновків експертів	Рівень комерційного потенціалу розробки
0 – 10	Низький
11 – 20	Нижче середнього
21 – 30	Середній
31 – 40	Вище середнього
41 – 48	Високий

В результаті було визначено, що рівень комерційного потенціалу розробки є вище середнього.

4.2 Прогнозування витрат на виконання та впровадження результатів наукової роботи

1. Основна заробітна плата розробників, які працюють над проектом визначається за формулою:

$$Зо = \frac{М}{Тр} * t \text{ [грн]}, \quad (4.1)$$

де М – місячний посадовий оклад розробника;

Тр – число робочих днів в місяці (T_p= 22 дні);

t – число днів роботи розробника.

Над створенням розробки працювали науковий співробітник та інженер-програміст, отже, виконаємо для них всі необхідні розрахунки:

$$Зо = \frac{18000,00}{22} * 10 = 8181,9 \text{ (грн)},$$

$$Зо = \frac{16000,00}{22} * 25 = 18181,8 \text{ (грн)}.$$

Результати обрахунків заробітної плати наведено в таблиці 4.4.

Таблиця 4.4 – Заробітна плата

Найменування посади	Місячний посадовий	Оплата за робочий	Число днів	Витрати на заробітну плату,
1.Науковий співробітник	18000,00	818,18	10	8181,9
2.Інженер-програміст	16000,00	727,27	25	18181,8
Всього				$\sum Зо = 26363,7$

2. Додаткова заробітна плата Зд всіх розробників та робітників, які брали участь у виконанні даного етапу роботи, розраховується як (10…12) % від суми основної заробітної плати всіх розробників та робітників, тобто за формулою:

$$З_д = (10…12\%) * З_о \text{ [грн]}, \quad (4.2)$$

де Зо – основана заробітна плата.

$$Зд = \frac{10 * 26363,7}{100} = 2636,37 \text{ (грн)}.$$

3. Нарахування на заробітну плату Нзп розробників та робітників, які брали участь у виконанні даного етапу роботи, розраховуються за формулою:

$$Нзп = 22\% \cdot (З_о + З_д)\ [грн], \quad (4.3)$$

$$Нзп = \frac{22 * (26363{,}7 + 2636{,}37)}{100} = 6380{,}01\ (грн).$$

4. Амортизація обладнання, комп'ютерів та приміщень А, які використовувались під час (чи для) виконання даного етапу роботи.

Дані відрахування розраховують по кожному виду обладнання, приміщенням тощо. Величина амортизаційних відрахувань наведена в таблиці 4.5.

Таблиця 4.5 – Величина амортизаційних відрахувань

Найменування обладнання	Балансова вартість, грн	Норма амортизації, %	Термін використання обладнання, міс.	Величина амортиза-ційних відрахувань, грн
1. Програмне забезпечення PyCharm	12700	25	2	529,1
3.Ноутбук	4300	15	2	107,5
4.Приміщення	12700	25	1	264
Всього				900,6

5. Витрати на послуги під час виконання даного етапу роботи.

Витрати на послуги наведено в талиці 4.6.

Таблиця 4.6 – Послуги, що використовуються при виготовленні інноваційного продукту

Найменування комплектуючих (робіт, послуг)	Кількість місяців	Ціна за одиницю, грн.	Сума, грн.
1. Послуги мережі Інтернет	1	100	100
Всього	К = 100 грн.		

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

$$В_е = В*П*Ф*К_п \text{ [грн]}, \quad (4.4)$$

де В – вартість 1 кВт електроенергії, грн;

П – установлена потужність обладнання, кВт/год;

Ф – фактична кількість годин роботи обладнання, яке задіяне на виготовлення одного виробу, годин;

$К_п$ – коефіцієнт використання потужності, $К_п \leq 1$.

Вартість 1кВт електроенергії рівна 2,50 (грн), потужність обладнання рівна 90 Вт, що тотожно 0,09 кВт, фактична робота обладнання для роботи рівна 289 год, а коефіцієнт використання потужності приймемо за 0,8.

$$В_е = 2{,}50 * 0{,}09 * 289 * 0{,}8 = 52{,}02 \text{ (грн)}.$$

7. Інші витрати. Інші витрати охоплюють: загально виробничі витрати (витрати управління організацією, ремонт та експлуатація основних засобів, витрати на опалення, освітлення тощо), адміністративні витрати (проведення зборів, оплата юридичних та аудиторських послуг, тощо), витрати на збут (витрати на рекламу, перепідготовка кадрів) на інші операційні витрати (штрафи, пені, матеріальні допомоги, втрати від знецінення запасів тощо).

Інші витрати можна розрахувати за нормативами відносно основної заробітної плати основних робітників, які виготовляють продукцію, за формулою:

$$В_{ін} = Н \cdot З_о \text{ [грн]}, \quad (4.5)$$

де Н – норматив загальновиробничих витрат 100-300%.

$$В_{ін} = \frac{100*26363{,}7}{100} = 26363{,}7 \text{ (грн)}.$$

8. Сума всіх попередніх статей витрат дає витрати на виконання даної частини (розділу, етапу) роботи – В.

$$В = 26363{,}7 + 2636{,}37 + 6380{,}01 + 900{,}6 + 100 + 52{,}02 + 26363{,}7$$
$$= 70872{,}79 \text{ (грн)}.$$

9. Розрахунок загальних витрат на розробку. Загальна вартість всієї НДДКТ $В_{заг}$ визначається за формулою:

$$В_{заг} = \frac{В_{ін}}{\alpha} \text{ [грн]}, \quad (4.6)$$

де α – частка витрат, які безпосередньо здійснює виконавець даної НДДКР, у відносних одиницях. Приймемо $\alpha = 0{,}65$.

$$В_{заг} = \frac{26363{,}7}{0{,}65} = 40558{,}47 \text{ (грн)}.$$

10. Прогнозування загальних витрат на виконання та впровадження дослідження. Прогнозування загальних витрат ЗВ на виконання та впровадження результатів виконаної НДДКР здійснюється за формулою:

$$ЗВ = \frac{В_{заг}}{\beta} \text{ [грн]}, \quad (4.7)$$

де β – коефіцієнт, який характеризує етап виконання даної НДДКР. Приймаємо $\beta = 0{,}6$.

$$ЗВ = \frac{40558{,}47}{0{,}6} = 71813 \text{ (грн)}.$$

4.3 Прогнозування комерційних ефектів від реалізації результатів розробки

У даному підрозділі виконано прогнозування, яку вигоду можна отримати у майбутньому від впровадження результатів даної наукової роботи.

Передбачається, що виконання наукової роботи та впровадження результатів по розробці займе 1 рік.

Основні позитивні результати від впровадження розробки очікуються протягом 3 років після її впровадження.

Саме зростання чистого прибутку забезпечить підприємству (організації) надходження додаткових коштів, які дозволять покращити фінансові результати діяльності. Збільшення чистого прибутку підприємства обчислюється за формулою:

$$\Delta П_i = \sum_{1}^{n} (\Delta П_я \cdot N + П_я \Delta N)_i \ [грн], \quad (4.8)$$

де

$\Delta П_i$ – покращення основного кількісного показника діяльності підприємства від впровадження результатів наукової розробки;

$\Delta П_я$ – покращення основного якісного показника діяльності підприємства у даному році після впровадження результатів наукової розробки;

N – основний кількісний показник, який визначає діяльність підприємства у даному році до впровадження результатів наукової розробки;

$П_я$ – основний якісний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки;

n – кількість років, протягом яких очікується отримання позитивних результатів від впровадження розробки.

В результаті впровадження результатів наукової розробки покращується якість розробки, що дозволяє підвищити ціну його реалізації на 120 грн, а кількість

потенційних користувачів ресурсу збільшиться: протягом першого року – на

Орієнтовно: реалізація продукції до впровадження результатів наукової розробки складала 12000 шт., а прибуток, що його отримувало підприємство на одиницю продукції до впровадження результатів наукової розробки - 350 грн.

Спрогнозуємо збільшення чистого прибутку підприємства від впровадження результатів наукової розробки у кожному році відносно базового.

Збільшення чистого прибутку підприємства протягом наступних трьох років складе:

Збільшення чистого прибутку підприємства $\Delta П_i$ протягом першого року складе:

$$\Delta П_1 = 120 \cdot 12000 + (350 + 120) \cdot 1200 = 1728564 \text{ (грн)}.$$

Збільшення чистого прибутку підприємства $\Delta П_i$ протягом другого року (відносно базового року, тобто року до впровадження результатів наукової розробки) складе:

$$\Delta П_2 = 120 \cdot 12000 + (350 + 120) \cdot 600 = 864282 \text{ (грн)}.$$

Збільшення чистого прибутку підприємства $\Delta П_i$протягом третього року (відносно базового року, тобто року до впровадження результатів наукової розробки) складе:

$$\Delta П_3 = 120 \cdot 12000 + (350 + 120) \cdot 300 = 432141 \text{ (грн)}.$$

Загальне збільшення прибутку підприємства, або комерційний ефект, від впровадження розробки за 3 роки становить:

$$\Delta П = 1728564 + 864282 + 432141 = 3024987 \text{ (грн)}.$$

4.4 Розрахунок ефективності вкладених інвестицій та періоду їх окупності

Розрахований комерційний ефект від можливого впровадження розробки ще не означає, що ця розробка реально буде впроваджена. Якщо збільшення прогнозованого прибутку від впровадження результатів наукової розробки є вигідним для підприємства, то це ще не означає, що інвестор погодиться фінансувати розробку.

Основними показниками, які визначають доцільність фінансування наукової розробки певним інвестором, є абсолютна і відносна ефективність вкладених інвестицій та термін їх окупності.

Розрахунок ефективності вкладених інвестицій передбачає проведення таких робіт:

1-й крок. Розрахуємо теперішню вартість інвестицій PV, що вкладаються в наукову розробку. Такою вартістю, можна вважати прогнозовану величину загальних витрат ЗВ на виконання та впровадження результатів НДДКР, розраховану нами раніше за формулою (4.7), тобто будемо вважати, що ЗВ = PV=71813.

2-й крок. Розрахуємо очікуване збільшення прибутку $\Delta П_i$, що його отримає підприємство (організація) від впровадження результатів наукової розробки, для кожного із років, починаючи з першого року впровадження. Таке збільшення прибутку, також було розраховане раніше за формулою (4.8). Сумарне значення $\Delta П_i = 3024987$ (грн).

3-й крок. Для спрощення подальших розрахунків побудуємо вісь часу, на яку нанесемо всі платежі (інвестиції та прибутки), що мають місце під час виконання науково-дослідної роботи та впровадження її результатів.

Платежі показуються у ті терміни, коли вони здійснюються. Рисунок, що характеризує рух платежів (інвестицій та додаткових прибутків) буде мати вигляд, наведений на рисунку 4.1.

Рисунок 4.1 - Вісь часу з фіксацією платежів, що мають місце під час розробки та впровадження результатів НДДКР

4-й крок. Розрахуємо абсолютну ефективність вкладених інвестицій $E_{абс}$.

Для цього користуються формулою:

$$E_{абс} = (ПП - PV), \quad (4.9)$$

де ПП – приведена вартість всіх чистих прибутків, що їх отримає підприємство (організація) від реалізації результатів наукової розробки, грн;

PV – теперішня вартість інвестицій PV = ЗВ = 71813 (грн).

У свою чергу, приведена вартість всіх чистих прибутків ПП розраховується за формулою:

$$ПП = \sum_{1}^{T} \frac{\Delta П_i}{(1+\tau)^t} [тис. грн], \quad (4.10)$$

де $\Delta П_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої НДДКР, грн;

Т – період часу, протягом якого виявляються результати впровадженої НДДКР, роки;

τ – ставка дисконтування, яку можна взяти за щорічний прогнозований рівень інфляції в країні; для України цей показник знаходиться на рівні 0,1;

t – період часу (в роках) від моменту отримання чистого прибутку до точки „0”.

Отримаємо:

$$ПП = \frac{1728}{(1+0,1)^2} + \frac{864}{(1+0,1)^3} + \frac{432}{(1+0,1)^4} = \frac{1728}{1,21} + \frac{864}{1,33} + \frac{432}{1,46} =$$
$$= 1428,09 + 649,62 + 739,726 = 2817,44 \text{ (тис. грн)}.$$

Тоді $Е_{абс}$ = (2817,44-71,813) = 2745,627 (тис. грн).

Оскільки $Е_{абс} > 0$, то результат від проведення наукових досліджень та їх впровадження може принести прибуток, але це також ще не свідчить про те, що інвестор буде зацікавлений у фінансуванні даного проекту (роботи).

5-й крок. Розраховують відносну (щорічну) ефективність вкладених в наукову розробку інвестицій $Е_в$. Для цього користуються формулою:

$$Е_в = \sqrt[Т_ж]{1 + \frac{Е_{абс}}{PV}} - 1 \, [\%], \quad (4.11)$$

де $Е_{абс}$ – абсолютна ефективність вкладених інвестицій, грн;

PV – теперішня вартість інвестицій PV = ЗВ, грн;

$Т_ж$ – життєвий цикл наукової розробки, роки.

$$Е_в = \sqrt[4]{1+\frac{2745,627}{71,813}} - 1 = \sqrt[4]{1+38,23} - 1 = 2.25 - 1 = 1.25 = 125\ (\%).$$

Далі, розрахована величина $Е_в$ порівнюється з мінімальною (бар'єрною) ставкою дисконтування $\tau_{мін}$, яка визначає ту мінімальну дохідність, нижче за яку інвестиції вкладатися не будуть. У загальному вигляді мінімальна (бар'єрна) ставка дисконтування $\tau_{мін}$визначається за формулою:

$$t = d + f\ [\%], \qquad (4.12)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2018 році в Україні d = (0,16...0,2);

f – показник, що характеризує ризикованість вкладень; зазвичай, величина f = (0,05...0,1), але може бути і значно більше.

$$t = 0,2 + 0,1 = 0,3 = 30\ (\%).$$

Величина $Е_в > \tau_{мін}$, інвестор може бути зацікавлений у фінансуванні даної наукової розробки.

Розрахуємо термін окупності вкладених коштів у реалізацію наукового проекту за формулою:

$$Т_{ок} = \frac{1}{Е_в}\ [років], \qquad (4.13)$$

$$Т_{ок} = \frac{1}{1,25} = 0,80\ (року).$$

$Ток$ = 0,80 (років), то фінансування розробки є доцільним.

4.5 Висновки

У четвертому розділі магістерської кваліфікаційної роботи проведено розрахунки, що доводять економічну доцільність та ефективність впровадження розробленого продукту. Розрахунки поділено на 4 частини, які утворили відповідні підрозділи цього розділу. У комплексі ці підрозділи дозволяють побачити цілісну картину економічної доцільності нового інтелектуального рішення.

У першому підрозділі здійснено оцінювання комерційного потенціалу розробки. Виставлені бали, говорять про рівень комерційного потенціалу, який є середнім.

Другий підрозділ економічної частини демонструє витрати на розробку, що розраховуються, як сума усіх статей витрат поділена на ступінь готовності продукту. Розрахований кошторис витрат на розробку складав 71813 грн.

Далі прогнозуються комерційні ефекти від реалізації розробки, тобто яку вигоду, можна отримати у майбутньому від впровадження результатів виконаної наукової роботи. Загальне збільшення прогнозованого прибутку за 3 роки становить 3024987 грн.

Останній підрозділ висвітлює основі показники, які визначають доцільність фінансування наукової розробки певним інвестором.

Обрахована абсолютна ефективність становить 2745,627 тис. грн, що свідчить про те, що інвестор буде зацікавлений у фінансуванні даної розробки.

Відносна (щорічна) ефективність становить 125 %, що більше мінімальної ставки дисконтування, що ще раз підтверджує зацікавленість інвестора.

Термін окупності вкладених коштів у реалізацію наукового проекту становить 0,8, що означає, що вкладені кошти повернуться через дев’ять з половиною місяців.

Таким чином, можна стверджувати, що фінансування розробки є доцільним.

ВИСНОВКИ

В процесі виконання магістерської кваліфікаційної роботи було розроблено та програмно реалізовано інтелектуальну технологю аналізу та рекомендування кінофільмів для глядачів.

В першому розділі було сформовано поняття рекомендаційних систем. Розглянуто три основні методи побудови рекомендаціних систем. Також було розглянуто та проаналізовано існуючі системи рекомендацій. На основі проаналізованих даних було обрано гібридну стратегію розробки інформаційної технології, як таку яка найкраще підходить для досягнення поставлених цілей.

У другому розділі було розроблено загальну структуру інформаційної технології. Вона складається з двох модулів, які працюють послідовно. Було розроблено і самі модулі, модуль спільної фільтрації та модуль фільтрації на основі вмісту.

У третьому розділі для програмної реалізації було обрану мову програмування Python та використовуючи її реалізовано два основних модулі, модуль спільної фільтрації та модуль фільтрації на основі вмісту.

За результатами тестування було визначено, що розроблена технологія в середньому надає точніші рекомендації та прогнозує оцінку з похибками MAE = 0,188, RMSE = 0,428, що приблизно на 12% точніше ніж в аналогів. Даний результат було отримано завдяки тому, що розроблена технологія на відміну від існуючих, використовує удосконалений метод гібридної фільтрації.

У ході виконання економічної частини кваліфікаційної роботи, на основі розрахунків, було доведено, що продукт є економічно доцільним, оскільки витрати на розробку вказаного засобу з використанням відповідної технології становлять 71813 грн. Показник ефективності вкладених інвестицій $Е_{абс}$ = 2745,627 тис. грн, відносної – 125%, а термін окупності інвестицій становить 0,8 роки.

За результатами магістерської кваліфікаційної роботи опубліковано тези доповідей на науковій конференції.

Отже, поставлені задачі магістерської кваліфікаційної роботи були виконані в повному обсязі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Корпало А. В. Обґрунтування розробки інформаційної системи аналізу та рекомендацій кінофільмів [Електронний ресурс] / А. Корпало // Матеріали XV міжнародної конференції "Контроль і управління в складних системах (КУСС-2020)", м. Вінниця, 8-10 жовтня 2020 р.– Електрон. текст. дані. – Вінниця : ВНТУ, 2020. – Режим доступу: http://ir.lib.vntu.edu.ua//handle/123456789/30583.

2. Recommender Systems: An Introduction. / [Jannach D., Zanker M., Felfernig A., Friedrich G.]. — Cambridge: Cambridge University Press, 2010. — 215 c.

3. P. Lops Recommender Systems Handbook / P. Lops, M. Gemmis, G. Semeraro; — Boston: Springer-Verlag, 2011. — 840 p.

4. Introduction to recommender systems [Електронний ресурс]. Режим доступу: https://towardsdatascience.com/introduction-to-recommender-systems-6c66cf15ada

5. K. Falk Practical Recommender Systems 1st Edition / K. Falk. - New York: ManningPublications, 2019. – 432 p.

6. Recommender Systems in Practice [Електронний ресурс]. Режим доступу: https://towardsdatascience.com/recommender-systems-in-practice-cef9033bb23a

7. Xiaoyuan Su A Survey of Collaborative Filtering Techniques. Advances in Artificial Intelligence / Xiaoyuan Su, Khoshgoftaar T. M. - Florida: Hindawi, 2009. - 292 p.

8. U. K. Viswanathan. Optimization of User Based Collaborative Filtering./ U. K. Viswanathan. - New York: LAP LAMBERT Academic Publishing, 2019. – 325 p.

9. S., Berkovsky Optimization of User Based Collaborative Filtering./ S., Berkovsky, I., Cantador, Domonkos T. - Boston: World Scientific Publishing Co , 2019. – 635 p.

10. B. Jesús A collaborative filtering approach to mitigate the new user cold start problem / B., Jesús; O., Fernando; A.; Bernal. - Seattle: ArtPublications, 2012. – 432 p.

11. Salter, J. CinemaScreen Recommender Agent: Combinin Collaborative and Content-Based Filtering / Salter, J.; Antonopoulos, N.IEEE Intelligent Systems, 2006 – 512 p.

12. Knowledge-based recommender systems [Електронний ресурс] Режим доступу: https://en.wikipedia.org/wiki/Knowledge-based_recommender_system.

13. Р. Берк Системы рекомендаций, основанные на знаниях / Р. Берк, Берлин: Стар Паблишынг 2000 – 269 с.

14. Y. Salem History-Guided Conversational Recommendation / Y. Salem, J. Hong, W. Liu., Texas: Gold and Co. 2014. – 623 p.

15. Л. Чен Рекомендации наоснове критики: обзор и новые тенденции. «Моделирование пользователей и адаптированное к пользователю взаимодействие» / Л. Чен, П. Пу Лондон: Барли Паблишинг 2012. – 213 с.

16. T. Chen Content Recommendation System Based on Private Dynamic User Profile / T. Chen, W. LiHan, H. D. Wang,Y., X. Zhou, Y. Zang. California, 2014 – 139 p.

17. Amazon [Електронний ресурс]. – Режим доступу до ресурсу: https://www.aboutamazon.com/?utm_source=gateway&utm_medium=footer

18. Ebay [Електронний ресурс]. – Режим доступу до ресурсу: https://www.ebayinc.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. Deezer [Електронний ресурс]. – Режим доступу до ресурсу: https://www.deezer.com/ru/company

22. LinkedIn [Електронний ресурс]. – Режим доступу до https://about.linkedin.com/?trk=homepage-basic_directory_aboutUrl

23. YouTube [Електронний ресурс]. – Режим доступу до ресурсу: https://www.youtube.com/howyoutubeworks

24. The Python Tutorial [Електронний ресурс]. – Режим доступу: https://docs.python.org/3/tutorial/index.html

25. Методичні вказівки до виконання та оформлення магістерських кваліфікаційних робіт для студентів спеціальності 126 –«Інформаційні системи та технології» денної форми навчання / Уклад. В. Б. Мокін, С. О. Жуков, А. Р. Ящолт, О.М.Козачко, Л.М.Скорина. – Електронне видання. – Вінниця : ВНТУ, 2018. – 48 с.

Додаток А

Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет комп'ютерних систем і автоматики

ЗАТВЕРДЖУЮ
Завідувач кафедри САІТ
_____ д.т.н., проф. В.Б. Мокін
(підпис)
"__" _____ 2020

ТЕХНІЧНЕ ЗАВДАННЯ

на магістерську кваліфікаційну роботу

«ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ АНАЛІЗУ ТА РЕКОМЕНДУВАННЯ КІНОФІЛЬМІВ ДЛЯ ГЛЯДАЧІВ»

08-53.МКР.001.03.000.ПЛ

Керівник магістерської кваліфікаційної роботи
доцент, к.т.н.
_____ О. М. Козачко
(підпис)
"__" _____ 2020 р.

Розробив студент гр. 2ІСТ-19м
_____ А. В. Корпало
(підпис)
"__" _____ 2020 р.

Вінниця 2020

1. Підстава для проведення робіт

Підставою для виконання роботи є наказ № __ по ВНТУ від «__» _____ 2020 р., та індивідуальне завдання на МКР, затверджене протоколом № __ засідання кафедри САІТ від «__» _____ 2020 р.

2. Джерела розробки:

- K. Falk Practical Recommender Systems 1st Edition / K. Falk. - New York: Manning Publications, 2019. – 432 p.
- Python для начинающих [Електронний ресурс]. – Режим доступу до ресурсу: https://ru.code-basics.com/languages/python

3. Мета і призначення роботи

Розробка інформаційної системи управління проєктами.

4. Вихідні дані для проведення робіт:
- мова програмування Python;
- датасет «The Movies Dataset»;

5. Методи дослідження:
- методи фільтрації даних;
- моделювання системи.

6. Етапи роботи і терміни їх виконання

a) Аналіз предметної області ..... __. – __
b) Розробка інформаційної технології ..... __ – __
c) Реалізація інформаційної технології..... __ – __
d) Розробка інструкції користувача..... __ – __

7. Очікувані результати та порядок реалізації

Отримання програмного забезпечення інформаційної аналітичної технології аналізу та рекомендування кінофільмів для глядачів.

8. Вимоги до розробленої документації

Пояснювальна записка оформлена у відповідності до вимог «Методичних вказівок до виконання та оформлення магістерських кваліфікаційних робіт для студентів спеціальності 126 – «Інформаційні системи та технології» денної форми навчання».

9. Порядок приймання роботи

Публічний захист…….__.__.2020 р.
Початок розробки «___» _____ 2020 р.
Граничні терміни виконання МКР «___» _____2020 р.

Розробив студент групи 2ІСТ-19м _____ Корпало А. В..

Додаток Б

Інструкція користувача

Для початку роботи з програмою користувачу потрібно запустити її та авторизуватись. На рисунку Б.1 зображено вікно авторизації.

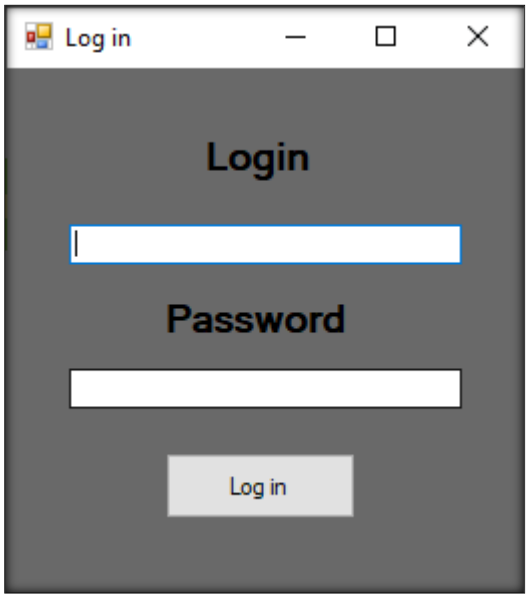


Рисунок Б.1 – Вікно авторизації

Після входу в систему користувач побачить головне вікно програми, яке зображено на рисунку Б.2. На ньому присутні 5 текстових полів і одна кнопка.

Щоб отримати рекомендацію лише на основі власних вподобань, користувачу потрібно не заповнюючи текстові поля, натиснути на кнопку з написом «Start».

Щоб отримати рекомендацію на основі фільму, користувачу потрібно у текстовому полі «Title» ввести назву фільму та натиснути кнопку - «Start».

Щоб отримати рекомендацію на основі критерії користувачу потрібно ввести відповідні дані в одне або кілька полів з назвами «Actors», «Key words», «Director», «Genres» та натиснути кнопку - «Start».

Щоб закінчити роботу з програмую користувачу потрібно натиснути на кнопку «X», яка знаходиться з правої сторони вікна програми.

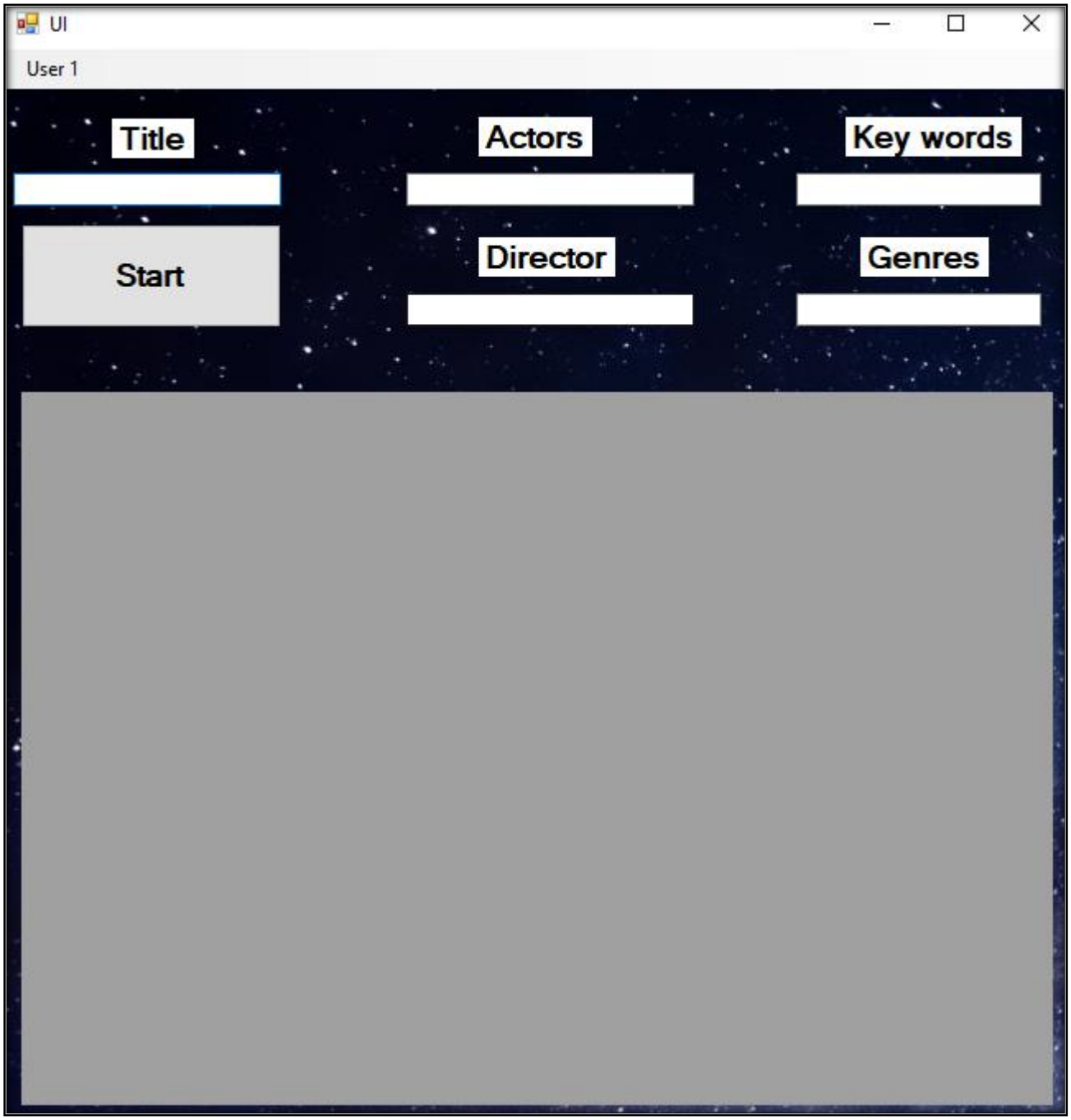


Рисунок Б.2 – Головне вікно програми

Додаток В

Лістинг програми

```
import numpy as np
import pandas as pd
import csv
from ast import literal_eval
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.metrics.pairwise import cosine_similarity
import gc
import itertools

ratings = pd.read_csv('../input/the-movies-dataset/ratings_small.csv')
ratings = ratings.drop(ratings[ratings.userId > 10000].index)

def GetUserSimilarity(df, targetUserDic):
    userRatingSum = 0.0
    numberMovies = 0
    for row in df.itertuples(index=False, name='Pandas'):
        if targetUserDic.get(row.movieId) is not None:
            userRatingSum += abs(float(row.rating) - float(targetUserDic[row.movieId]))
            numberMovies += 1

    if (userRatingSum != 0.0 and numberMovies != 0):
        df['similarity'] = 2 - np.sqrt(userRatingSum/numberMovies)
    else:
        df['similarity'] = 0

    return df

def GetMovieRatingForecast(movieGroup):
    predictRating = 0.0
    nMovie = 0
    sumSimilarity = movieGroup.similarity.sum()
    for row in movieGroup.itertuples(index=False, name='Pandas'):
        if sumSimilarity != 0:
            predictRating += ((row.similarity * 100) / (sumSimilarity * 100)) * float(row.rating)
            nMovie += 1

    cols = ['rating', 'nusers']
```

```
    return pd.Series((predictRating, nMovie), index=cols)

def combineKeywords(x):
    return ' '.join(x['keywords']) + ' ' + ' '.join(x['cast']) + ' ' + x['director'] + ' ' + ' '.join(x['genres'])

def getDirector(x):
    for i in x:
        if i['job'] == 'Director':
            return i['name']
    return np.nan

def getFirstThree(x):
    if isinstance(x, list):
        names = [i['name'] for i in x]

        if len(names) > 3:
            names = names[:3]
        return names

    return []

def deletingSpaces(x):
    if isinstance(x, list):
        return [str.lower(i.replace(" ", "")) for i in x]
    else:

        if isinstance(x, str):
            return str.lower(x.replace(" ", ""))
        else:
return ''
def GetMovieRecommendation(userId, ratings, title = '', genres = [''], director = '', cast= [''], keywords = ['']
):
    targetUserData = ratings[ratings.userId == userId]
    targetUserDic = dict(zip(targetUserData.movieId, targetUserData.rating))

    UsersSimilarity = ratings.groupby(['userId'], group_keys=False,
as_index=False).apply(GetUserSimilarity, targetUserDic)
    UsersSimilarity = UsersSimilarity.drop(UsersSimilarity[UsersSimilarity.similarity < 1.2].index)
    MovieRatingForecast = UsersSimilarity[UsersSimilarity.similarity > 1].groupby(['movieId'],
as_index=False).apply(GetMovieRatingForecast)
```

```
    #return
MovieRatingForecast[~MovieRatingForecast.movieId.isin([*targetUserDic])][MovieRatingForecast.nuser
s > 5].sort_values(by=['rating', 'nusers'],ascending=False)
    #return MovieRatingForecast[~MovieRatingForecast.movieId.isin([*targetUserDic])]

    if title != '' or genres != [''] or director != '' or cast != [''] or keywords != ['']:
        credits = pd.read_csv('../input/the-movies-dataset/credits.csv')
        moviesMetaData = pd.read_csv('../input/the-movies-dataset/movies_metadata.csv',
low_memory=False)
        keywords = pd.read_csv('../input/the-movies-dataset/keywords.csv')
        links = pd.read_csv('../input/the-movies-dataset/links.csv')

        moviesMetaData['id'] = moviesMetaData['id'].astype(str)
        credits['id'] = credits['id'].astype(str)
        keywords['id'] = keywords['id'].astype(str)
        links['tmdbId'] = links['tmdbId'].fillna(0.0).astype(int)
        links['tmdbId'] = links['tmdbId'].astype(str)
        MovieRatingForecast['movieId'] = MovieRatingForecast['movieId'].astype(str)
        links['movieId'] = links['movieId'].astype(str)

        moviesMetaData.rename(columns={'id': 'MMD_id'}, inplace = True)
        credits.rename(columns={'id': 'C_id'}, inplace = True)
        keywords.rename(columns={'id': 'K_id'}, inplace = True)
        links.rename(columns={'movieId': 'L_id'}, inplace = True)
        MovieRatingForecast.rename(columns={'movieId': 'movieId_r'}, inplace = True)

        moviesMetaData = moviesMetaData.merge(links, left_on='MMD_id', right_on='tmdbId')
        moviesMetaData = moviesMetaData.merge(MovieRatingForecast, left_on='L_id',
right_on='movieId_r')
        mainList= moviesMetaData.merge(credits, left_on='MMD_id', right_on='C_id')
        mainList= mainList.merge(keywords, left_on='MMD_id', right_on='K_id')
        mainList = mainList.drop_duplicates()

        mainList.rename(columns={'MMD_id': 'id'}, inplace = True)

        mainList = mainList[['id', 'title', 'cast', 'crew', 'keywords', 'genres', 'rating', 'nusers']]

        features = ['cast', 'crew', 'keywords', 'genres']
        for feature in features:
            mainList[feature] = mainList[feature].apply(literal_eval)
```

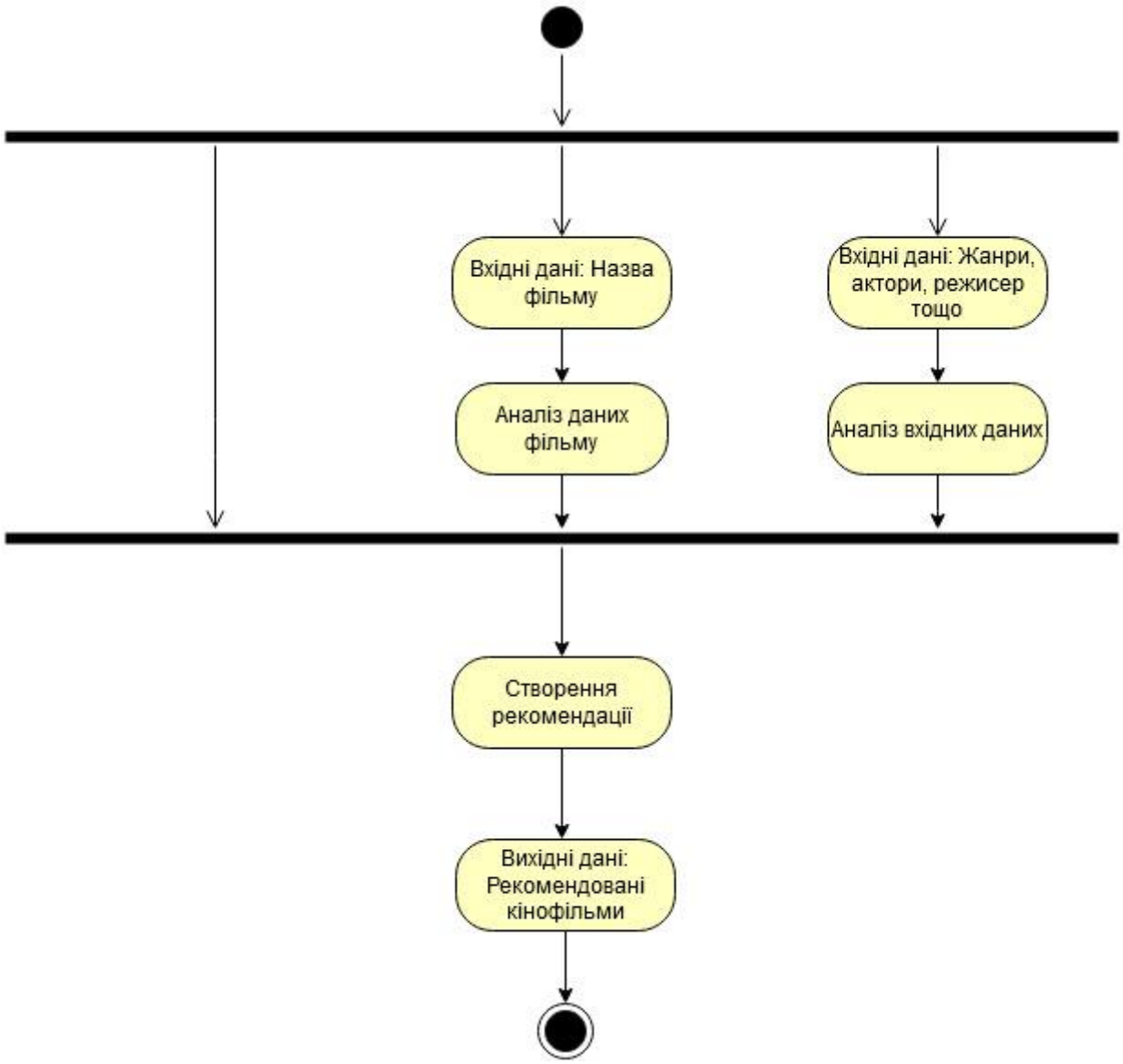
```
    mainList['director'] = mainList['crew'].apply(getDirector)
    del mainList['crew']

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

    if title == '' and (genres != [''] or director != '' or cast != [''] or keywords != ['']):
        new_row = {'id':'0', 'title':'Current', 'cast':cast, 'keywords':keywords, 'genres':genres, 'rating':0.0,
'nusers':0, 'director':director}
        mainList = mainList.append(new_row, ignore_index=True)

    features = ['cast', 'keywords', 'director']

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

    mainList['myKeywords'] = mainList.apply(combineKeywords, axis=1)

    count = CountVectorizer(stop_words='english')
    count_matrix = count.fit_transform(mainList['myKeywords'])
    cosine_sim = cosine_similarity(count_matrix, count_matrix)

    mainList = mainList.reset_index()
    indices = pd.Series(mainList.index, index=mainList['title'])

    if title != '':
        idx = indices[title]
    else:
        idx = indices['Current']

    sim_scores = list(enumerate(cosine_sim[idx]))

    sim_scores = sorted(sim_scores, key=lambda x: x[0], reverse=False)

    sim_value = [x[1] for x in sim_scores]

    mainList['cosine_sim'] = sim_value

    return mainList
```

```
        moviesMetaData = pd.read_csv('../input/the-movies-dataset/movies_metadata.csv',
    low_memory=False)
        links = pd.read_csv('../input/the-movies-dataset/links.csv')

        moviesMetaData['id'] = moviesMetaData['id'].astype(str)
        links['tmdbId'] = links['tmdbId'].fillna(0.0).astype(int)
        links['tmdbId'] = links['tmdbId'].astype(str)
        MovieRatingForecast['movieId'] = MovieRatingForecast['movieId'].astype(str)
        links['movieId'] = links['movieId'].astype(str)

        moviesMetaData.rename(columns={'id': 'MMD_id'}, inplace = True)
        links.rename(columns={'movieId': 'L_id'}, inplace = True)
        MovieRatingForecast.rename(columns={'movieId': 'movieId_r'}, inplace = True)

        moviesMetaData = moviesMetaData.merge(links, left_on='MMD_id', right_on='tmdbId')
        moviesMetaData = moviesMetaData.merge(MovieRatingForecast, left_on='L_id',
    right_on='movieId_r')
        moviesMetaData = moviesMetaData.drop_duplicates()
        moviesMetaData.rename(columns={'MMD_id': 'id'}, inplace = True)

return moviesMetaData[['id', 'title', 'genres', 'rating', 'nusers']]
```

Додаток Г

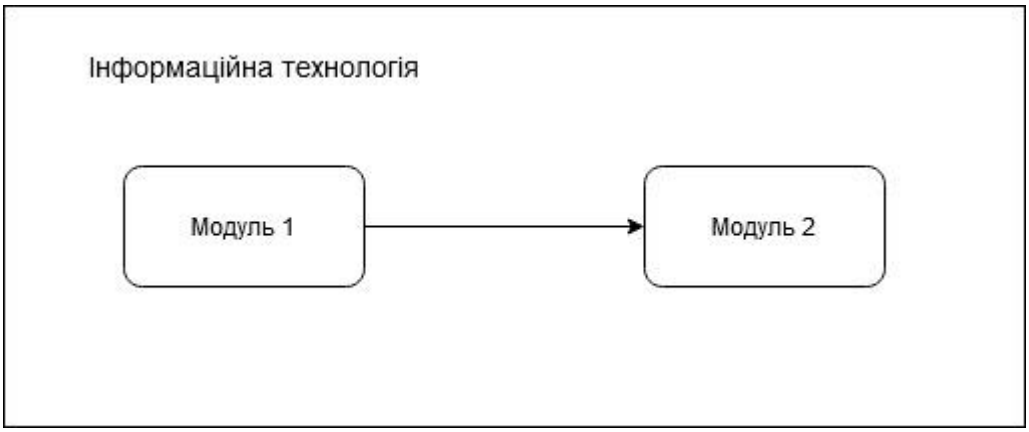
Графічна частина

UML діаграма діяльності основних сценаріїв роботи інформаційної технології



					08-53.МКР.001.03.000.ПЛ		
					Інформаційна технологія аналізу та рекомендування кінофільмів для глядачів	Літ.	Маса	Масштаб
Змн.	Арк.	№ докум.	Підпис	Дата				1 : 1
Розроб.		Корпало А. В.				Арк. 1	Аркушів 8	
Перевір.		Козачко О. М.						
Т. Контр.								
Реценз.		Бойко О.Р.			UML діаграма діяльності основних сценаріїв роботи інформаційної технології	2ІСТ-19м		
Н. Контр.		Жуков С. О.						
Затверд.		Мокін В.Б.						

Структурна схема інформаційної технології



					08-53.МКР.001.03.000.ПЛ		
					Інформаційна технологія аналізу та рекомендування кінофільмів для глядачів	Літ.	Маса	Масштаб
Змн.	Арк.	№ докум.	Підпис					1 : 1
Розроб.		Корпало А. В.				Арк. 2	Аркушів 8	
Перевір.		Козачко О. М.						
Т. Контр.								
Реценз.		Бойко О.Р.			Структурна схема інформаційної технології	2ІСТ-19м		
Н. Контр.		Жуков С. О.						
Затверд.		Мокін В.Б.						

UML діаграма діяльності модулю спільної фільтрації інформаційної технології

					08-53.МКР.001.03.000.ПЛ		
					Інформаційна технологія аналізу та рекомендування кінофільмів для глядачів	Літ.	Маса	Масштаб
Змн.	Арк.	№ докум.	Підпис					1 : 1
Розроб.		Корпало А. В.				Арк. 3	Аркушів 8	
Перевір.		Козачко О. М.						
Т. Контр.					UML діаграма діяльності модулю спільної фільтрації інформаційної технології	2ІСТ-19м		
Реценз.		Бойко О.Р.						
Н. Контр.		Жуков С. О.						
Затверд.		Мокін В.Б.						

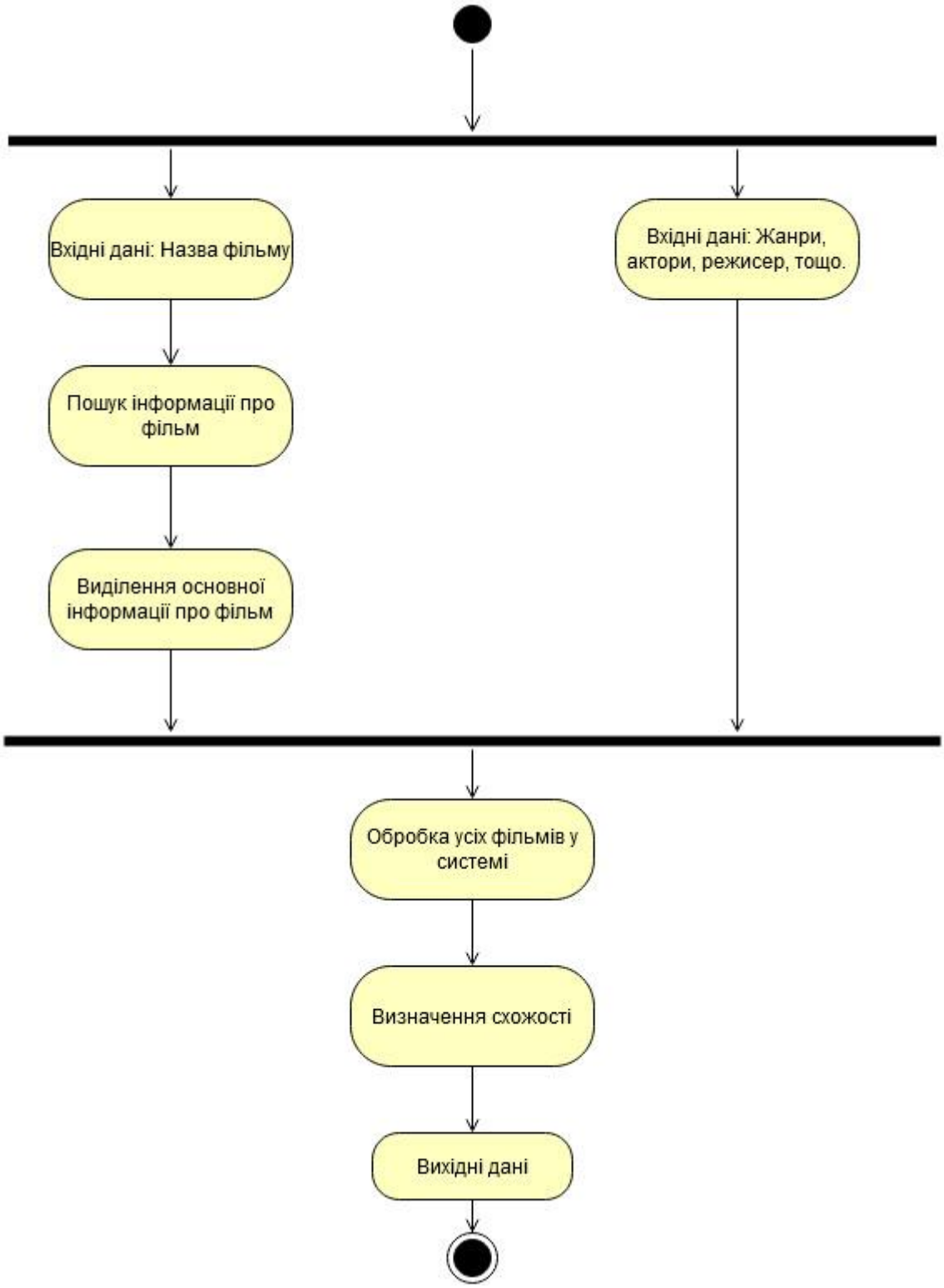
UML діаграма діяльності другого сценарію роботи модулю фільтрації на основі вмісту інформаційної технології

					08-53.МКР.001.03.000.ПЛ		
					Інформаційна технологія аналізу та рекомендування кінофільмів для глядачів	Літ.	Маса / Масштаб
Змн.	Арк.	№ докум.	Підпис				1 : 1
Розроб.		Корпало А. В.				Арк. 4	Аркушів 8
Перевір.		Козачко О. М.					
Т. Контр.							
Реценз.		Бойко О.Р.			UML діаграма діяльності другого сценарію роботи модулю фільтрації на основі вмісту інформаційної технології	2ІСТ-19м	
Н. Контр.		Жуков С. О.					
Затверд.		Мокін В.Б.					

UML діаграма діяльності третього сценарію роботи модулю фільтрації на основі вмісту інформаційної технології

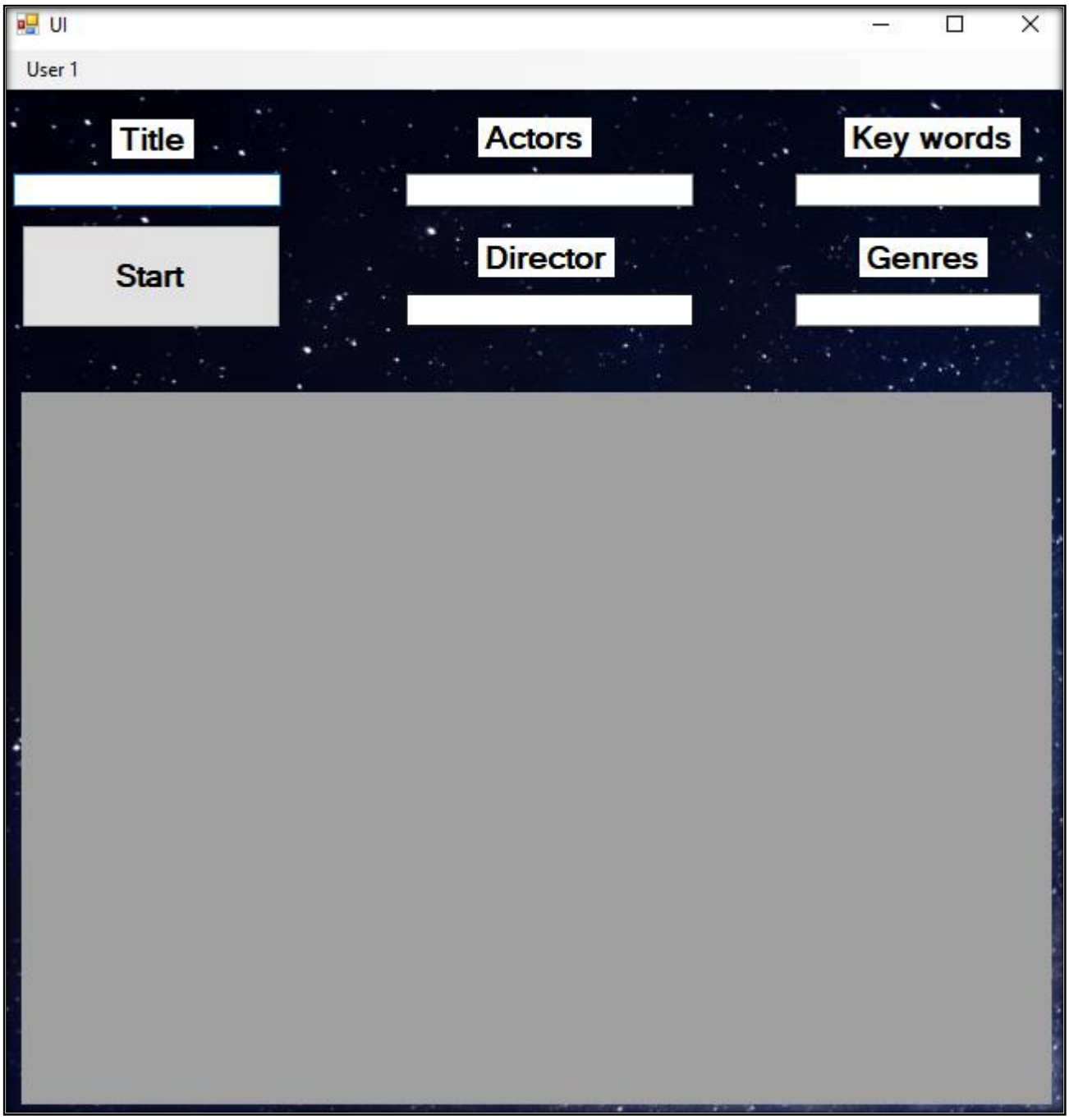
					08-53.МКР.001.03.000.ПЛ		
					Інформаційна технологія аналізу та рекомендування кінофільмів для глядачів	Літ.	Маса	Масштаб
Змн.	Арк.	№ докум.	Підпис					1 : 1
Розроб.		Корпало А. В.				Арк. 5	Аркушів 8	
Перевір.		Козачко О. М.						
Т. Контр.								
Реценз.		Бойко О.Р.			UML діаграма діяльності третього сценарію роботи модулю фільтрації на основі вмісту інформаційної технології	2ІСТ-19м		
Н. Контр.		Жуков С. О.						
Затверд.		Мокін В.Б.						

UML діаграма діяльності модулю фільтрації на основі вмісту інформаційної технології



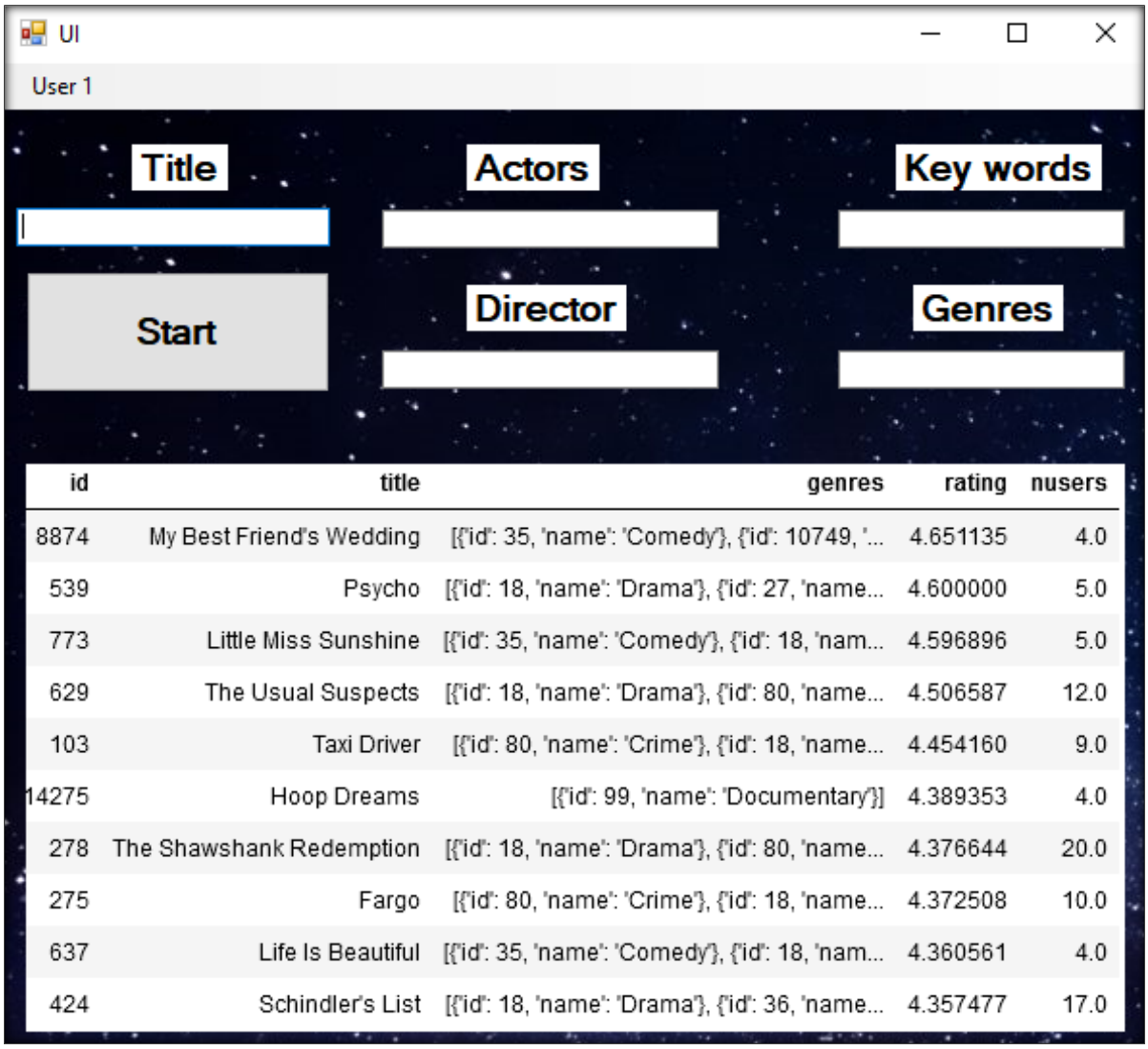
					08-53.МКР.001.03.000.ПЛ		
					Інформаційна технологія аналізу та рекомендування кінофільмів для глядачів	Літ.	Маса	Масштаб
Змн.	Арк.	№ докум.	Підпис					1 : 1
Розроб.		Корпало А. В.				Арк. 6	Аркушів 8	
Перевір.		Козачко О. М.						
Т. Контр.								
Реценз.		Бойко О.Р.			UML діаграма діяльності модулю фільтрації на основі вмісту інформаційної технології	2ІСТ-19м		
Н. Контр.		Жуков С. О.						
Затверд.		Мокін В.Б.						

Головне вікно програми



					08-53.МКР.001.03.000.ПЛ		
					Інформаційна технологія аналізу та рекомендування кінофільмів для глядачів	Літ.	Маса	Масштаб
Змн.	Арк.	№ докум.	Підпис					1 : 1
Розроб.		Корпало А. В.				Арк. 7	Аркушів 8	
Перевір.		Козачко О. М.						
Т. Контр.								
Реценз.		Бойко О.Р.			Головне вікно програми	2ІСТ-19м		
Н. Контр.		Жуков С. О.						
Затверд.		Мокін В.Б.						

Результат роботи головного сценарію інформаційної технології



id	title	genres	rating	nusers
8874	My Best Friend's Wedding	[{'id': 35, 'name': 'Comedy'}, {'id': 10749, '...	4.651135	4.0
539	Psycho	[{'id': 18, 'name': 'Drama'}, {'id': 27, 'name...	4.600000	5.0
773	Little Miss Sunshine	[{'id': 35, 'name': 'Comedy'}, {'id': 18, 'nam...	4.596896	5.0
629	The Usual Suspects	[{'id': 18, 'name': 'Drama'}, {'id': 80, 'name...	4.506587	12.0
103	Taxi Driver	[{'id': 80, 'name': 'Crime'}, {'id': 18, 'name...	4.454160	9.0
14275	Hoop Dreams	[{'id': 99, 'name': 'Documentary'}]	4.389353	4.0
278	The Shawshank Redemption	[{'id': 18, 'name': 'Drama'}, {'id': 80, 'name...	4.376644	20.0
275	Fargo	[{'id': 80, 'name': 'Crime'}, {'id': 18, 'name...	4.372508	10.0
637	Life Is Beautiful	[{'id': 35, 'name': 'Comedy'}, {'id': 18, 'nam...	4.360561	4.0
424	Schindler's List	[{'id': 18, 'name': 'Drama'}, {'id': 36, 'name...	4.357477	17.0

					08-53.МКР.001.03.000.ПЛ		
					Інформаційна технологія аналізу та рекомендування кінофільмів для глядачів	Літ.	Маса	Масштаб
Змн.	Арк.	№ докум.	Підпис					1 : 1
Розроб.		Корпало А. В.				Арк. 8	Аркушів 8	
Перевір.		Козачко О. М.						
Т. Контр.								
Реценз.		Бойко О.Р.			Результат роботи головного сценарію інформаційної технології			
Н. Контр.		Жуков С. О.				2ІСТ-19м		
Затверд.		Мокін В.Б.						