МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління

Середа Іван Володимирович

Програмний модуль контентних рекомендацій фільмів і серіалів Netflix на основі косинусної подібності текстових описів / Software module for content-based recommendations of Netflix movies and TV shows using cosine similarity of textual descriptions

Спеціальність 122 «Комп'ютерні науки»
освітньо-професійна програма – Комп'ютерні науки
Кваліфікаційна робота

Виконав студент групи КНШІ-41
І.В. Середа
_____

Науковий керівник
к.т.н., доцент Т.В. Лендюк
_____

Кваліфікаційну роботу
допущено до захисту
«___»_____20___р.

Завідувач кафедри
_____Н.В. Дзюбановська

ТЕРНОПІЛЬ – 2026

Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління
Ступінь вищої освіти «бакалавр»
спеціальність 122 – «Комп'ютерні науки»
освітньо-професійна програма –« Комп'ютерні науки

«ЗАТВЕРДЖУЮ»
В.о. завідувача кафедри
_____ Н.В. Дзюбановська
«____» _____20__р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ
СЕРЕДІ Івану Володимировичу
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи

Програмний модуль контентних рекомендацій фільмів і серіалів Netflix на основі косинусної подібності текстових описів / Software module for content-based recommendations of Netflix movies and TV shows using cosine similarity of textual descriptions

керівник роботи к.т.н., доцент Т.В. Лендюк
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджений наказом по університету від 01 грудня 2025 року № 894.
2. Строк подання студентом закінченої кваліфікаційної роботи 25 травня 2026 р.
3. Вихідні дані до кваліфікаційної роботи: завдання на кваліфікаційну роботу студента, наукові статті, технічна література.
4. Основні питання, які потрібно розробити:
– провести аналіз предметної області та обґрунтувати вибір контентного підходу;
– дослідити структуру даних Netflix та визначити правила обробки пропусків;
– реалізувати конвеєр передобробки тексту та сформувати інтегрований профіль;
– виконати векторизацію (BoW/TF-IDF) та побудувати матрицю ознак;
– обчислити матрицю косинусної подібності та реалізувати алгоритм ранжування Top-N;
– провести тестування модуля та розробити прототип інтерфейсу.
5. Перелік графічного матеріалу в роботі:
– класифікація рекомендаційних систем;
– функціональна та компонентна схеми модуля;
– концептуальна модель даних;
– діаграми аналізу структури датасету;
– результати тестування: приклади ранжування та списки рекомендацій.

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 01 грудня 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів кваліфікаційної роботи	Примітка
1	Затвердження теми кваліфікаційної роботи, ознайомлення з літературними джерелами та складання плану роботи	до 01.01. 2026 р.	
2	Написання 1 розділу кваліфікаційної роботи	до 01.02. 2026 р.	
3	Написання 2 розділу кваліфікаційної роботи	до 01.04.2026 р.	
4	Написання 3 розділу кваліфікаційної роботи	до 01.05. 2026 р.	
5	Представлення попереднього варіанту кваліфікаційної роботи, перевірка та внесення змін керівником	до 15.05.2026 р.	
6	Опрацювання зауважень та представлення завершеного варіанту кваліфікаційної роботи. Підготовка супроводжуючих документів	до 25.05.2026 р.	
7	Перевірка кваліфікаційної роботи на оригінальність тексту	до 30.05.2026 р.	
8	Оформлення кваліфікаційної роботи та отримання допуску до захисту	до 10.06.2026 р.	
9	Подання кваліфікаційної роботи до захисту на засіданні атестаційної комісії	до 10.06. 2026 р.	

Студент_____ І.В. Середа
(підпис) (прізвище та ініціали)

Керівник кваліфікаційної роботи_____ Т.В. Лендюк

АНОТАЦІЯ

Кваліфікаційна робота на тему «Програмний модуль контентних рекомендацій фільмів і серіалів Netflix на основі косинусної подібності текстових описів» на здобуття освітнього ступеня «бакалавр» зі спеціальності 122 «Комп'ютерні науки» освітньої програми «Комп'ютерні науки» виконана обсягом у 57 сторінок і містить 19 ілюстрацій, 1 таблиця, 2 додатки та 32 використаних джерела.

Метою роботи є розроблення та експериментальна перевірка програмного модуля контентних рекомендацій для об'єктів каталогу Netflix на основі формування інтегрованого текстового профілю та обчислення косинусної подібності між векторизованими описами з подальшим ранжуванням Top-N результатів.

Методами дослідження та розроблення обрано аналіз предметної області рекомендаційних систем стрімінгових сервісів, дослідницький аналіз даних та оцінювання якості метаданих, передобробку тексту (очищення, нормалізація, вилучення стоп-слів), векторно-просторове представлення контенту методами Bag-of-Words/CountVectorizer (із можливістю застосування TF-IDF), обчислення матриці косинусної подібності, а також тестування функції рекомендацій і прототипу інтерфейсу у середовищі Notebook.

У результаті виконання роботи реалізовано програмний модуль мовою Python із використанням бібліотек pandas та scikit-learn, який підтримує повний конвеєр: завантаження датасету Netflix (8807 записів, 12 атрибутів), обробку пропусків (зокрема заміну відсутніх значень у полі director на маркер Unknown та вилучення записів із критичними пропусками у полях cast і country), формування інтегрованого текстового поля overall_infos, векторизацію та обчислення матриці косинусної подібності N×N. Практичні випробування показали коректність ранжування рекомендацій: для запиту «LEGO Ninjago» максимальна подібність у Top-1 становить 0,466, а для запиту «Teenage Mutant

Ninja Turtles» — 0,333, зі спаданням значень у межах Top-5, що забезпечує інтерпретовану підтримку вибору контенту.

Результати роботи можуть бути використані як базове відтворюване рішення для навчальних і демонстраційних задач рекомендаційних систем, а також як основа для подальшого розвитку (перехід до TF-IDF і семантичних подань, гібридизація з поведінковими сигналами, оптимізація обчислень та розгортання у вебінтерфейсі).

Ключові слова: РЕКОМЕНДАЦІЙНА СИСТЕМА, КОНТЕНТНА ФІЛЬТРАЦІЯ, NETFLIX, ОБРОБКА ПРИРОДНОЇ МОВИ, ВЕКТОРИЗАЦІЯ ТЕКСТУ, COUNT VECTORIZER, TF-IDF, КОСИНУСНА ПОДІБНІСТЬ, РАНЖУВАННЯ, TOP-N.

ANNOTATION

The qualification work entitled «Software Module for Content-Based Recommendations of Netflix Movies and TV Shows Using Cosine Similarity of Textual Descriptions», submitted for the Bachelor's degree in speciality 122 «Computer Science» within the educational and professional program «Computer Science», comprises 57 pages and includes 19 figures, 1 tables, 2 annexes, and 32 references.

The purpose of the work is to develop and experimentally validate a content-based recommendation software module for Netflix catalogue items by constructing an integrated textual profile and computing cosine similarity between vectorized descriptions, followed by Top-N ranking.

The research and development methods include analysis of recommender systems in the context of streaming services, exploratory data analysis and metadata quality assessment, text preprocessing (cleaning, normalization, stop-word removal), vector-space content representation using Bag-of-Words/CountVectorizer (with an option to apply TF-IDF), cosine-similarity matrix computation, and functional testing of the recommendation routine and a Notebook-based user interface prototype.

As a result, a Python module was implemented using pandas and scikit-learn. The full pipeline covers loading the Netflix dataset (8,807 records, 12 attributes), handling missing values (e.g., filling missing director values with the marker Unknown and removing records with critical missing values in cast and country), building the integrated field overall_infos, vectorization, and constructing an N×N cosine-similarity matrix. Empirical tests confirm correct recommendation ranking: for the query «LEGO Ninjago» the Top-1 similarity reaches 0.466, and for «Teenage Mutant Ninja Turtles» it reaches 0.333, with decreasing similarity values across the Top-5 list, enabling interpretable decision support for content selection.

The results can be used as a reproducible baseline for educational and demonstration purposes in recommender systems, and as a foundation for further

improvements (TF-IDF and semantic representations, hybridization with behavioural signals, computational optimization, and deployment in a web interface).

Keywords: RECOMMENDER SYSTEM, CONTENT-BASED FILTERING, NETFLIX, NATURAL LANGUAGE PROCESSING, TEXT VECTORIZATION, COUNT VECTORIZER, TF-IDF, COSINE SIMILARITY, RANKING, TOP-N.

ЗМІСТ

ВСТУП

Стрімінгові сервіси стали однією з ключових інфраструктур цифрового споживання аудіовізуального контенту, забезпечуючи користувачам доступ до каталогів, що налічують тисячі фільмів і серіалів. Зростання обсягів контенту, його жанрова та мовна неоднорідність, а також висока динаміка оновлення бібліотек призводять до ефекту інформаційного перевантаження: навіть за наявності фільтрів і пошуку користувачеві складно оперативно обрати релевантний об’єкт для перегляду. У такому середовищі рекомендаційні системи виступають прикладним інструментом підтримки прийняття рішень, виконуючи функції персоналізованої фільтрації та ранжування й тим самим підвищуючи зручність навігації в каталозі.

Традиційні механізми підбору контенту на основі ручного пошуку або жорстких фільтрів (жанр, рік, країна) не враховують латентні текстові сигнали та комбінований вплив метаданих (опис, акторський склад, режисер, тематичні категорії). Додатково практичні обмеження пов’язані з неповнотою та варіативністю метаданих: частина записів містить пропуски (наприклад, режисер або акторський склад), описи мають різну довжину та стилістику, а перелік категорій подається як множинний рядковий атрибут. За таких умов актуальною є побудова формалізованого контентного профілю об’єктів і застосування стійких методів текстового зіставлення, здатних працювати за відсутності поведінкової історії користувачів.

Суттєвим викликом для рекомендаційних систем є проблема «холодного старту», коли нові об’єкти або нові користувачі не мають достатньої кількості взаємодій для застосування колаборативних методів. У рамках кваліфікаційної роботи, де часто доступні лише відкриті метадані без індивідуальних логів перегляду, контентний підхід є методично обґрунтованою альтернативою. Його практична перевага полягає в тому, що рекомендації формуються на основі безпосередньої схожості описів і атрибутів контенту, а отже забезпечується

відтворюваність експериментів і можливість інтерпретації результатів через спільні терміни й ознаки.

З огляду на постановку задачі, доцільним є використання векторно-просторової моделі представлення тексту, в якій кожному об’єкту каталогу відповідає числовий вектор ознак. Для побудови таких векторів широко застосовують підходи «мішок слів» або TF-IDF, що дозволяють перетворити текстові поля та частину метаданих (жанри, актори, режисер) у формалізований профіль. Як міра близькості доцільною є косинусна подібність, яка оцінює кут між векторами та є стійкою до різної довжини описів, що типово для каталогів стрімінгових платформ. У результаті формується матриця попарних подібностей, яка забезпечує ранжування кандидатів і відбір Top-N рекомендацій у прикладних сценаріях.

Отже, актуальним є розроблення програмного модуля контентних рекомендацій для каталогу стрімінгового сервісу на основі текстових метаданих із реалізацією повного конвеєра: підготовка та очищення даних, формування інтегрованого текстового профілю, векторизація, обчислення подібності та інтерфейсний прототип для тестування. Практична цінність такого модуля визначається не лише коректністю математичної постановки, а й технологічною відтворюваністю реалізації, можливістю контролю якості даних, а також наявністю прозорого механізму отримання рекомендацій із числовими оцінками подібності.

Мета роботи — розробити та експериментально перевірити програмний модуль контентних рекомендацій для об’єктів каталогу Netflix на основі текстового профілю та косинусної подібності векторизованих описів, забезпечивши формування ранжованих списків Top-N і базову інтерпретацію результатів через значення метрики подібності та вивід ключових метаданих.

Для досягнення мети поставлено такі завдання:

– проаналізувати предметну область рекомендаційних систем у контексті стрімінгових сервісів та обґрунтувати вибір контентного підходу для умов обмежених поведінкових даних;

– дослідити структуру та якість набору даних Netflix, визначити набір релевантних атрибутів і правила обробки пропусків;

– реалізувати алгоритмічний конвеєр передобробки текстових даних (очищення, нормалізація, токенізація/фільтрація) та сформувати інтегрований текстовий профіль об’єкта;

– реалізувати векторизацію текстових профілів (BoW або TF-IDF) та побудувати матрицю ознак;

– обчислити матрицю косинусної подібності, реалізувати механізм ранжування та відбору Top-N рекомендацій для заданого об’єкта;

– провести тестування модуля на контрольних запитах та реалізувати прототип інтерфейсу взаємодії, що повертає рекомендації разом із числовими оцінками подібності й метаданими для інтерпретації.

Об’єкт дослідження — процес автоматизованого формування рекомендацій у стрімінгових інформаційних системах на основі метаданих контенту.

Предмет дослідження — методи та програмні засоби контентної фільтрації для рекомендацій фільмів/серіалів, що включають передобробку тексту, векторизацію описів (BoW/TF-IDF), обчислення косинусної подібності та алгоритми ранжування Top-N результатів.

Методи дослідження: методи інтелектуального аналізу тексту та інформаційного пошуку (нормалізація тексту, вилучення стоп-слів, векторно-просторові моделі), методи статистичної векторизації (BoW, TF-IDF), методи обчислення міри подібності (косинусна подібність), методи експериментального тестування програмних модулів і аналізу результатів ранжування.

Практичне значення роботи полягає у створенні відтворюваного програмного модуля контентних рекомендацій для каталогу Netflix, який може бути використаний як базове рішення для інтеграції в інформаційні системи стрімінгового типу, навчальні та демонстраційні середовища, а також як основа для подальших удосконалень (перехід до TF-IDF/семантичних подань,

гібридизація з поведінковими сигналами, оптимізація обчислень і розгортання в вебінтерфейсі).

Апробація результатів дослідження здійснена в межах XI Міжнародної студентської конференції «Сучасні аспекти та перспективні напрямки розвитку науки», яка відбулася в місті Запоріжжі (Україна) 6 березня 2026 року.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ РЕКОМЕНДАЦІЙНИХ СИСТЕМ

1.1 Аналіз предметної області стрімінгових сервісів

Стрімінгові платформи є цифровими інформаційними системами, що забезпечують інтерактивний доступ до каталогу аудіовізуального контенту (фільми, серіали, документалістика тощо) у режимі запиту користувача. Для цієї предметної області визначальними є три взаємопов’язані характеристики: масштаб контентної бібліотеки, неоднорідність метаданих, динаміка споживання. Масштаб каталогу зумовлює складність навігації, оскільки за наявності тисяч позицій навіть структурований пошук за жанром, роком або країною виробництва не гарантує швидкого знаходження релевантного контенту. Неоднорідність метаданих проявляється у варіативності якості описів: частина позицій має розгорнуті анотації, частина — короткі синопсиси, а також різні набори атрибутів (теги, акторський склад, режисер, категорії, вікові обмеження). Динаміка споживання характеризується швидкою зміною інтересів і контексту перегляду (час доби, настрій, тип пристрою, перегляд наодинці чи з родиною), що підсилює потребу в адаптивних механізмах добору.

У межах стрімінгових сервісів рекомендаційний модуль слугує інструментом підтримки прийняття рішень користувачем: він виконує функції фільтрації, ранжування і персоналізованого представлення контенту, зменшуючи інформаційне перевантаження. Інформаційне перевантаження в даному контексті є наслідком розриву між обсягом доступних альтернатив і когнітивними можливостями користувача виконати коректне порівняння варіантів. У результаті зростає частка випадкового вибору, скорочується час на взаємодію з каталогом, а також підвищується ризик незадоволеності переглядом через нерелевантність обраного контенту.

Для типового стрімінгового сервісу важливою є також прикладна вимога до оперативності: рекомендації мають формуватися в межах прийнятного часу інтерфейсної взаємодії. Тому практичні рішення часто будуються як багатокрокові конвеєри: підготовка даних (офлайн), обчислення

подібності/моделі (офлайн або напівофлайн) та швидкий інференс (онлайн). Для контентного підходу на основі текстових описів це означає, що критичними етапами є коректна передобробка тексту, стабільна векторизація та ефективне обчислення міри схожості.

З огляду на постановку теми роботи, ключовою інформаційною сутністю є контентний профіль об’єкта (фільму/серіалу), сформований із його текстових характеристик. Саме цей профіль використовується для визначення подібності між позиціями каталогу та генерації рекомендацій. Текстові описи, попри їхню лінгвістичну варіативність, є універсальним джерелом ознак, яке дозволяє будувати рекомендації навіть у ситуаціях обмеженої поведінкової історії користувачів.

1.2 Класифікація рекомендаційних систем

Рекомендаційні системи — це програмно-алгоритмічні компоненти інформаційних систем, призначені для автоматизованого добору та ранжування об’єктів відповідно до цільової функції (релевантність, ймовірність перегляду, очікувана задоволеність, різноманітність тощо). Залежно від типу даних і принципу формування рекомендацій, у практиці найчастіше виокремлюють контентні, колаборативні та гібридні підходи. Узагальнену класифікацію підходів до побудови рекомендаційних систем подано на рисунку 1.1.

Як видно з рисунка 1.1, рекомендаційні системи поділяються на три основні класи: контентні, колаборативні та гібридні. Контентні системи ґрунтуються на аналізі характеристик об’єктів контенту (описів, жанрів, ключових слів). Колаборативні системи використовують дані взаємодій користувачів із контентом, формуючи рекомендації на основі схожості профілів або об’єктів. Гібридні системи поєднують обидва підходи для підвищення якості рекомендацій.

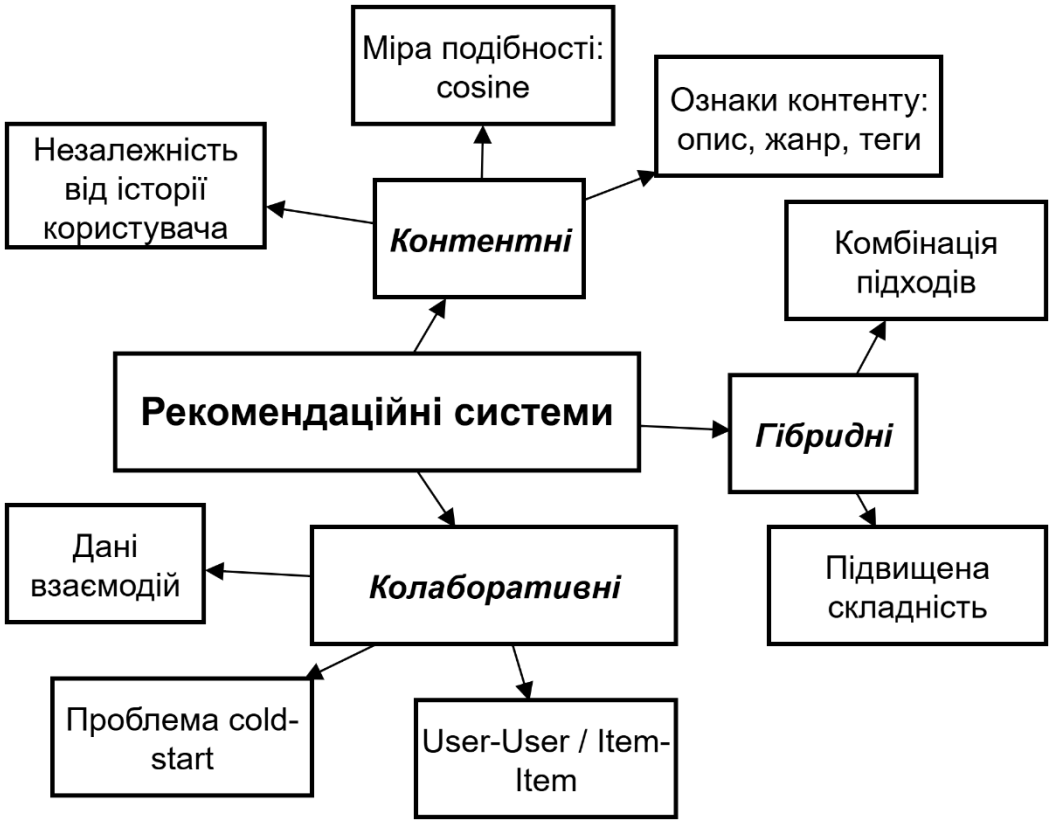

Рисунок 1.1 – Класифікація рекомендаційних систем

Контентні рекомендаційні системи (content-based filtering) формують рекомендації на основі схожості атрибутів об’єктів контенту. У випадку стрімінгових сервісів такими атрибутами можуть виступати текстові описи, жанрові категорії, ключові слова, склад акторів, країна виробництва тощо. Перевагою контентного підходу є відносна незалежність від масивних історичних даних користувацьких взаємодій, що суттєво знижує проблему «холодного старту» для нових об’єктів. Додатково цей підхід може бути інтерпретованим: рекомендація обґрунтовується спільністю ознак (наприклад, близькістю векторів TF-IDF описів). Обмеження контентних систем пов’язані з якістю метаданих і ризиком надмірної спеціалізації: система може «замикатися» на вузькому тематичному сегменті, пропонуючи надто схожі позиції.

Колаборативні рекомендаційні системи (collaborative filtering) використовують дані про взаємодії користувачів із контентом (оцінювання, перегляди, тривалість перегляду, кліки) і формують рекомендації через виявлення схожих користувачів або схожих об’єктів у просторі взаємодій.

Перевагою є здатність захоплювати приховані закономірності споживання, які не описані явними атрибутами контенту. Водночас колаборативні системи суттєво залежать від обсягу та якості поведінкових даних і є чутливими до розрідженості матриці взаємодій, що ускладнює запуск і підтримку таких рішень за обмежених даних.

Гібридні рекомендаційні системи (hybrid approaches) поєднують контентні та колаборативні механізми з метою компенсації слабких сторін кожного підходу. Гібридизація дозволяє зменшити ефект холодного старту, підвищити різноманітність рекомендацій та забезпечити стійкішу якість на різних сегментах користувачів. Недоліком гібридних систем є підвищена складність архітектури, налаштування й оцінювання, а також необхідність підтримки кількох контурів даних і моделей.

У межах цієї кваліфікаційної роботи обґрунтовано вибір контентної рекомендаційної моделі, оскільки вона безпосередньо узгоджується з доступним типом вхідних даних (текстові описи) та забезпечує відтворювану реалізацію модуля на основі формалізованої міри подібності — косинусної подібності векторизованих описів.

1.3 Аналіз існуючих алгоритмів текстової подібності

Алгоритми текстової подібності становлять основу контентних рекомендаційних систем, оскільки забезпечують формалізоване порівняння описів об’єктів у векторному просторі ознак. У межах векторно-просторової моделі документ подається як вектор, а релевантність/схожість визначається через функції подібності або відстані, що дає змогу виконувати ранжування та відбір Top-N рекомендацій для заданого запиту [1–4].

Початковим і найбільш поширеним способом представлення тексту є Bag-of-Words (BoW), за якого документ характеризується частотами термінів без урахування порядку слів. Перевагами BoW є простота реалізації, інтерпретованість і сумісність із класичними моделями інформаційного пошуку

[1, 5]. Водночас BoW ігнорує синтаксичні зв’язки та семантичні відношення, формує високовимірний розріджений простір і є чутливим до синонімії та полісемії, що потенційно знижує якість зіставлення описів, які передають схожий зміст різними лексемами [1, 6].

Для усунення впливу надмірно частотних слів і підсилення дискримінативних термінів використовується TF-IDF (term frequency–inverse document frequency), який зважує локальну частоту терміна в документі на глобальну рідкісність терміна в колекції [2, 7, 8]. У задачах пошуку та рекомендацій TF-IDF зазвичай забезпечує більш якісне ранжування порівняно з «чистими» частотами, оскільки зменшує внесок загальних слів та підвищує внесок тематично значущих токенів [2, 7]. Для Netflix-описів це є важливим через типову наявність шаблонних конструкцій, які не мають визначального впливу на зміст і не повинні домінувати в подібності [1–3].

Практична реалізація TF-IDF у програмних модулях часто виконується з використанням стандартних бібліотечних засобів, що забезпечує відтворюваність експериментів і узгодженість обчислень. Зокрема, TfidfVectorizer (scikit-learn) дозволяє керувати токенізацією, обмеженням словника, нормалізацією та вагами, що безпосередньо впливає на стабільність подальшого оцінювання подібності між описами [9].

Після формування TF-IDF-векторів базовою метрикою зіставлення виступає косинусна подібність, яка оцінює кут між векторами і, за умови нормалізації, знижує залежність від довжини тексту [1, 10]. Для контентних рекомендацій це означає, що об’єкти зі схожим «термінологічним профілем» отримують високі оцінки схожості навіть за різних обсягів опису. Косинусна подібність також зручна тим, що допускає ефективні пакетні обчислення (матриця подібності) та пряме ранжування кандидатів для формування рекомендацій [10, 11].

Як альтернатива косинусній подібності, міра Жаккара (Jaccard) є природною для представлення тексту як множини токенів або бінарних ознак і оцінює частку перетину відносно об’єднання [12, 13]. Для задач, де важлива

наявність спільних термінів, Jaccard може бути інтерпретованою метрикою, однак у класичному вигляді вона не враховує ваги термінів та їхню статистичну значущість у колекції, тому у випадку коротких описів може бути менш чутливою до змістових нюансів порівняно з TF-IDF + cosine [1, 12].

Ще одним підходом є використання евклідової відстані, проте в задачах порівняння високовимірних розріджених векторів вона часто демонструє небажані властивості: відстані можуть «концентруватися» і втрачати розрізнювальну здатність, а також посилюється чутливість до масштабу компонент [14]. Навіть за нормалізації евклідова відстань зазвичай поступається кутовим метрикам у сценаріях, де визначальним є напрямок вектора (тематична структура), а не його модуль [1, 14].

Отже, для реалізації програмного модуля контентних рекомендацій на основі текстових описів доцільно застосувати зв'язку TF-IDF-векторизації та косинусної подібності як компроміс між якістю, інтерпретованістю та простотою впровадження [1–3, 9–11]. Такий підхід є типовим базовим рішенням у прикладних системах і може бути розширений у подальших дослідженнях за рахунок семантичних подань (word embeddings, трансформери) або гібридизації з поведінковими сигналами [18–20, 23–27].

1.4 Постановка задачі

Актуальність теми зумовлена стрімким зростанням ролі стрімінгових сервісів як базових цифрових платформ споживання аудіовізуального контенту та одночасним збільшенням обсягів їхніх каталогів. Для користувача масштаб бібліотеки (тисячі позицій) трансформується в проблему навігації: навіть за наявності фільтрів за жанром, роком випуску або країною виробництва процес вибору часто залишається повільним і когнітивно затратним. У таких умовах зростає ризик інформаційного перевантаження, випадкового вибору та незадоволеності переглядом, а для платформи — ризик зниження залученості та скорочення часу взаємодії з каталогом. Тому рекомендаційні модулі виступають

ключовою прикладною складовою стрімінгових інформаційних систем, оскільки забезпечують фільтрацію, ранжування та релевантне представлення контенту у відповідь на запит користувача.

Додатковим чинником актуальності є неоднорідність і неповнота метаданих, притаманна реальним каталогам стрімінгових сервісів. Частина записів має розгорнуті описи, частина — короткі синопсиси; деякі позиції містять детальну інформацію про акторський склад і режисера, тоді як інші — мають пропуски або узагальнені значення. Це створює обмеження для колаборативних моделей, які потребують поведінкових даних (перегляди, оцінювання, кліки), що не завжди доступні у відкритих наборах або є комерційно чутливими. Відповідно, практично значущим є контентний підхід, який може працювати лише на основі описових атрибутів об’єктів і забезпечує відтворюваність експериментів у навчально-дослідницькому форматі.

З позицій класифікації рекомендаційних систем, вибір контентної фільтрації є актуальним ще й тому, що вона дозволяє формально інтерпретувати отриманий результат через близькість ознак. На відміну від колаборативних методів, де рекомендації часто пояснюються опосередковано через «схожих користувачів», контентна модель дає можливість обґрунтувати рекомендацію спільністю термінів у профілях об’єктів (жанрових маркерів, ключових слів опису, повторюваних імен акторів тощо). Для стрімінгових сервісів це важливо, оскільки прозорість і пояснюваність рекомендацій підвищує довіру користувача до результатів добору та зменшує ймовірність відмови від запропонованих варіантів.

Окрему актуальність має підбір адекватної методики текстової подібності для описів контенту. Оскільки текстові анотації формуються природною мовою, вони містять шум (службові слова, стандартні конструкції), варіативність формулювань та різний обсяг інформації. Це вимагає застосування алгоритмів, які з одного боку є достатньо простими для реалізації у вигляді модуля, а з іншого — забезпечують стабільну дискримінацію між тематично близькими та далекими описами. Саме тому векторно-просторові методи (BoW/TF-IDF) у

поєднанні з косинусною подібністю залишаються актуальними як базове, відтворюване та інтерпретоване рішення для контентних рекомендацій.

Практична значущість аналізу алгоритмів текстової подібності (підрозділ 1.3) також зумовлена потребою компромісу між якістю та обчислювальною ефективністю. У прикладних сценаріях рекомендації мають формуватися швидко, в межах допустимого часу інтерфейсної взаємодії, тому доцільними є підходи, що допускають попереднє (офлайн) формування матриці ознак і подібності та швидкий (онлайн) відбір Top-N. TF-IDF зменшує домінування частотних загальних слів і підсилює вплив термінів, що краще характеризують тематику об’єкта, тоді як косинусна подібність забезпечує коректне порівняння векторів за їхнім напрямком у просторі ознак, знижуючи залежність від довжини опису.

Таким чином, актуальність роботи визначається сукупністю факторів: масштабністю та динамікою каталогів стрімінгових платформ (1.1), прикладною потребою у відтворюваних контентних рекомендаціях за обмежених поведінкових даних і вимогами до інтерпретованості (1.2), а також необхідністю обґрунтованого вибору формалізованих і ефективних методів текстової подібності для побудови рекомендаційного конвеєра на основі метаданих (1.3).

Мета роботи — розробити та експериментально перевірити програмний модуль контентних рекомендацій для об’єктів каталогу Netflix на основі текстового профілю та косинусної подібності векторизованих описів, забезпечивши формування ранжованих списків Top-N і базову інтерпретацію результатів через значення метрики подібності та вивід ключових метаданих.

Для досягнення мети поставлено такі завдання:

1. Проаналізувати предметну область рекомендаційних систем у контексті стрімінгових сервісів та обґрунтувати вибір контентного підходу для умов обмежених поведінкових даних.

2. Дослідити структуру та якість набору даних Netflix, визначити набір релевантних атрибутів і правила обробки пропусків.

3. Реалізувати алгоритмічний конвеєр передобробки текстових даних (очищення, нормалізація, токенізація/фільтрація) та сформувати інтегрований текстовий профіль об’єкта.

4. Реалізувати векторизацію текстових профілів (BoW або TF-IDF) та побудувати матрицю ознак.

5. Обчислити матрицю косинусної подібності, реалізувати механізм ранжування та відбору Top-N рекомендацій для заданого об’єкта.

6. Провести тестування модуля на контрольних запитах та реалізувати прототип інтерфейсу взаємодії, що повертає рекомендації разом із числовими оцінками подібності й метаданими для інтерпретації.

2 АЛГОРИТМІЧНЕ ТА ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ МОДУЛЯ

2.1 Загальна структура модуля

Програмний модуль контентних рекомендацій призначений для формування переліку Top-N фільмів/серіалів, подібних до заданого користувачем об’єкта, на основі порівняння текстових описів та пов’язаних метаданих. Загальна логіка роботи модуля передбачає два контури: підготовчий (офлайн), де здійснюється очищення даних і побудова простору ознак, та експлуатаційний (онлайн), де виконується запит користувача, пошук індексу об’єкта й ранжування результатів за мірою подібності. Функціональну схему роботи системи подано на рисунку 2.1.

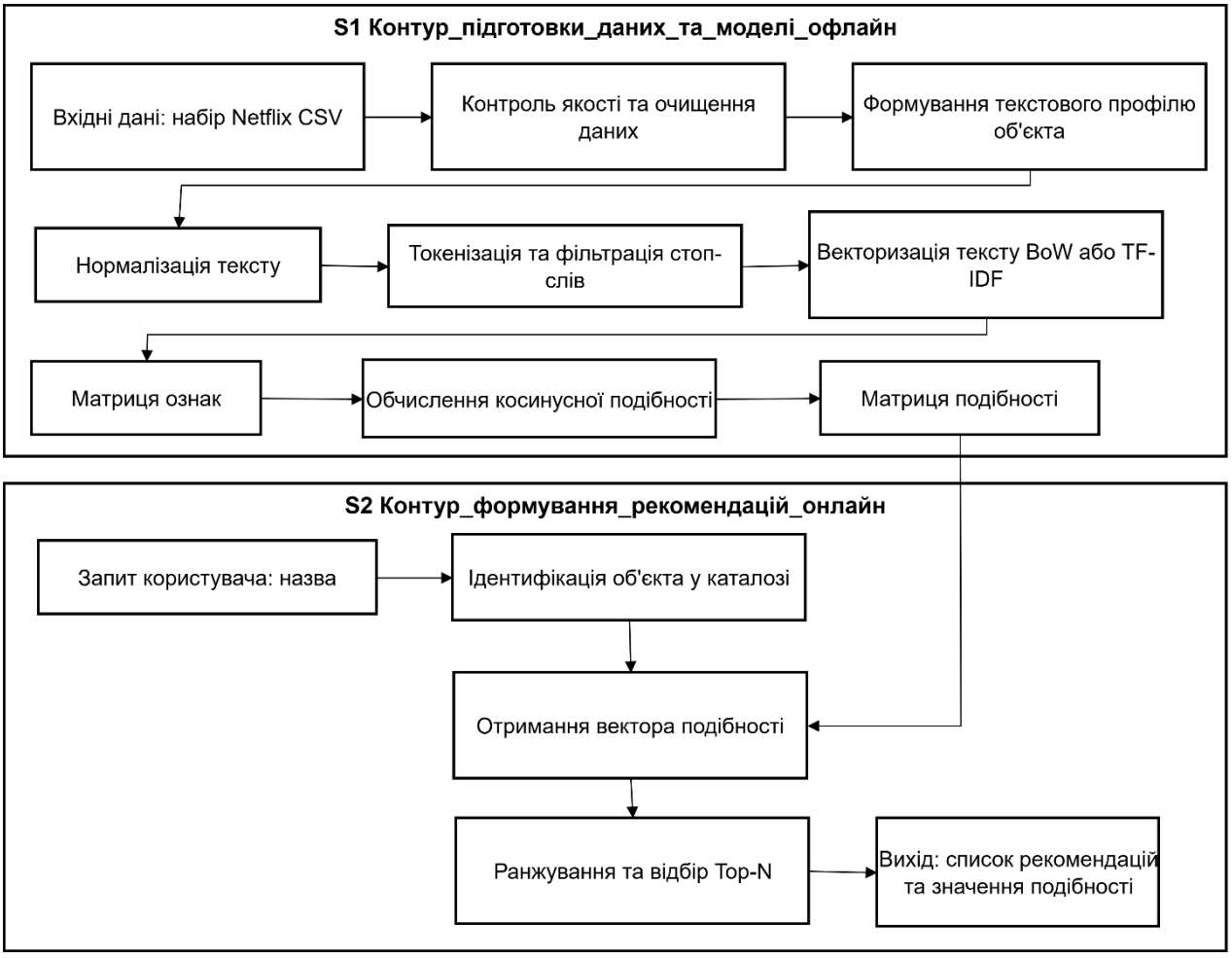

Рисунок 2.1 – Функціональна схема модуля контентних рекомендацій

Джерелом даних для модуля є відкритий набір “Netflix Movies and TV Shows”, розміщений на платформі Kaggle, який містить записи про об’єкти каталогу та їхні метадані (зокрема назву, тип, режисера, склад акторів, країну, рік випуску, опис тощо). У межах реалізації модуль завантажує таблицю з CSV-файлу та виконує первинну перевірку якості даних: ідентифікацію пропусків і прийняття рішень щодо їх обробки. Практично доцільним є підхід, за якого критичні для порівняння текстів поля (наприклад, склад акторів або опис) мають бути доступними, а пропуски в другорядних атрибутах можуть бути заповнені маркерними значеннями (наприклад, «Невідомо»), щоб уникнути невиправданої втрати записів.

Ключовою інженерною ідеєю логіки модуля є формування інтегрованого текстового представлення об’єкта шляхом конкатенації декількох ознак (тип, назва, режисер, актори, країна, опис). Це дозволяє перейти від різнорідних атрибутів до єдиного текстового “профілю” контенту, який далі піддається стандартним процедурам передобробки (нормалізація регістру, усунення пунктуації та службових фрагментів, вилучення стоп-слів) з метою підвищення стабільності векторизації й зменшення шуму в просторі ознак.

Після передобробки формується матриця ознак за допомогою методів векторизації (залежно від обраного варіанта реалізації — мішок слів або TF-IDF). На наступному кроці обчислюється матриця косинусної подібності, де кожен елемент відображає ступінь схожості двох об’єктів каталогу. Саме ця матриця використовується в експлуатаційному режимі: за введеною назвою знаходиться відповідний індекс, береться рядок подібності, виконується сортування та формується список рекомендацій без включення самого об’єкта-запиту.

Логічну архітектуру модуля доцільно подати у вигляді компонентів, що відповідають основним підсистемам обробки даних, формування ознак, обчислення подібності та інтерфейсу взаємодії. На рисунку 2.2 наведено компонентну схему (умовна UML-логіка на рівні модулів), яка відображає потоки даних і відповідальності компонентів.

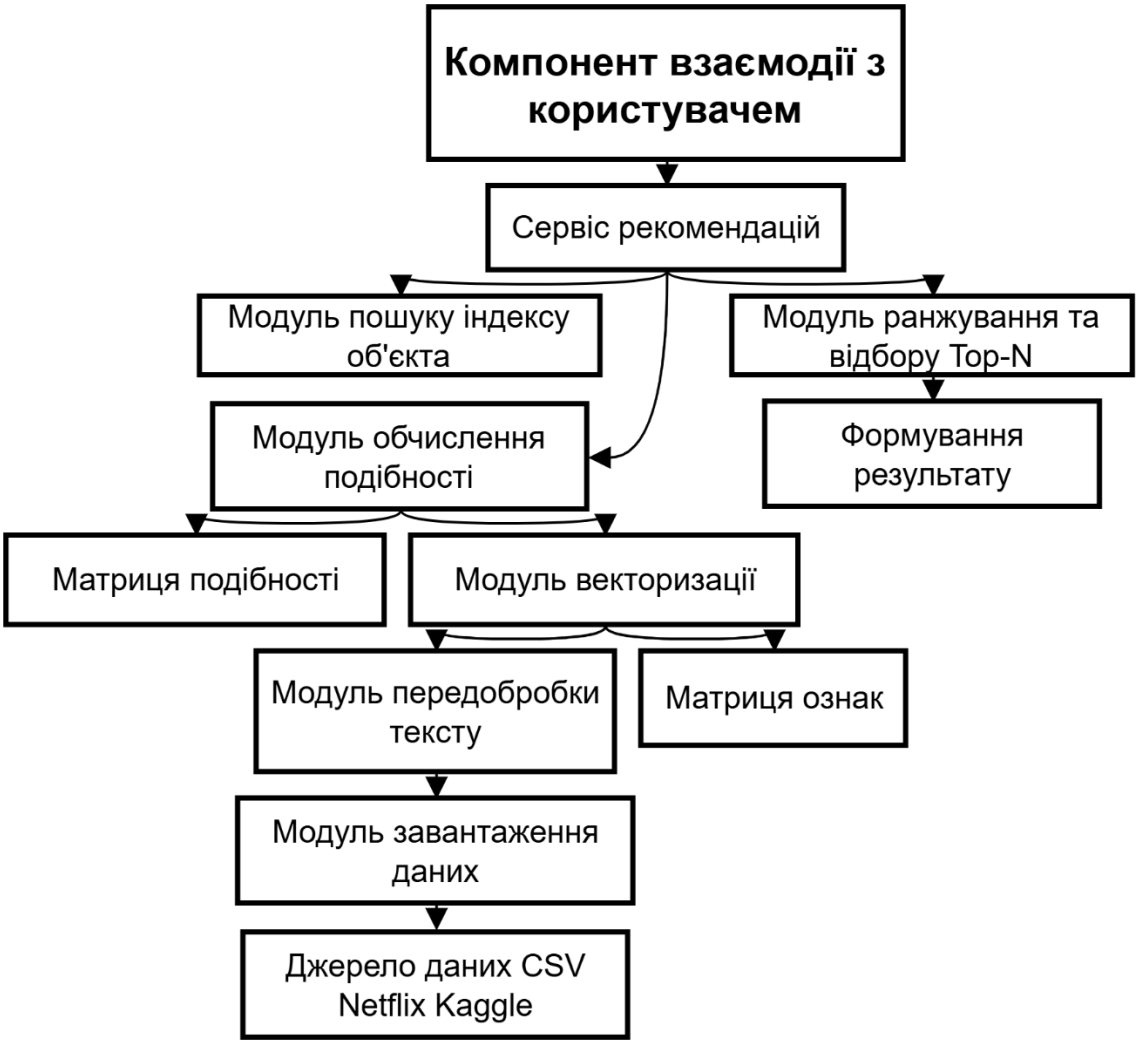

Рисунок 2.2 – Логічна архітектура програмного модуля (компонентний рівень)

Згідно з рисунком 2.2, компонент завантаження даних забезпечує доступ до CSV та формує табличне подання. Компонент передобробки тексту нормалізує інтегрований текстовий профіль об’єктів. Компонент векторизації перетворює тексти на числові вектори та формує матрицю ознак. Компонент подібності обчислює матрицю косинусної подібності. Компонент сервісу рекомендацій реалізує бізнес-логіку: пошук індексу, ранжування та формування відповіді. Компонент взаємодії з користувачем забезпечує введення назви й відображення рекомендацій (у вигляді таблиці з оцінками подібності та ключовими метаданими).

2.2 Алгоритмічне забезпечення

Алгоритмічне забезпечення програмного модуля контентних рекомендацій спрямоване на формалізоване представлення текстових описів об’єктів каталогу Netflix та подальше обчислення їхньої попарної близькості для формування ранжованого списку рекомендацій. Обраний підхід належить до контентної фільтрації та реалізує класичний векторно-просторовий механізм зіставлення документів: кожному об’єкту (фільму/серіалу) відповідає вектор ознак, а міра подібності визначається як функція взаємного розташування векторів у просторі. Загальний алгоритмічний конвеєр включає етапи попередньої обробки тексту, векторизації описів та обчислення косинусної подібності з подальшим ранжуванням результатів.

2.2.1 Попередня обробка тексту

Попередня обробка тексту виконується з метою зниження впливу шумових компонентів, уніфікації представлення та підвищення стабільності подальшої векторизації. На цьому етапі здійснюються такі операції.

Очищення передбачає вилучення елементів, що не несуть семантичного навантаження для порівняння описів: зайвих пробілів, знаків пунктуації, службових символів, URL-адрес та інших фрагментів, які можуть спотворювати статистику термінів. У межах набору Netflix додатково доцільно усувати технічні артефакти, що виникають під час зчитування CSV (наприклад, неочікувані роздільники або зайві лапки).

Токенізація полягає у сегментації тексту на послідовність токенів (слів або термінів), що забезпечує подальшу побудову словника та підрахунок частот. Для задачі контентних рекомендацій достатньою є словесна токенізація, оскільки порівняння виконується на рівні термінів та їхніх ваг.

Нормалізація спрямована на приведення тексту до уніфікованого вигляду. Типово вона включає перетворення до нижнього регістру, видалення стоп-слів

та, за потреби, лематизацію або стемінг. Вилучення стоп-слів зменшує частку частотних, але малозмістовних термінів, що покращує дискримінативність ознак у векторному просторі.

2.2.2 Векторизація описів

Векторизація забезпечує перехід від текстового представлення описів до числових векторів, придатних для обчислення подібності. У роботі використано зважування TF-IDF (term frequency–inverse document frequency), яке підсилює внесок інформативних термінів і зменшує вплив загальних слів, що часто зустрічаються в багатьох описах.

Вага терміна tу документі dвизначається так:

$$\mathrm{tfidf}(t,d) = \mathrm{tf}(t,d) \cdot \log\left(\frac{N}{\mathrm{df}(t)}\right), \tag{2.1}$$

де $\mathrm{tf}(t,d)$— частота терміна tу документі d;

$\mathrm{df}(t)$— кількість документів колекції, що містять термін t;

N— загальна кількість документів у колекції.

Результатом етапу є матриця ознак $X \in \mathbb{R}^{N\times M}$, де N— кількість об’єктів каталогу, M— розмір словника, а кожен рядок відповідає TF-IDF-вектору окремого опису.

2.2.3 Косинусна подібність

Для порівняння векторів TF-IDF застосовується косинусна подібність, яка оцінює кут між двома векторами та є стійкою до різниці довжин текстів за умови нормалізації. Косинусна подібність для двох векторів A та Bвизначається як:

$$\mathrm{cosine}(A,B) = \frac{A\cdot B}{\|A\|\cdot\|B\|}, \tag{2.2}$$

де $A \cdot B$— скалярний добуток векторів;

$\| A \|$та $\| B \|$— їхні евклідові норми.

Значення подібності належить інтервалу $[0; 1]$ для невід'ємних TF-IDF-векторів: чим ближче значення до 1, тим вищою є схожість термінологічних профілів відповідних описів.

Опис алгоритму формування рекомендацій у межах обраного підходу включає такі кроки:

1. Формування матриці TF-IDF для очищених та нормалізованих текстових профілів усіх об'єктів каталогу.

2. Обчислення матриці косинусної подібності між усіма парами об'єктів, результатом чого є матриця $S \in \mathbb{R}^{N \times N}$.

3. Сортування результатів для заданого об'єкта-запиту за спаданням значень подібності з виключенням самого об'єкта (самоподібність).

4. Виведення Top-N рекомендацій, тобто перших Nпозицій ранжованого списку разом із відповідними значеннями подібності та метаданими, необхідними для інтерпретації результату.

Такий алгоритм є відтворюваним, інтерпретованим і придатним для реалізації у вигляді програмного модуля, оскільки всі етапи мають формалізовані процедури обробки та чітко визначені проміжні артефакти (матриця ознак і матриця подібності), що спрощує тестування та подальше розширення функціоналу.

2.3 Інформаційне забезпечення

Інформаційне забезпечення програмного модуля визначає склад, структуру та правила використання вхідних даних, необхідних для формування контентних рекомендацій. У межах роботи як джерело використовується табличний набір даних Netflix (формат CSV), що містить метадані про фільми та серіали. Ключовими для побудови рекомендацій є текстові атрибути та пов'язані з ними категоріальні характеристики, які дозволяють сформувати узагальнений

текстовий профіль об’єкта і надалі виконувати порівняння на основі векторної подібності.

2.3.1 Опис вхідних даних

Для реалізації модулю контентних рекомендацій використовуються такі поля:

— title (назва) — унікальна або майже унікальна текстова назва об’єкта каталогу. Атрибут використовується як основний ідентифікатор під час запиту користувача, а також для відображення результатів рекомендацій. У програмній реалізації можливе застосування пошуку за точним збігом та пошуку за частковим збігом (у разі наявності варіантів написання);

— description (опис) — текстова анотація, що стисло характеризує сюжет, тему або контекст контенту. Саме цей атрибут є головним носієм змістовної інформації і виступає базою для побудови векторного подання (BoW/TF-IDF) та обчислення косинусної подібності між об’єктами;

— genre (жанр/категорія) — множинний або багатозначний атрибут, що відображає тематичну належність контенту (наприклад, драма, комедія, трилер тощо). У датасеті жанрова інформація зазвичай подається як рядок із переліком категорій, розділених комами. У контентному підході жанр доцільно включати до інтегрованого текстового профілю або кодувати окремо як категоріальні ознаки, залежно від вибраної схеми векторизації;

— cast (акторський склад) — текстове поле зі списком основних виконавців ролей. Атрибут є інформативним для рекомендацій, оскільки спільні актори часто є сильним сигналом подібності для користувача. Як і жанр, акторський склад може бути включений до інтегрованого текстового поля для векторизації.

З огляду на зазначене, вхідні дані формують як мінімум один інформаційний об’єкт «Контент», який описується текстовими (title, description, cast) та категоріальними (genre) властивостями. Надалі на їхній основі будується концептуальна та логічна моделі даних.

2.3.2 Концептуальна модель даних

Концептуальна модель відображає сутності предметної області та зв’язки між ними на рівні незалежному від конкретної реалізації у вигляді таблиці. Для задачі контентних рекомендацій доцільно виділити сутність «Фільм/Серіал», що пов’язана з текстовим описом та жанровими категоріями. Узагальнену діаграму подано на рисунку 2.3.

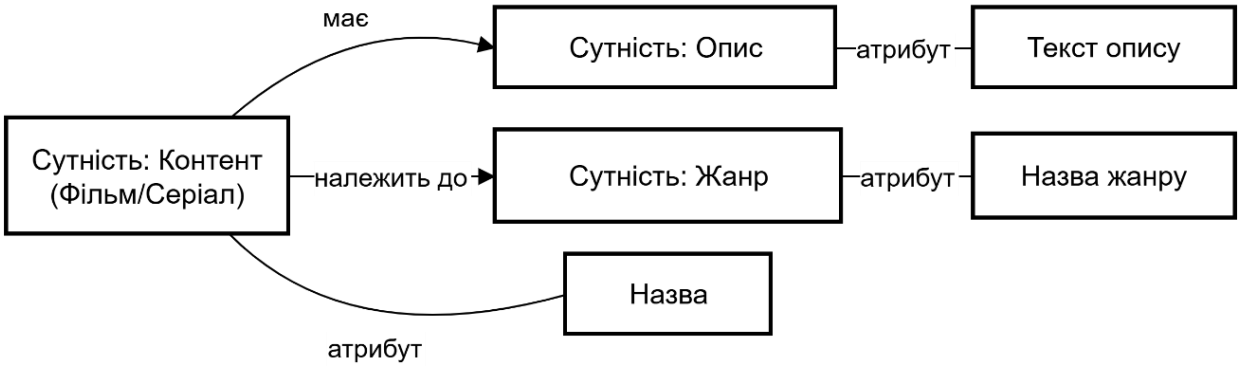

Рисунок 2.3 – Концептуальна модель даних

Як показано на рисунку 2.3, центральною сутністю виступає «Контент», для якого визначено зв’язок із сутністю «Опис» (як джерелом змістового представлення) та сутністю «Жанр» (як категоріальною ознакою). У табличному датасеті ці сутності можуть бути представлені у денормалізованому вигляді (у межах одного запису), однак концептуально вони відображають різні типи інформації: ідентифікаційну, змістову та класифікаційну.

2.3.3 Логічна модель

Логічна модель конкретизує концептуальну модель на рівні структури даних, придатної для реалізації у вигляді таблиці датасету. У межах Netflix CSV кожен рядок відповідає одному об’єкту контенту, а атрибути зберігаються у стовпцях. Для алгоритмічного контуру модуля використовується підмножина полів, достатня для формування текстового профілю й інференсу рекомендацій. Табличну структуру ключових полів наведено в таблиці 2.1.

Таблиця 2.1 – Фрагмент логічної структури вхідного датасету (ключові поля)

Поле	Тип даних (логічний)	Опис призначення	Приклад значення
title	рядок	Назва об’єкта контенту; використовується для запиту і відображення	*Stranger Things*
description	рядок	Текстовий опис сюжету/тематики; основа для векторизації	*When a young boy vanishes…*
genre	рядок (множина у рядку)	Перелік жанрів; використовується як категоріальний сигнал	*Drama, Sci-Fi & Fantasy*
cast	рядок (множина у рядку)	Перелік акторів; використовується як контентний сигнал	*Winona Ryder, David Harbour…*

У програмній реалізації логічна модель може бути доповнена технічними атрибутами (наприклад, внутрішній числовий ідентифікатор, сформований для стабільної адресації рядків), а також похідними полями (інтегроване текстове поле для векторизації). Таке розширення не змінює первинну логіку даних, але підвищує відтворюваність і надійність роботи модуля.

3 ПРОГРАМНО-ТЕХНОЛОГІЧНА РЕАЛІЗАЦІЯ МОДУЛЯ

3.1 Реалізація підготовки даних та дослідницького аналізу

На початковому етапі реалізації програмного модуля здійснено завантаження та первинний аналіз датасету *Netflix Movies and TV Shows* https://www.kaggle.com/datasets/shivamb/netflix-shows, що містить 8807 записів та 12 атрибутів. Результати виконання методу df.info() (рисунок 3.1) засвідчили, що структура набору включає 11 текстових полів типу object та один числовий атрибут release_year. Обсяг пам'яті, що використовується набором, становить близько 825 КБ, що дозволяє здійснювати обчислення без потреби в розподілених обчислювальних ресурсах.

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020	PG-13	90 min	Documentaries	As her father nears the end of his life, filmm...
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	After crossing paths at a party, a Cape Town t...
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	September 24, 2021	2021	TV-MA	1 Season	Crime TV Shows, International TV Shows, TV Act...	To protect his family from a powerful drug lor...
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	September 24, 2021	2021	TV-MA	1 Season	Docuseries, Reality TV	Feuds, flirtations and toilet talk go down amo...
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, Romantic TV Shows, TV ...	In a city of coaching centers known to train I...

Рисунок 3.1 – Структура датасету Netflix (df.info())

Аналіз повноти даних показав наявність пропусків у низці полів. Зокрема, у полі director зафіксовано 2634 відсутніх значення (29,9% від загальної кількості записів), у полі cast – 825 записів (9,4%), у полі country – 831 запис (9,4%), у полі date_added – 10 записів, а у полі rating – 4 записи. Кількісний розподіл пропусків наведено на рис. 3.2. Найбільш проблемним атрибутом виявився director, однак його вилучення призвело б до значної втрати інформації, тому було прийнято рішення виконати заміну пропусків значенням *Unknown*.

З метою підвищення якості подальшої текстової обробки було вилучено записи з відсутніми значеннями у полях cast та country, оскільки ці атрибути є суттєвими для формування контентного профілю. Після очищення здійснено переіндексацію таблиці для забезпечення цілісності структури даних. Такий

підхід дозволив зменшити кількість неповних записів та підготувати дані до етапу векторизації.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8807 entries, 0 to 8806
Data columns (total 12 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   show_id       8807 non-null   object
 1   type          8807 non-null   object
 2   title         8807 non-null   object
 3   director      6173 non-null   object
 4   cast          7982 non-null   object
 5   country       7976 non-null   object
 6   date_added    8797 non-null   object
 7   release_year  8807 non-null   int64
 8   rating        8803 non-null   object
 9   duration      8804 non-null   object
 10  listed_in     8807 non-null   object
 11  description   8807 non-null   object
dtypes: int64(1), object(11)
memory usage: 825.8+ KB
```

Рисунок 3.2 – Кількість пропусків у полях датасету

Дослідницький аналіз структури каталогу показав, що більшість контенту становлять фільми. Згідно з діаграмою розподілу типів (рисунок 3.3), частка фільмів становить 72,3%, тоді як частка телешоу – 27,7%. Така пропорція свідчить про домінування повнометражного контенту в каталозі та впливає на статистичну структуру рекомендаційної моделі.

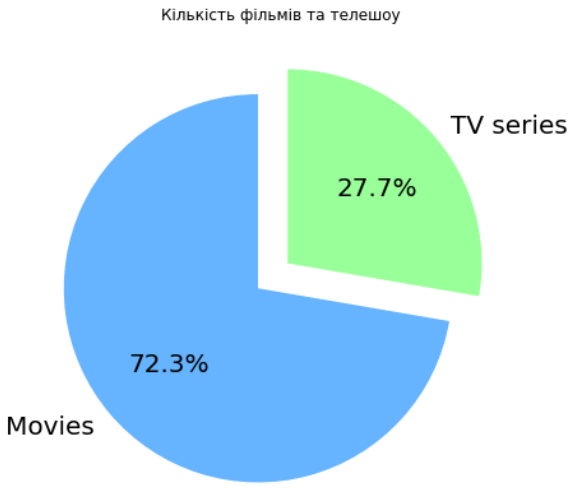

Рисунок 3.3 – Співвідношення фільмів та телешоу

Аналіз топ-10 режисерів для фільмів (рисунок 3.4) демонструє концентрацію контенту у невеликої кількості авторів, що створює потенційні семантичні кластери в текстових описах. Аналогічний аналіз для телешоу (рисунок 3.5) показує відмінну структуру розподілу, що пояснюється особливостями серіального виробництва.

Рисунок 3.4 – Топ-10 режисерів фільмів на Netflix

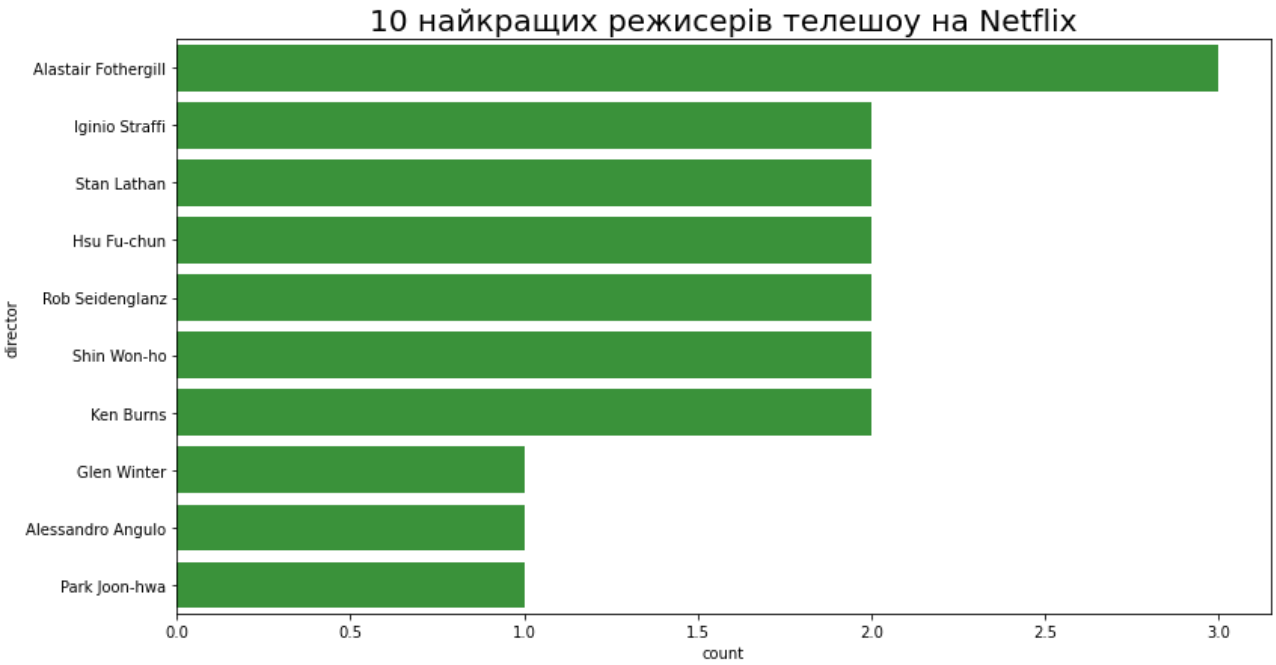

Рисунок 3.5 – Топ-10 режисерів телешоу на Netflix

Розподіл країн-виробників (рисунок 3.6) засвідчив домінування США у виробництві контенту, що відповідає глобальній структурі стримінгового ринку. Водночас значну частку займають Канада, Великобританія, Індія та Японія, що свідчить про мультикультурний характер каталогу.

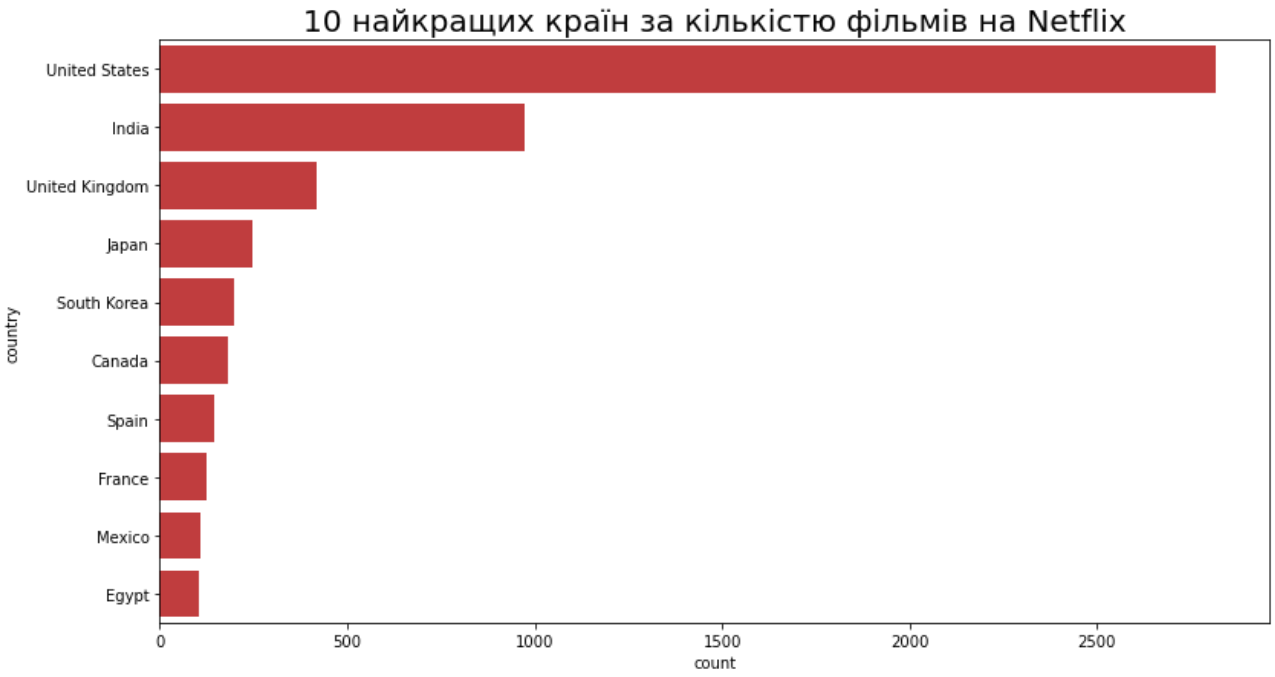

Рисунок 3.6 – Топ-10 країн-виробників контенту Netflix

Таким чином, проведений аналіз дозволив не лише виявити структурні особливості набору даних, але й обґрунтувати подальші алгоритмічні рішення щодо формування інтегрованого текстового профілю та вибору методу подібності.

3.2 Реалізація алгоритмічного ядра контентних рекомендацій

Алгоритмічне ядро модуля базується на формуванні контентного профілю кожного об’єкта та обчисленні косинусної подібності між текстовими векторами. Для цього було створено інтегроване поле overall_infos, що поєднує значення атрибутів type, title, director, cast, description та country. Така конкатенація дозволяє сформувати розширене семантичне представлення кожного об’єкта.

У зв’язку з неконсистентністю початкового поля show_id було створено новий числовий ідентифікатор id, який формується як послідовність натуральних чисел. Це забезпечило стабільну адресацію об’єктів при зверненні до матриці подібності та виключило потенційні помилки індексації.

На етапі попередньої обробки тексту реалізовано приведення символів до нижнього регістру, видалення пунктуації та шумових символів, токенізацію тексту, а також вилучення стоп-слів за допомогою бібліотеки NLTK. Порівняння тексту до та після обробки (рисунок 3.7) демонструє зменшення обсягу лексичних шумів та підвищення концентрації семантично значущих термінів.

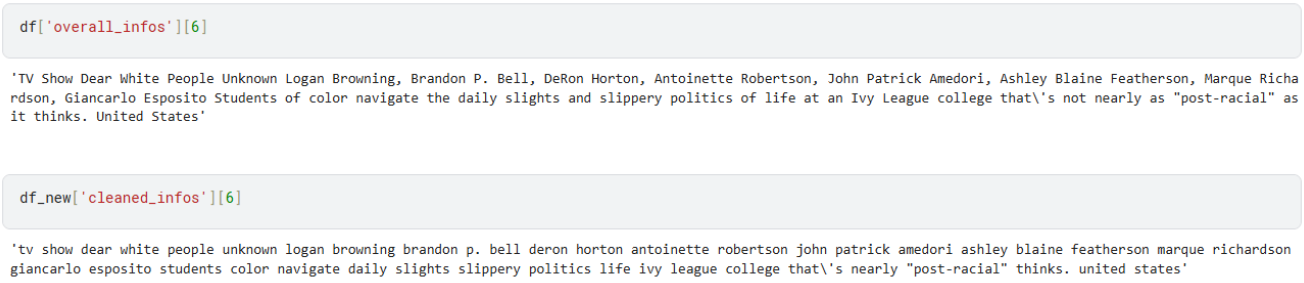

Рисунок 3.7 – Порівняння тексту до та після передобробки

Після нормалізації тексту застосовано CountVectorizer, у результаті чого сформовано матрицю ознак розмірності N × M, де N – кількість об’єктів після очищення, а M – кількість унікальних термінів словника. Кожен рядок матриці відповідає одному об’єкту каталогу, а кожен стовпець – окремому терміну.

На основі отриманої матриці ознак обчислено матрицю косинусної подібності розмірності N × N (рисунок 3.8). Матриця є симетричною, а значення на головній діагоналі дорівнюють 1. Позадіагональні елементи знаходяться в інтервалі [0; 1], що відображає ступінь семантичної близькості між відповідними об’єктами.

```
array([[1.        , 0.05856516, 0.        , ..., 0.        , 0.        ,
        0.        ],
       [0.05856516, 1.        , 0.        , ..., 0.02678358, 0.04938648,
        0.07919455],
       [0.        , 0.        , 1.        , ..., 0.10006256, 0.09225312,
        0.04931137],
       ...,
       [0.        , 0.02678358, 0.10006256, ..., 1.        , 0.10846523,
        0.02898855],
       [0.        , 0.04938648, 0.09225312, ..., 0.10846523, 1.        ,
        0.02672612],
       [0.        , 0.07919455, 0.04931137, ..., 0.02898855, 0.02672612,
        1.        ]])
```

Рисунок 3.8 – Фрагмент матриці косинусної подібності

Алгоритм рекомендацій передбачає визначення індексу об’єкта-запиту, формування списку пар (індекс, коефіцієнт подібності), сортування за спаданням та відбір перших N результатів, за винятком самого об’єкта. Приклад ранжування для запиту *Teenage Mutant Ninja Turtles* (рисунок 3.9) демонструє, що максимальний коефіцієнт подібності серед рекомендованих об’єктів становить 0,333, а мінімальний у межах Top-5 – близько 0,198.

```
show_id          0
type             0
title            0
director      2634
cast           825
country        831
date_added      10
release_year     0
rating           4
duration         3
listed_in        0
description      0
dtype: int64
```

Рисунок 3.9 – Приклад ранжування за коефіцієнтом подібності

Отримані значення подібності свідчать про коректну роботу моделі, оскільки найбільш релевантними виявляються об’єкти зі схожою тематикою (наприклад, інші фільми серії Ninja Turtles або контент із подібними ключовими словами).

3.3 Тестування модуля та прототип користувацького інтерфейсу

Тестування програмного модуля виконано шляхом серії контрольних запусків функції recommend_ui(title, top_k) у середовищі Notebook. Для кожного запиту користувача формувався список Top-5 рекомендацій, ранжований за значенням косинусної подібності між векторним поданням інтегрованого текстового профілю об’єкта та профілями інших елементів каталогу. Результат подавався у табличному вигляді з фіксацією позиції в рейтингу, числового значення подібності та базових метаданих (назва, тип, режисер, країна).

Для запиту “Teenage Mutant Ninja Turtles” сформовано перелік п’яти найближчих об’єктів (рис. 3.10). Максимальне значення подібності для першої

рекомендації становить 0,333333, що відповідає об’єкту *Teenage Mutant Ninja Turtles II: The Secret of…*. Наступні позиції мають значення 0,253629 та 0,251312, а мінімальні у межах Top-5 — 0,198680 (дві позиції), що вказує на зменшення семантичної близькості зі зниженням рангу.

```
# ======= "Екран" користувача =======
TITLE = "Teenage Mutant Ninja Turtles"
TOP_K = 5

recommend_ui(TITLE, TOP_K)
```

```
Рекомендації для: Teenage Mutant Ninja Turtles (Top-5)
```

	Місце	Подібність (cosine)	title	type	director	country
0	1	0.333333	Teenage Mutant Ninja Turtles II: The Secret of...	Movie	Michael Pressman	United States, Hong Kong
1	2	0.253629	Kevin James: Sweat the Small Stuff	Movie	Paul Miller	United States
2	3	0.251312	Kevin James: Never Don't Give Up	Movie	Andy Fickman	United States
3	4	0.198680	Kevin Hart: Irresponsible	Movie	Leslie Small	United States
4	5	0.198680	Ninja Turtles: The Next Mutation	TV Show	Unknown	Canada, United States

Рисунок 3.10 – Результати рекомендацій для запиту «Teenage Mutant Ninja Turtles» (Top-5)

Для запиту “LEGO Ninjago” отримано найвищі значення косинусної подібності серед розглянутих прикладів (рисунок 3.11). Максимальна подібність першої рекомендації становить 0,466004 і відповідає *LEGO Ninjago: Masters of Spinjitzu*, що є очікуваним результатом через належність до спільної франшизи. Подальші рекомендації мають значення 0,304408, 0,253546, 0,230022 та 0,204124, тобто спостерігається послідовне спадання релевантності у межах сформованого рейтингу.

```
# ======= "Екран" користувача =======
TITLE = "LEGO Ninjago"
TOP_K = 5

recommend_ui(TITLE, TOP_K)
```

Рекомендації для: LEGO Ninjago (Top-5)

	Місце	Подібність (cosine)	title	type	director	country
0	1	0.466004	LEGO Ninjago: Masters of Spinjitzu	TV Show	Unknown	Denmark, Singapore, Canada, United States
1	2	0.304408	Tobot	TV Show	Unknown	Canada
2	3	0.253546	The Deep	TV Show	Unknown	Canada, Australia
3	4	0.230022	Kong: King of the Apes	TV Show	Unknown	United States, Japan, Canada
4	5	0.204124	Hatchimals | Adventures in Hatchtopia	TV Show	Unknown	United States

Рисунок 3.11 – Результати рекомендацій для запиту «LEGO Ninjago» (Top-5)

У випадку запиту “Five Elements Ninjas” (рисунок 3.12) максимальна подібність першої позиції становить 0,252755 (об’єкт *The Five Venoms*). Наступні значення подібності дорівнюють 0,234701 та 0,211830, тоді як мінімальні у Top-5 — 0,199852 і 0,198593. Отримані оцінки відображають помірний рівень лексико-семантичної близькості, що є типовим для випадків, коли подібність формується не за прямою франшизною спорідненістю, а переважно за тематичними та жанровими маркерами в описах.

```
TITLE = "Five Elements Ninjas"
TOP_K = 5

recommend_ui(TITLE, TOP_K)
```

Рекомендації для: Five Elements Ninjas (Top-5)

	Місце	Подібність (cosine)	title	type	director	country
0	1	0.252755	The Five Venoms	Movie	Cheh Chang	Hong Kong
1	2	0.234701	Weeds on Fire	Movie	Chi Fat Chan	Hong Kong
2	3	0.211830	ОСТВ	TV Show	Unknown	Hong Kong
3	4	0.199852	Flying Guillotine 2	Movie	Kang Cheng, Shan Hua	Hong Kong
4	5	0.198593	The Grandmaster	Movie	Wong Kar Wai	Hong Kong, China

Рисунок 3.12 – Результати рекомендацій для запиту «Five Elements Ninjas» (Top-5)

Для запиту “Power Rangers Ninja Storm” (рисунок 3.13) максимальна подібність першої рекомендації становить 0,297775 (*Power Rangers Dino Thunder*). Значення для інших позицій знаходяться у відносно вузькому інтервалі: 0,288800, 0,284901, 0,278271, 0,272166. Така компактність інтервалу свідчить про високу однорідність рекомендаційного набору та стабільний рівень подібності між об’єктами однієї серії, що корелює з повторюваними лексичними структурами у назвах та описах сезонів.

```
TITLE = "Power Rangers Ninja Storm"
TOP_K = 5

recommend_ui(TITLE, TOP_K)
```

Рекомендації для: Power Rangers Ninja Storm (Top-5)

	Місце	Подібність (cosine)	title	type	director	country
0	1	0.297775	Power Rangers Dino Thunder	TV Show	Unknown	United States
1	2	0.288800	Power Rangers Samurai	TV Show	Unknown	United States
2	3	0.284901	Power Rangers Ninja Steel	TV Show	Unknown	United States
3	4	0.278271	Mighty Morphin Power Rangers	TV Show	Unknown	United States, Japan
4	5	0.272166	Power Rangers Dino Fury	TV Show	Unknown	United States

Рисунок 3.13 – Результати рекомендацій для запиту «Power Rangers Ninja Storm» (Top-5)

Для запиту “Power Rangers Dino Thunder” (рисунок 3.14) отримано ще вищий рівень внутрішньосерійної близькості: максимальна подібність першої рекомендації становить 0,332923 (*Power Rangers Dino Charge*). Наступні позиції мають значення 0,324176, 0,318696, 0,307365, 0,305281, що демонструє збереження подібності на рівні понад 0,30 для всього переліку Top-5. Це підтверджує, що за наявності стійких франшизних маркерів контентний підхід на основі косинусної подібності формує тематично узгоджені та інтерпретовані рекомендації.

```
TITLE = "Power Rangers Dino Thunder"
TOP_K = 5

recommend_ui(TITLE, TOP_K)
```

```
Рекомендації для: Power Rangers Dino Thunder (Top-5)
```

	Місце	Подібність (cosine)	title	type	director	country
0	1	0.332923	Power Rangers Dino Charge	TV Show	Unknown	United States
1	2	0.324176	Power Rangers Dino Fury	TV Show	Unknown	United States
2	3	0.318696	Power Rangers RPM	TV Show	Unknown	United States
3	4	0.307365	Power Rangers Zeo	TV Show	Unknown	United States, France, Japan
4	5	0.305281	Mighty Morphin Power Rangers	TV Show	Unknown	United States, Japan

Рисунок 3.14 – Результати рекомендацій для запиту «Power Rangers Dino Thunder» (Top-5)

Аналіз отриманих рекомендацій демонструє тематичну згрупованість результатів: система коректно пропонує інші серії франшизи Power Rangers або об’єкти з ключовим словом “Ninja”. Це підтверджує, що модель ефективно використовує лексичну інформацію інтегрованого текстового профілю.

Представлений прототип інтерфейсу є мінімалістичним, однак повністю функціональним. Його архітектура дозволяє відокремити обчислювальну частину від рівня представлення результатів, що створює передумови для подальшої інтеграції у вебзастосунок або десктопне середовище.

Таким чином, проведене тестування підтвердило коректність реалізації алгоритму контентних рекомендацій та його здатність генерувати семантично узгоджені списки об’єктів на основі текстової подібності.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи поставлену мету досягнуто, а визначені завдання реалізовано в повному обсязі. Отримані результати можна узагальнити відповідно до сформульованих завдань.

1. Досліджено специфіку функціонування стрімінгових платформ, визначено ключові характеристики каталогу (масштабність, неоднорідність метаданих, динаміка споживання). Проведено порівняльний аналіз контентних, колаборативних і гібридних рекомендаційних систем. Обґрунтовано доцільність використання контентної фільтрації в умовах відсутності або обмеженості поведінкових даних, а також з позицій відтворюваності та інтерпретованості результатів.

2. Виконано аналіз структури датасету (8807 записів, 12 атрибутів), виявлено пропуски в окремих полях (зокрема director, cast, country). Сформовано правила обробки пропусків: заміна технічно другорядних атрибутів маркерними значеннями та вилучення записів із критично відсутніми полями. Визначено перелік релевантних атрибутів (type, title, director, cast, description, country) для формування інтегрованого текстового профілю об’єкта.

3. Запроваджено процедури очищення тексту, приведення до нижнього регістру, видалення пунктуації та стоп-слів, токенізації та нормалізації. Реалізовано механізм конкатенації декількох атрибутів у поле overall_infos, що забезпечило розширене семантичне представлення кожного об’єкта каталогу. Підхід дозволив перейти від розрізнених метаданих до уніфікованого текстового профілю, придатного для векторизації.

4. Застосовано CountVectorizer для формування векторно-просторового представлення описів. Сформовано матрицю ознак розмірності $N \times M$, де N — кількість об’єктів каталогу, M — кількість унікальних термінів словника. Побудована матриця стала основою для подальшого обчислення подібності між об’єктами та забезпечила формалізоване числове представлення контенту.

5. На основі матриці ознак побудовано симетричну матрицю косинусної подібності розмірності N × N. Реалізовано алгоритм пошуку індексу об’єкта-запиту, сортування за спаданням коефіцієнтів подібності та відбору Top-N рекомендацій із виключенням самоподібності. Отримані значення подібності (наприклад, до 0,466 для споріднених об’єктів однієї франшизи) підтвердили коректність реалізації алгоритму та його здатність формувати тематично узгоджені рекомендації.

6. Виконано серію контрольних запусків функції рекомендацій для різних запитів (зокрема об’єктів франшизного типу). Результати продемонстрували логічне ранжування та монотонне зменшення коефіцієнта подібності в межах Top-5. Реалізовано прототип інтерфейсу у середовищі Notebook, який забезпечує введення назви об’єкта та виведення рекомендацій із числовими оцінками подібності та ключовими метаданими, що підвищує інтерпретованість результатів.

Таким чином, розроблений програмний модуль контентних рекомендацій є функціонально завершеним, відтворюваним та інтерпретованим рішенням для задачі добору подібного контенту в каталозі стрімінгового сервісу. Отримані результати створюють підґрунтя для подальшого розширення системи шляхом переходу до TF-IDF-векторизації, використання семантичних подань або гібридизації з поведінковими сигналами користувачів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Manning C. D., Raghavan P., Schütze H. *Introduction to Information Retrieval*. Cambridge: Cambridge University Press, 2008. 482 p. URL: https://www.cambridge.org/highereducation/books/introduction-to-information-retrieval/669D108D20F556C5C30957D63B5AB65C (дата звернення: 24.02.2026).

2. Salton G., Buckley C. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*. 1988. Vol. 24(5). P. 513–523. DOI: https://doi.org/10.1016/0306-4573(88)90021-0 (дата звернення: 24.02.2026).

3. Baeza-Yates R., Ribeiro-Neto B. *Modern Information Retrieval: The Concepts and Technology behind Search*. 2nd ed. Harlow: Addison-Wesley, 2011. 944 p. URL: https://books.google.com/books/about/Modern_Information_Retrieval.html?id=HbyAAAAACAAJ (дата звернення: 24.02.2026).

4. Croft W. B., Metzler D., Strohman T. *Search Engines: Information Retrieval in Practice*. Boston: Addison-Wesley, 2010. 520 p. URL: https://www.worldcat.org/title/search-engines-information-retrieval-in-practice/oclc/317919336 (дата звернення: 24.02.2026).

5. Lewis D. D. Feature selection and feature extraction for text categorization. In: *Proceedings of the Workshop on Speech and Natural Language (HLT'92)*. 1992. P. 212–217. DOI: https://doi.org/10.3115/1075527.1075574. URL: https://aclanthology.org/H92-1041.pdf (дата звернення: 24.02.2026).

6. Deerwester S., Dumais S. T., Furnas G. W., Landauer T. K., Harshman R. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*. 1990. Vol. 41(6). P. 391–407. DOI: https://doi.org/10.1002/(SICI)1097-4571(199009)41:6<391::AID-ASI1>3.0.CO;2-9. URL: https://asistdl.onlinelibrary.wiley.com/doi/abs/10.1002/%28SICI%291097-4571%28199009%2941%3A6%3C391%3A%3AAID-ASI1%3E3.0.CO%3B2-9 (дата звернення: 24.02.2026).

7. Robertson S. E., Spärck Jones K. Relevance weighting of search terms. *Journal of the American Society for Information Science*. 1976. Vol. 27(3). P. 129–146.

DOI: https://doi.org/10.1002/asi.4630270302. URL: https://asistdl.onlinelibrary.wiley.com/doi/abs/10.1002/asi.4630270302 (дата звернення: 24.02.2026).

8. Jones K. S. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*. 1972. Vol. 28(1). P. 11–21. DOI: https://doi.org/10.1108/eb026526. URL: https://www.staff.city.ac.uk/~sbrp622/idfpapers/ksj_orig.pdf (дата звернення: 24.02.2026).

9. TfidfVectorizer — scikit-learn documentation. URL: https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html (дата звернення: 24.02.2026).

10. cosine_similarity — scikit-learn documentation. URL: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.pairwise.cosine_similarity.html (дата звернення: 24.02.2026).

11. Muhammad Ayman. Recommendation System Using Cosine Similarity. *Kaggle Notebook*. URL: https://www.kaggle.com/code/muhammadayman/recommendation-system-using-cosine-similarity/notebook (дата звернення: 24.02.2026).

12. Jaccard P. Étude comparative de la distribution florale dans une portion des Alpes et du Jura. *Bulletin de la Société Vaudoise des Sciences Naturelles*. 1901. Vol. 37(142). P. 547–579. DOI: https://doi.org/10.5169/seals-266450. URL: https://www.e-periodica.ch/digbib/view?pid=bsv-002%3A1901%3A37%3A%3A579 (дата звернення: 24.02.2026).

13. Levandowsky M., Winter D. Distance between sets. *Nature*. 1971. Vol. 234. P. 34–35. DOI: https://doi.org/10.1038/234034a0. URL: https://www.nature.com/articles/234034a0 (дата звернення: 24.02.2026).

14. Aggarwal C. C., Hinneburg A., Keim D. A. On the surprising behavior of distance metrics in high dimensional space. In: *Database Theory — ICDT 2001*. Berlin: Springer, 2001. P. 420–434. DOI: https://doi.org/10.1007/3-540-44503-X_27. URL:

https://link.springer.com/chapter/10.1007/3-540-44503-X_27 (дата звернення: 24.02.2026).

15. Netflix Movies and TV Shows dataset. *Kaggle*. URL: https://www.kaggle.com/datasets/shivamb/netflix-shows (дата звернення: 24.02.2026).

16. Han J., Kamber M., Pei J. *Data Mining: Concepts and Techniques*. 3rd ed. Waltham: Morgan Kaufmann, 2011. 744 p. URL: https://www.sciencedirect.com/book/9780123814791/data-mining-concepts-and-techniques (дата звернення: 24.02.2026).

17. Russell S., Norvig P. *Artificial Intelligence: A Modern Approach*. 4th ed. Boston: Pearson, 2020. 1136 p. URL: https://aima.cs.berkeley.edu/ (дата звернення: 24.02.2026).

18. Ricci F., Rokach L., Shapira B. (eds.) *Recommender Systems Handbook*. 2nd ed. Boston: Springer, 2015. 1003 p. URL: https://link.springer.com/book/10.1007/978-1-4899-7637-6 (дата звернення: 24.02.2026).

19. Adomavicius G., Tuzhilin A. Toward the next generation of recommender systems: a survey. *IEEE Transactions on Knowledge and Data Engineering*. 2005. Vol. 17(6). P. 734–749. DOI: https://doi.org/10.1109/TKDE.2005.99. URL: https://dl.acm.org/doi/10.1109/TKDE.2005.99 (дата звернення: 24.02.2026).

20. Lops P., de Gemmis M., Semeraro G. Content-based recommender systems: state of the art and trends. In: Ricci F., Rokach L., Shapira B. (eds.) *Recommender Systems Handbook*. Springer, 2011. P. 73–105. URL: https://link.springer.com/book/10.1007/978-0-387-85820-3 (дата звернення: 24.02.2026).

21. Sarwar B., Karypis G., Konstan J., Riedl J. Item-based collaborative filtering recommendation algorithms. In: *Proceedings of WWW 2001*. 2001. P. 285–295. DOI: https://doi.org/10.1145/371920.372071. URL: https://dl.acm.org/doi/10.1145/371920.372071 (дата звернення: 24.02.2026).

22. Koren Y., Bell R., Volinsky C. Matrix factorization techniques for recommender systems. *Computer*. 2009. Vol. 42(8). P. 30–37. DOI: https://doi.org/10.1109/MC.2009.263. URL: https://dl.acm.org/doi/10.1109/MC.2009.263 (дата звернення: 24.02.2026).

23. Gomez-Uribe C. A., Hunt N. The Netflix recommender system: Algorithms, business value, and innovation. *ACM Transactions on Management Information Systems*. 2015. Vol. 6(4). Article 13. DOI: https://doi.org/10.1145/2843948. URL: https://dl.acm.org/doi/10.1145/2843948 (дата звернення: 24.02.2026).

24. Mikolov T., Chen K., Corrado G., Dean J. Efficient estimation of word representations in vector space. *arXiv preprint*. 2013. arXiv:1301.3781. URL: https://arxiv.org/abs/1301.3781 (дата звернення: 24.02.2026).

25. Pennington J., Socher R., Manning C. D. GloVe: Global vectors for word representation. In: *Proceedings of EMNLP 2014*. 2014. P. 1532–1543. DOI: https://doi.org/10.3115/v1/D14-1162. URL: https://aclanthology.org/D14-1162/ (дата звернення: 24.02.2026).

26. Devlin J., Chang M.-W., Lee K., Toutanova K. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint*. 2018. arXiv:1810.04805. URL: https://arxiv.org/abs/1810.04805 (дата звернення: 24.02.2026).

27. Vaswani A., Shazeer N., Parmar N., et al. Attention is all you need. In: *Advances in Neural Information Processing Systems (NeurIPS 2017)*. 2017. P. 5998–6008. URL: https://arxiv.org/abs/1706.03762 (дата звернення: 24.02.2026).

28. Ramos J. Using TF-IDF to determine word relevance in document queries. 2003. URL: https://www.semanticscholar.org/paper/Using-TF-IDF-to-Determine-Word-Relevance-in-Queries-Ramos/b3bf6373ff41a115197cb5b30e57830c16130c2c (дата звернення: 24.02.2026).

29. Manning C. D., Schütze H. *Foundations of Statistical Natural Language Processing*. Cambridge, MA: MIT Press, 1999. 680 p. URL:

https://mitpress.mit.edu/9780262303798/foundations-of-statistical-natural-language-processing/ (дата звернення: 24.02.2026).

30. Jurafsky D., Martin J. H. *Speech and Language Processing*. 3rd ed. draft. Stanford University, 2023. URL: https://web.stanford.edu/~jurafsky/slp3/ (дата звернення: 24.02.2026).

31. І. Середа. Розробка контентної рекомендаційної системи для стрімінгових сервісів на основі методів обробки природної мови. матеріали XI міжнароднох студентської наукової конференції Сучасні Перспективи та Перспективні Напрямки Розвитку Науки. м. Запоріжжя, Україна, 6 березня 2026 рік. – С. 73-75.

32. Комар М. П., Саченко А. О., Васильків Н. М., Гладій Г. М., Коваль В. С., Ліп’яніна-Гончаренко Х. В. Методичні рекомендації до виконання кваліфікаційної роботи з освітньо-професійної програми «Комп’ютерні науки» спеціальності 122 «Комп’ютерні науки» за першим (бакалаврським) рівнем вищої освіти. Тернопіль: ЗУНУ, 2024. 52 с.

Додаток А

Програмний код

```
import re
import numpy as np
import pandas as pd

from sklearn.feature_extraction.text import CountVectorizer
from sklearn.metrics.pairwise import cosine_similarity

# --- 1) Завантаження даних ---
DATA_PATH = "../input/netflix-shows/netflix_titles.csv"
df = pd.read_csv(DATA_PATH)

# --- 2) Мінімальне очищення даних (для стабільної побудови профілю) ---
# Для рекомендацій використовуємо cast + country (як у вашому ноутбуці) ->
NaN видаляємо.
df = df.dropna(subset=["cast", "country"]).copy()
df["director"] = df["director"].fillna("Unknown")
df = df.reset_index(drop=True)

# --- 3) Формування інтегрованої текстової ознаки ---
# Об'єднуємо ключові поля в один текстовий "профіль" об'єкта.
df["overall_infos"] = (
    df["type"].astype(str) + " " +
    df["title"].astype(str) + " " +
    df["director"].astype(str) + " " +
    df["cast"].astype(str) + " " +
    df["description"].astype(str) + " " +
    df["country"].astype(str)
)

# --- 4) Попередня обробка тексту (компактно, без NLTK) ---
def normalize_text(s: str) -> str:
    s = s.lower()
    s = re.sub(r"http\S+|www\.\S+", " ", s)      # прибрати посилання
    s = re.sub(r"[^a-z\s]", " ", s)              # лишити латиницю та пробіли
    s = re.sub(r"\s+", " ", s).strip()           # нормалізація пробілів
    return s

df["cleaned_infos"] = df["overall_infos"].astype(str).map(normalize_text)

# --- 5) Векторизація + Cosine Similarity ---
vectorizer = CountVectorizer(stop_words="english")
X = vectorizer.fit_transform(df["cleaned_infos"])
```

```
cos_sim = cosine_similarity(X)

# --- 6) Функція рекомендацій ---
def recommend(title: str, top_k: int = 5) -> pd.DataFrame:
    """
    Повертає Top-k рекомендацій за косинусною подібністю для заданої назви.
    Якщо точної назви немає — повертає варіанти для уточнення.
    """
    if not isinstance(title, str) or not title.strip():
        return pd.DataFrame({"message": ["Вкажіть назву фільму/серіалу."]})

    title_q = title.strip().lower()
    exact = df.index[df["title"].str.lower() == title_q]

    if len(exact) == 0:
        cand = df.loc[df["title"].str.contains(title.strip(), case=False, na=False),
"title"].head(10)
        if cand.empty:
            return pd.DataFrame({"message": ["Збігів не знайдено. Спробуйте іншу
назву."]})
        return pd.DataFrame({"possible_titles": cand.tolist()})

    idx = int(exact[0])

    scores = list(enumerate(cos_sim[idx]))
    scores = sorted(scores, key=lambda x: x[1], reverse=True)
    scores = [s for s in scores if s[0] != idx][:top_k]

    rec_idx = [i for i, _ in scores]
    rec_sc = [float(sc) for _, sc in scores]

    out = df.loc[rec_idx, ["title", "type", "director", "country"]].copy()
    out.insert(0, "rank", range(1, len(out) + 1))
    out.insert(1, "cosine_similarity", rec_sc)
    return out.reset_index(drop=True)

# --- 7) Приклад використання ---
print(recommend("Teenage Mutant Ninja Turtles", top_k=5))
```

Додаток Б

Апробація отриманих результатів

МОЛОДІЖНА НАУКОВА ЛІГА

МАТЕРІАЛИ XI МІЖНАРОДНОЇ СТУДЕНТСЬКОЇ НАУКОВОЇ

КОНФЕРЕНЦІЇ

СУЧАСНІ АСПЕКТИ ТА ПЕРСПЕКТИВНІ НАПРЯМКИ РОЗВИТКУ НАУКИ

м. Запоріжжя, Україна
6 березня 2026 рік

Вінниця, Україна
«UKRLOGOS Group»
2026

УДК 082:001
С 89

Голова оргкомітету: Карєнок І.О.
Верстка: Білоус Т.В.
Дизайн: Бондаренко І.В.

Рекомендовано до видання Вченою Радою Інституту науково-технічної інтеграції та співпраці. Протокол № 8 від 05.03.2026 року.

Конференцію зареєстровано Державною науковою установою «УкрІНТЕІ» в базі даних науково-технічних заходів України та бюлетені «План проведення наукових, науково-технічних заходів в Україні» (Посвідчення № 109 від 26.01.2026).

С 89 **Сучасні аспекти та перспективні напрямки розвитку науки**: матеріали XI Міжнародної студентської наукової конференції, м. Запоріжжя, 6 березня, 2026 рік / ГО «Молодіжна наукова ліга». — Вінниця: ТОВ «УКРЛОГОС Груп», 2026. — 140 с.

ISBN 978-617-8582-25-8
DOI 10.62732/liga-inter-06.03.2026

Викладено матеріали учасників XI Міжнародної мультидисциплінарної студентської наукової конференції «Сучасні аспекти та перспективні напрямки розвитку науки», яка відбулася 6 березня 2026 року у місті Запоріжжя, Україна.

УДК 082:001

ISBN 978-617-8582-25-8

СЕКЦІЯ 8.
КОМП'ЮТЕРНА ТА ПРОГРАМНА ІНЖЕНЕРІЯ

СЕКЦІЯ 9.
ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ТА СИСТЕМИ

СЕКЦІЯ 10.
ФІЗИКО-МАТЕМАТИЧНІ НАУКИ

СЕКЦІЯ 11.
ФІЛОЛОГІЯ ТА ЖУРНАЛІСТИКА

Іван Середа, здобувач вищої освіти факультету комп'ютерних інформаційних технологій
Західноукраїнський національний університет, Україна

Науковий керівник: Христина Ліп'яніна-Гончаренко, д-р.техн.наук, доцент, доцент кафедри інформаційно-обчислювальних систем і управління
Західноукраїнський національний університет, Україна

РОЗРОБКА КОНТЕНТНОЇ РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ ДЛЯ СТРІМІНГОВИХ СЕРВІСІВ НА ОСНОВІ МЕТОДІВ ОБРОБКИ ПРИРОДНОЇ МОВИ

Стрімінгові платформи сьогодні є одними з найпопулярніших інформаційних систем, що забезпечують доступ до величезних каталогів аудіовізуального контенту. Масштаб таких бібліотек зумовлює явище «інформаційного перевантаження», коли користувач не здатний ефективно обрати фільм чи серіал серед тисяч альтернатив. Саме тому рекомендаційні модулі стають критично важливим інструментом підтримки прийняття рішень, виконуючи функції фільтрації та персоналізованого ранжування контенту [1].

Метою даної роботи є розробка та тестування алгоритмічного і програмного забезпечення модуля контентних рекомендацій на основі аналізу текстових метаданих фільмів та телешоу.

На відміну від колаборативної фільтрації, яка вимагає великих масивів історичних даних про поведінку користувачів та страждає від проблеми «холодного старту», обраний контентний підхід (content-based filtering) базується виключно на атрибутах самих об'єктів. Джерелом даних для дослідження слугував відкритий набір «Netflix Movies and TV Shows» (8807 записів).

Під час розробки алгоритмічного ядра модуля було реалізовано процес формування єдиного інтегрованого текстового профілю для кожного фільму чи серіалу. Для цього здійснювалась конкатенація ключових атрибутів: типу контенту, назви, режисера, акторського складу, опису сюжету та країни-виробника (формування поля *overall_infos*). Подальша попередня обробка тексту (NLP) включала приведення до нижнього регістру, видалення пунктуації, токенізацію та вилучення стоп-слів за допомогою інструментів бібліотеки NLTK [2].

Векторизація очищених текстових профілів здійснювалася з використанням підходу TF-IDF (Term Frequency – Inverse Document Frequency), що дозволило перетворити тексти на числові вектори з урахуванням статистичної значущості кожного терміна в межах всього каталогу.

Для оцінки семантичної близькості між векторами контенту було застосовано метрику косинусної подібності. Ця метрика оцінює кут між двома векторами у багатовимірному просторі ознак за формулою (1):

$$cosine(A, B) = (A \cdot B) / (||A|| \cdot ||B||), \quad (1)$$

де: A та B – це TF-IDF вектори двох різних текстових описів контенту; (A · B) – скалярний добуток векторів; ||A|| та ||B|| – їхні евклідові норми. Значення подібності належить інтервалу від 0 до 1, де значення близькі до 1 свідчать про високу семантичну спорідненість об'єктів.

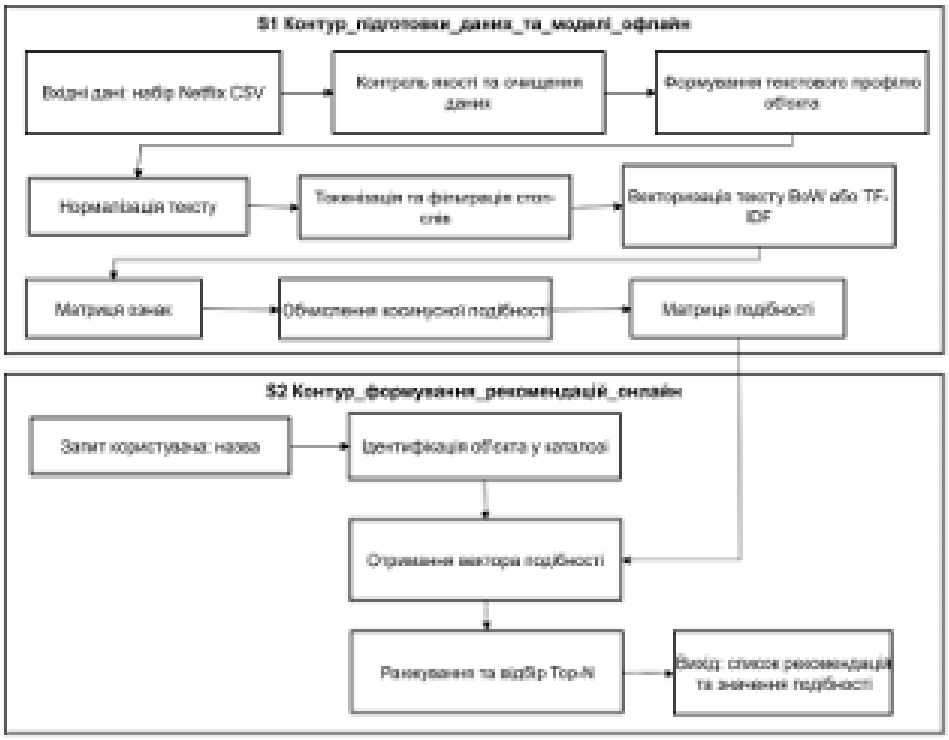

Рис. 1. **Функціональна схема модуля контентних рекомендацій**

Програмна реалізація модуля була успішно протестована на серії контрольних запитів. За введеною назвою система ідентифікує об'єкт, звертається до попередньо обчисленої матриці косинусної подібності, виконує сортування за спаданням значень та формує список Top-5 рекомендацій.

Результати роботи алгоритму для пошукового запиту «LEGO Ninjago» представлені у таблиці (табл. 1).

Таблиця 1

Результати формування рекомендацій для запиту «LEGO Ninjago»

Ранг	Назва рекомендованого контенту	Тип	Значення подібності
1	LEGO Ninjago: Masters of Spinjitzu	TV Show	0,466004
2	LEGO Ninjago: Masters of Spinjitzu: Day of the Departed	Movie	0,304408
3	LEGO Jurassic World: Secret Exhibit	Movie	0,253546
4	LEGO Marvel Super Heroes: Black Panther	Movie	0,230022
5	Power Rangers Ninja Storm	TV Show	0,204124

Максимальна подібність (0,466) першої рекомендації є очікуваним результатом через належність до спільної франшизи. Система коректно виявляє не лише прямі сиквели чи приквели (серії LEGO), але й знаходить тематично спорідненний контент (наприклад, франшизу *Power Rangers*) завдяки перетину семантичних кластерів

(ключові слова «ніндзя», «бойові мистецтва», «дитячий контент») в описах сюжету [3].

Схожі результати з високою внутрішньосерійною близькістю (значення подібності > 0,3) були отримані при тестуванні запитів «Teenage Mutant Ninja Turtles» та «Power Rangers Dino Thunder».

Висновки. Розроблений програмний модуль довів свою дієздатність та ефективність. Використання інтегрованого текстового профілю об'єктів у поєднанні з методами NLP (TF-IDF векторизація та косинусна подібність) дозволяє створювати якісні, інтерпретовані рекомендації виключно на основі метаданих каталогу. Це повністю вирішує проблему «холодного старту» для нових фільмів на платформі. У перспективі розроблений алгоритмічний конвеєр може бути покращений шляхом впровадження семантичних векторних подань (Word Embeddings або Transformer-моделей) для ще глибшого розуміння контексту сюжетних ліній.

Список використаних джерел:

1. Коваленко О. М., Петренко І. В. Аналіз методів побудови рекомендаційних систем для потокових медіасервісів. *Інформаційні технології та комп'ютерна інженерія*. 2021. № 2. С. 43–52.
2. Loper E., Bird S. NLTK: The Natural Language Toolkit. *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics*. 2002. Vol. 1. P. 63–70.
3. Aggarwal C. C. Recommender Systems: The Textbook. Springer, 2016. 498 p.