

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

магістр

(назва освітнього ступеня)

на тему: Моделювання та аналіз системи рекомендацій на веб-сайті на основі
теорії графів і статистичних алгоритмів

Виконав: студент VI курсу, групи СНнм-61
спеціальності 122 Комп'ютерні науки
(шифр і назва спеціальності)

Курчак Ю.Р.
(підпис) (прізвище та ініціали)

Керівник Боднарчук І.О.
(підпис) (прізвище та ініціали)

Нормоконтроль Боднарчук І.О.
(підпис) (прізвище та ініціали)

Завідувач кафедри Боднарчук І.О.
(підпис) (прізвище та ініціали)

Рецензент Боднарчук І.О.
(підпис) (прізвище та ініціали)

Тернопіль
2025

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Боднарчук І.О.
(підпис) (прізвище та ініціали)

«___» _____ 202_ р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня Магістр
(назва освітнього ступеня)

за спеціальністю 122 Комп'ютерні науки
(шифр і назва спеціальності)

Студенту Курчак Юрій Романович
(прізвище, ім'я, по батькові)

1. Тема роботи Моделювання та аналіз системи рекомендацій на веб-сайті на основі теорії графів і статистичних алгоритмів

Керівник роботи Боднарчук Ігор Орестович, к.т.н., доцент кафедри КН
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від « 29 » листопада 2024 року № 4/7-1139

2. Термін подання студентом завершеної роботи

3. Вихідні дані до роботи Публікації тез в наукових конференціях ТНТУ

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ. 1 Аналіз предметної області, 1.1 Рекомендаційні системи як інструмент персоналізації контенту, 1.1.1 значення рекомендаційних систем у сучасних веб-сервісах, 1.1.2 галузі застосування, 1.1.3 основні вимоги до ефективної рекомендаційної системи, 1.2 класифікація рекомендаційних систем, 1.3 використання графових моделей у рекомендаційних системах, 1.4 алгоритми рекомендацій у графових моделях, 1.5 метрики оцінювання якості, 1.6 Аналіз існуючих рішень та інструментів побудови. 2. Аналіз методів, систем та проєктування рекомендаційних систем на основі теорії графів, 2.1 методологічні основи побудови рекомендаційних систем, 2.2 системна модель рекомендаційної платформи, 2.3 Аналіз системи рекомендацій на основі теорії графів, 2.4 інструменти та технології розробки, 2.4.1 серверні та клієнтські середовища (flask/django, react, api), 2.4.2 бібліотеки для реалізації алгоритмів, 2.5 особливості інтеграції рекомендацій в інтерфейс користувача, 2.6 архітектура зберігання та обробки даних, 3 Розробка веб-сайту з рекомендаційною системою, 3.1 постановка технічного завдання, 3.2 архітектура та функціональні компоненти системи рекомендацій, 3.2.1 загальна архітектура системи, 3.2.2 серверна логіка та реалізація рекомендацій, 3.3 реалізація основних сторінок сайту та функцій, 3.4 реалізація алгоритмів рекомендацій, 3.4.1 content-based filtering, 3.4.2 collaborative filtering 3.4.3 інші алгоритми системи рекомендацій, 3.5 інтеграція та візуалізація рекомендацій на веб-сайті, 3.6 тестування роботи сайту та перевірка рекомендацій, 4 Охорона праці та безпека в надзвичайних ситуаціях. Висновки. Перелік джерел. Додатки 5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	Сенчишин В.С., доцент		
Безпека в надзвичайних ситуаціях	Клепчик В.М., ст. викладач		

7. Дата видачі завдання 29 листопада 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	29.11.2024	
2.	Підбір наукових джерел про рекомендаційні системи	02.12.2024-10.01.2025	
3.	Опрацювання наукових публікацій та збір даних по темі роботи	13.01.2025-14.03.2025	
4.	Виконання дослідження згідно мети кваліфікаційної роботи	17.03.2025-28.03.2025	
5.	Оформлення розділу «Аналіз предметної області та основні підходи до побудови рекомендаційних систем»	31.03.2025-11.04.2025	
6.	Оформлення розділу «Аналіз методів, систем та проєктування рекомендаційних систем на теорії графів»	14.04.2025-25.04.2025	
7.	Оформлення розділу «Розробка веб-сайту з рекомендаційною системою»	28.04.2025-02.05.2025	
8.	Виконання завдання до підрозділу «Охорона праці»	28.04.2025-02.05.2025	
9.	Виконання завдання до підрозділу «Безпека в надзвичайних ситуаціях»	28.04.2025-02.05.2025	
10.	Оформлення кваліфікаційної роботи	05.05.2025-09.05.2025	
11.	Нормоконтроль	12.05.2025-15.05.2025	
12.	Перевірка на плагіат	16.05.2024	
13.	Попередній захист кваліфікаційної роботи	19.05.2025	
14.	Захист кваліфікаційної роботи	.05.2025	

Студент

(підпис)

(прізвище та ініціали)

Керівник роботи

(підпис)

(прізвище та ініціали)

АНОТАЦІЯ

Моделювання та аналіз системи рекомендацій на веб-сайті на основі теорії графів і статистичних алгоритмів // Кваліфікаційна робота освітнього рівня «Магістр» // Курчак Юрій Романович// Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних наук, група СНм-61 // Тернопіль, 2024 // С. 83, рис. – 17, табл. – 3, додат. – 2, бібліогр. – 50.

Ключові слова: рекомендаційна система, вебсайт, flask, фільми, персоналізація, користувач, алгоритми, база даних

Кваліфікаційна робота присвячена розробці вебсайту з інтегрованою системою персоналізованих рекомендацій фільмів.

У першому розділі кваліфікаційної роботи описані теоретичні основи побудови рекомендаційних систем. Висвітлено класифікацію підходів, розглянуто алгоритмічні моделі, проаналізовано принципи оцінювання ефективності.

У другому розділі кваліфікаційної роботи представлено загальну архітектуру проєкту. Досліджено інструменти розробки, подано опис серверних і клієнтських технологій, а також моделі даних.

У третьому розділі кваліфікаційної роботи описано процес реалізації сайту. Проаналізовано логіку функціональних модулів, проведено тестування працездатності системи. Об'єкт дослідження: процеси формування персоналізованих рекомендацій на вебплатформах. Предмет дослідження: методи реалізації алгоритмів рекомендацій у середовищі Flask.

ANNOTATION

Modeling and analysis of a website recommendation system based on graph theory and statistical algorithms // The educational level "Master" qualification work // Kurchak Yuriy Romanovych // Ternopil Ivan Pulyuy National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Computer Science, SNnm-61 group // Ternopil, 2024 // P. 83, fig. - 17, tables - 3, , annexes - 2, ref. - 50.

Key words: recommendation system, website, flask, movies, personalization, user, algorithms, database

The qualification work is devoted to the development of a website with an integrated system of personalized movie recommendations.

The first section of the qualification work describes the theoretical foundations of building recommendation systems. The classification of approaches is highlighted, algorithmic models are considered, and the principles of evaluating effectiveness are analyzed.

The second section of the qualification work presents the general architecture of the project. Development tools are studied, a description of server and client technologies, as well as data models are provided.

The third section of the qualification work describes the process of implementing the site. The logic of functional modules is analyzed, and the system's performance is tested. Object of research: processes of forming personalized recommendations on web platforms. Subject of research: methods of implementing recommendation algorithms in the Flask environment.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

API (англ. Application Programming Interface) – інтерфейс прикладного програмування для взаємодії між програмними компонентами.

Bootstrap 5 – фронтенд-фреймворк для створення адаптивної верстки.

CBF (англ. Content-Based Filtering) – метод рекомендацій, який базується на характеристиках об'єктів, з якими взаємодіяв користувач.

CF (англ. Collaborative Filtering) – метод рекомендацій, що ґрунтується на поведінці схожих користувачів.

CRUD (англ. Create, Read, Update, Delete) – базові операції з даними в базі даних.

CSS (англ. Cascading Style Sheets) – каскадні таблиці стилів для оформлення вебсторінок.

CSV (англ. Comma-Separated Values) – формат для зберігання табличних даних.

est (від estimate) – прогнозована оцінка, яка генерується алгоритмом на основі взаємодій інших користувачів.

Flask – мікрофреймворк Python для створення серверної частини вебдодатків.

GNN (англ. Graph Neural Network) – нейронна мережа, яка працює з графовими структурами.

HTML (англ. HyperText Markup Language) – мова гіпертекстової розмітки документів у вебсередовищі.

ItemID – ідентифікатор фільму або об'єкта, до якого застосовується рекомендація.

Jinja2 – шаблонізатор, що використовується у Flask для створення HTML-сторінок.

JSON (англ. JavaScript Object Notation) – формат обміну даними між клієнтом і сервером.

JS (англ. JavaScript) – мова програмування для динамічного керування вмістом вебсторінок.

MovieLike – модель бази даних, яка фіксує взаємодію користувачів із фільмами у вигляді лайків.

ORM (англ. Object-Relational Mapping) – технологія для взаємодії з базами даних через об'єкти мов програмування.

Prediction (est) – результат роботи алгоритму рекомендацій, що виражає очікувану оцінку об'єкта користувачем.

REST API – архітектурний стиль для розробки вебсервісів на основі HTTP-запитів.

SQL (англ. Structured Query Language) – мова структурованих запитів для роботи з реляційними базами даних.

SQLite – легка файлова система керування базами даних.

SRS (англ. Software Requirements Specification) – специфікація програмних вимог.

SVD (англ. Singular Value Decomposition) – сингулярний розклад матриці, що використовується в колаборативній фільтрації.

TF-IDF (англ. Term Frequency-Inverse Document Frequency) – метод статистичної обробки тексту.

TMDB (англ. The Movie Database) – онлайн-сервіс метаданих про фільми та серіали.

TMDB API (англ. The Movie Database Application Programming Interface) – інтерфейс прикладного програмування для отримання інформації про фільми з онлайн-бази TMDB.

Trainset – навчальний набір даних, сформований для моделі SVD на основі взаємодій користувачів.

UI (англ. User Interface) – інтерфейс користувача.

UserID – унікальний ідентифікатор користувача в базі даних.

UX (англ. User Experience) – досвід користувача при взаємодії із сайтом.

ЗМІСТ

ВСТУП	9
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ОСНОВНІ ПІДХОДИ ДО ПОБУДОВИ РЕКОМЕНДАЦІЙНИХ СИСТЕМ	11
1.1 Рекомендаційні системи як інструмент персоналізації контенту	11
1.1.1 Значення рекомендаційних систем у сучасних веб-сервісах.....	12
1.1.2 Галузі застосування.....	13
1.1.3 Основні вимоги до ефективної рекомендаційної системи.....	15
1.2 Класифікація рекомендаційних систем	18
1.3 Використання графових моделей у рекомендаційних системах	19
1.4 Алгоритми рекомендацій у графових моделях	21
1.5 Метрики оцінювання якості рекомендаційних алгоритмів.....	22
1.6 Аналіз існуючих рішень та інструментів для побудови рекомендаційних систем	23
1.7 Висновки до першого розділу	25
2 АНАЛІЗ МЕТОДІВ, СИСТЕМ ТА ПРОЄКТУВАННЯ РЕКОМЕНДАЦІЙНИХ СИСТЕМ НА ОСНОВІ ТЕОРІЇ ГРАФІВ.....	26
2.1 Методологічні основи побудови рекомендаційних систем....	26
2.2 Системна модель рекомендаційної платформи	27
2.3 Аналіз системи рекомендацій на веб-сайті на основі теорії графів і статистичних алгоритмів	29
2.4 Інструменти та технології розробки	31
2.4.1 Серверні та клієнтські середовища (Flask/Django, React, API)	32
2.4.2 Бібліотеки для реалізації алгоритмів	34
2.5 Особливості інтеграції рекомендацій в інтерфейс користувача	36
2.6 Архітектура зберігання та обробки даних у системі рекомендацій фільмів.....	37
2.7 Висновки до другого розділу	40
3 РОЗРОБКА ВЕБ-САЙТУ З РЕКОМЕНДАЦІЙНОЮ СИСТЕМОЮ	41
3.1 Постановка технічного завдання	41

3.2 Архітектура та функціональні компоненти системи рекомендацій	42
.....	42
3.2.1 Загальна архітектура системи	42
3.2.2 Серверна логіка та реалізація рекомендацій	43
3.3 Реалізація основних сторінок сайту та функцій.....	45
3.4 Реалізація алгоритмів рекомендацій.....	48
3.4.1 Content-Based Filtering (фільтрація на основі вмісту).....	49
3.4.2 Collaborative Filtering (колаборативна фільтрація)	49
3.4.3 Інші алгоритми системи рекомендацій	52
3.5 Інтеграція та візуалізація рекомендацій на веб-сайті	53
3.6 Тестування роботи сайту та перевірка рекомендацій.....	55
3.7 Висновок до третього розділу	56
4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ	58
.....	58
4.1 Обов'язки роботодавця щодо створення безпечних і нешкідливих умов праці	58
4.2 Правило техніки безпеки при експлуатації обладнання.....	61
4.3 Забезпечення захисту населення від впливу іонізуючих випромінювань	63
ВИСНОВКИ.....	69
ПЕРЕЛІК ДЖЕРЕЛ	71
ДОДАТКИ	

ВСТУП

Актуальність теми. У міру розвитку цифрових платформ і зростання обсягів медіаконтенту зростає і потреба в інтелектуальних системах, що здатні ефективно персоналізувати інформаційні потоки для кожного користувача. У цьому контексті рекомендаційні системи стають критично важливими інструментами, особливо для платформ, що працюють із великими колекціями фільмів, де звичайна навігація є неефективною. Зі збільшенням кількості джерел даних та варіативності поведінкових сценаріїв користувачів, удосконалення алгоритмів персоналізації та їх ефективна інтеграція у вебінтерфейс набувають особливого значення. Рекомендаційні системи, хоча й активно застосовуються в комерційних проєктах, часто реалізовані як закриті рішення, обмежені у можливостях гнучкого налаштування або аналізу. Саме тому розроблення власної вебплатформи з рекомендаційною системою на основі сучасних алгоритмічних підходів і відкритих програмних технологій є актуальним і перспективним напрямком наукових досліджень.

Мета і задачі дослідження. Метою даної кваліфікаційної роботи освітнього рівня «Магістр» є розроблення вебсайту з вбудованою системою рекомендацій фільмів на основі гібридного підходу до персоналізації користувацького досвіду. Для досягнення цієї мети необхідно виконати низку завдань:

- проаналізувати поточний стан розробок у галузі рекомендаційних систем;
- дослідити існуючі алгоритмічні моделі формування рекомендацій;
- розробити архітектуру вебплатформи з урахуванням структурних вимог до обробки великих обсягів даних;
- виконати порівняння ефективності застосованих алгоритмів у межах конкретного проєкту;
- реалізувати систему у вигляді повнофункціонального вебзастосунку на базі Flask.

Об'єкт дослідження. Процеси персоналізації користувацької взаємодії на вебсайтах на основі аналізу історії переглядів і вподобань.

Предмет дослідження. Алгоритмічні методи формування рекомендацій і технології їх реалізації в умовах вебзастосунку на прикладі сайту з фільмами.

Наукова новизна одержаних результатів. Наукова новизна роботи полягає в інтеграції класичних методів колаборативної та контентної фільтрації з додатковими евристичними алгоритмами (на основі популярності, тегів, настроїв) у єдиній системі, адаптованій для роботи в середовищі Flask. У межах проекту створено прототип гібридної системи рекомендацій, орієнтованої на практичне використання в реальних умовах динамічного вебресурсу.

Практичне значення одержаних результатів. Розроблено повнофункціональний вебсайт із вбудованою системою персоналізованих рекомендацій, що базується на відкритих бібліотеках Python. Система реалізована у вигляді інтерактивного сервісу з підтримкою реєстрації користувачів, динамічного відображення рекомендацій та адміністративного керування контентом.

Апробація результатів магістерської роботи. Основні результати дослідження були обговорені на XII науково-технічній та XIII міжнародній студентській науково-практичній науковій конференції молодих учених та студентів Тернопільського національного технічного університету імені Івана Пулюя (м. Тернопіль, 2024 р.).

Публікації. Основні результати кваліфікаційної роботи опубліковано у двох працях конференції (Див. додатки А).

Структура й обсяг кваліфікаційної роботи. Кваліфікаційна робота складається зі вступу, чотирьох розділів, висновків, списку літератури з 50 найменувань та 2 додатків. Загальний обсяг кваліфікаційної роботи складає 83 сторінки, з них 60 сторінок основного тексту, який містить 18 рисунків та 3 таблиць.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ОСНОВНІ ПІДХОДИ ДО ПОБУДОВИ РЕКОМЕНДАЦІЙНИХ СИСТЕМ

1.1 Рекомендаційні системи як інструмент персоналізації контенту

Рекомендаційні системи дозволяють кожному користувачеві отримувати контент, максимально відповідний його інтересам, на основі аналізу поведінки, історії переглядів, уподобань та інших факторів.

Принцип роботи рекомендаційних систем ґрунтується на зборі даних про користувачів, їхній взаємодії з платформою та подальшому прогнозуванні, який контент може їх зацікавити. Основна ідея персоналізації полягає у відображенні саме тих товарів, послуг або матеріалів, які мають високу ймовірність привернути увагу конкретного користувача [1].

Персоналізація контенту за допомогою рекомендаційних систем може бути індивідуальною, груповою або глобальною. Опис такої персоналізації візуалізовано в таблиці 1.1.

Таблиця 1.1 – Рівні персоналізації контенту

Рівень персоналізації	Опис	Приклад застосування
Індивідуальні рекомендації	Аналіз поведінки конкретного користувача	Рекомендація товару на основі минулих покупок
Групові рекомендації	Врахування поведінки схожих користувачів	"Користувачі, які купили цей товар, також купили..."
Глобальні рекомендації	Популярні товари серед усіх користувачів	"Найпопулярніші товари тижня"

Ефективність персоналізації залежить від якості алгоритмів, що використовуються, та рівня аналізу вхідних даних. Використання графових моделей, машинного навчання та глибоких нейромереж дозволяє значно підвищити точність рекомендацій, забезпечуючи кращий досвід користувачів.

1.1.1 Значення рекомендаційних систем у сучасних веб-сервісах

Веб-сервіси, які працюють із великою кількістю даних, активно впроваджують рекомендаційні системи, щоб спростити цей процес і зробити взаємодію з платформою більш зручною.

Рекомендаційні алгоритми аналізують поведінку користувачів, визначаючи закономірності у виборі товарів, відео, музики чи інших матеріалів. Вони допомагають не тільки передбачати інтереси окремих відвідувачів, а й створювати персоналізований досвід для кожного. Наприклад, у сфері електронної комерції рекомендації можуть підштовхнути користувача до купівлі супутніх товарів, у стрімінгових сервісах – допомогти знайти фільми чи музику, які користувачу потенційно сподобаються. Рекомендації можуть також враховувати історію пошукових запитів користувача. Зібрані дані про перегляди, оцінки та реакції можуть бути використані для аналізу глядацьких вподобань, визначення трендів та оптимізації контентної стратегії платформи [2].

Аналогічно до фільмів, система може рекомендувати товари, які можуть зацікавити користувача на основі його попередніх покупок, переглядів, доданих до кошика товарів, а також вподобань інших користувачів зі схожою поведінкою.

Важливим аспектом роботи таких систем є їхній внесок у підвищення рівня залученості аудиторії. Якщо користувач отримує контент, що відповідає його вподобанням, він проводить більше часу на платформі, що сприяє зростанню активності та взаємодії. Крім того, точні рекомендації впливають на довіру до сервісу, оскільки створюють враження, що платформа розуміє потреби користувача [3].

Значення рекомендаційних систем виходить за межі покращення користувацького досвіду. Вони також є потужним інструментом бізнес-

оптимізації, адже дозволяють компаніям ефективніше просувати товари та послуги, збільшувати прибутки та утримувати клієнтів. З огляду на постійне зростання обсягів інформації та конкуренцію між цифровими платформами, впровадження ефективних механізмів рекомендацій стає невід'ємною складовою успішного функціонування веб-сервісів.

1.1.2 Галузі застосування

Рекомендаційні системи широко використовуються у різних сферах, де необхідно персоналізувати взаємодію з користувачем та оптимізувати вибір контенту чи товарів. Вони стали невід'ємною частиною цифрового середовища, забезпечуючи автоматизований відбір інформації відповідно до індивідуальних уподобань кожного відвідувача.

Однією з найпоширеніших галузей впровадження таких систем є електронна комерція. Онлайн-магазини аналізують поведінку покупців, історію їхніх замовлень, переглядів та взаємодій із товарами, щоб пропонувати найбільш релевантні варіанти [4]. Це не лише покращує користувацький досвід, а й сприяє зростанню обсягів продажів за рахунок вдалих персоналізованих пропозицій. До прикладу, рекомендації в інтернет-магазині Rozetka які зображено на рисунку 1.1.

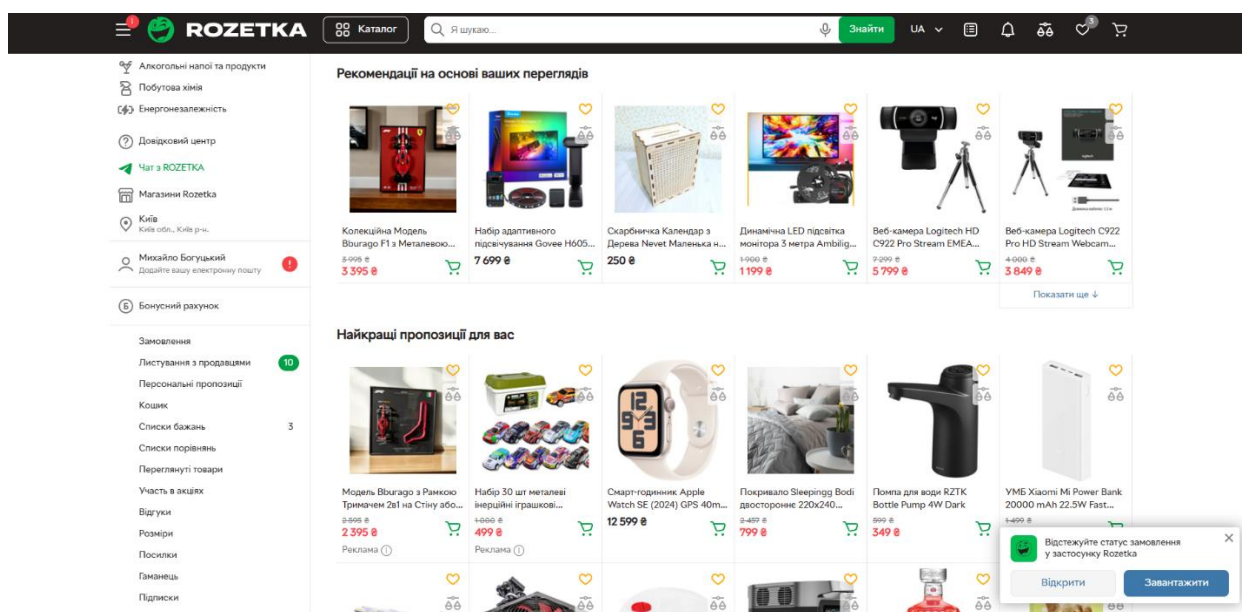


Рисунок 1.1 – Рекомендаційна система на Rozetka

Стрімінгові сервіси також активно використовують алгоритми рекомендацій. Платформи з відео- та музичним контентом підбирають фільми, серіали, пісні або подкасти на основі уподобань користувача, його оцінок, тривалості перегляду та подібностей із іншими користувачами [5]. Завдяки цьому створюється індивідуальний інформаційний простір, що мотивує людину проводити більше часу на платформі та відкривати новий контент, візуалізацію однієї з таких систем зображено на рис. 1.2.

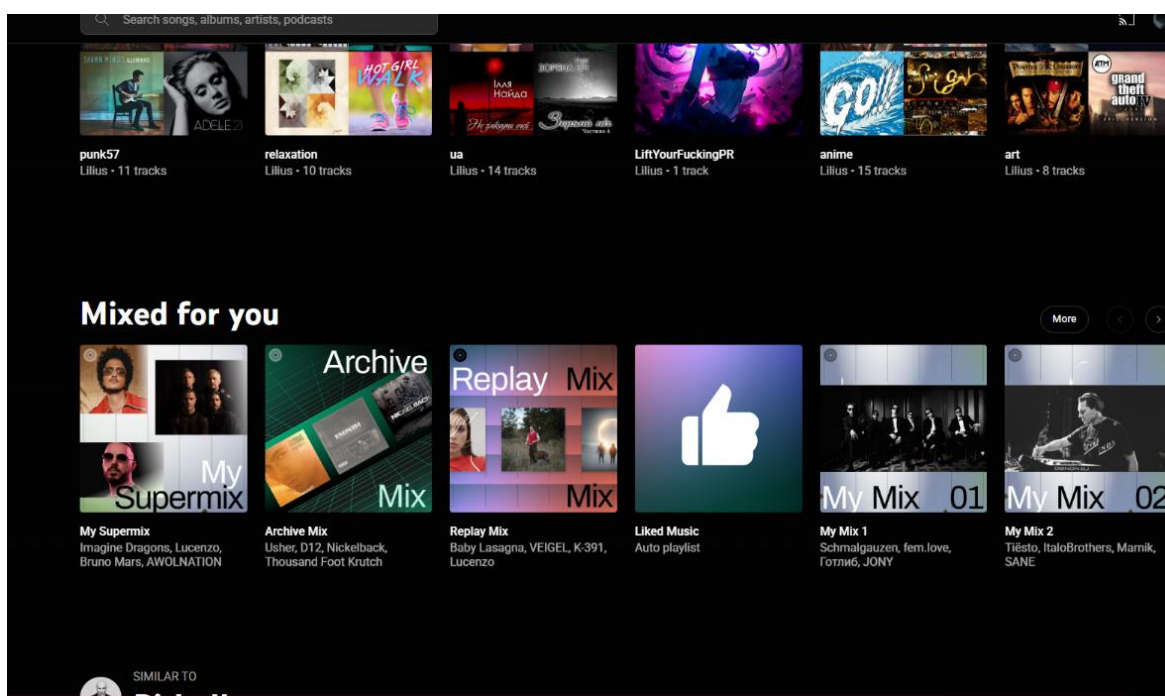


Рисунок 1.2 – Рекомендаційна система на Youtube music

До прикладу, на таких платформах як Spotify, Apple music або Youtube music такі рекомендації використовуються найактивніше, і рекомендації на таких платформах видають більш персоналізовані рекомендації за рахунок великої кількості даних, адже користувачі таких платформ слухають певну композицію частіше чим оформляють замовлення в тому ж інтернет-магазині.

У сфері освіти рекомендаційні системи допомагають адаптувати навчальний процес під конкретного студента. Аналізуючи попередні результати тестів, пройдені курси та інтереси, системи формують персоналізовані навчальні траєкторії, пропонуючи відповідні матеріали для поглиблення знань.

Активне впровадження рекомендацій відбувається і в новинних агрегаторах та соціальних мережах, де алгоритми відбирають найрелевантніші статті, публікації або відео відповідно до вподобань користувача. Це сприяє підвищенню залученості аудиторії, але водночас може призводити до ефекту «інформаційної бульбашки», коли людині показують лише ті новини, що відповідають її поглядам.

Застосування рекомендаційних систем суттєво впливає на цифрову економіку, підвищуючи ефективність бізнесу та покращуючи взаємодію між платформами та користувачами. У майбутньому ці технології продовжать розвиватися, інтегруючи нові алгоритми штучного інтелекту для ще більш точного прогнозування потреб аудиторії.

1.1.3 Основні вимоги до ефективної рекомендаційної системи

Рекомендаційні системи повинні відповідати ряду вимог, які забезпечують їхню ефективність і користь для користувачів. Одним із ключових аспектів є персоналізація, що дозволяє системі адаптуватися до унікальних інтересів кожного відвідувача. Вона досягається через глибокий аналіз даних, включаючи історію переглядів, оцінки, покупки та інші параметри взаємодії, а також, точність передбачення інтересів користувача. Вона повинна аналізувати історію дій, виявляти приховані закономірності та будувати прогнози, що дійсно відповідають очікуванням людини. Якщо алгоритм часто пропонує нерелевантний контент, користувач швидко втрачає довіру до рекомендацій, що може негативно вплинути на його активність на платформі. Без якісної персоналізації система може видавати загальні або випадкові рекомендації, які не залучатимуть користувача до подальшої взаємодії [5].

Прозорість алгоритмів також має значення. Користувачі мають розуміти, чому їм пропонуються певні товари чи контент. Це допомагає підвищити довіру до системи та покращує її сприйняття. У деяких сервісах передбачають можливість налаштовувати параметри рекомендацій, щоб людина могла самостійно впливати на те, що вона бачить [5].

Ще одним важливим фактором є стійкість до холодного старту. Це проблема, яка виникає при роботі з новими користувачами або товарами, для яких ще немає достатньо даних. У таких випадках система повинна вміти генерувати початкові рекомендації, використовуючи загальні тренди, мета-дані або змішані підходи. Опис цих факторів описаний в таблиці 1.2

Таблиця 1.2 – Фактори які виникають при роботі з новими даними

Проблема	Причина виникнення	Наслідки	Методи вирішення
Новий користувач	Відсутність історії взаємодій	Нерелевантні рекомендації, низька зацікавленість	Використання демографічних даних, популярного контенту або початкових опитувань
Новий товар	Відсутність попередніх переглядів та покупок	Товар залишається малопомітним	Аналіз категорій, мета-даних або характеристик для створення зв'язків із подібними продуктами
Зміна вподобань користувачів	Динамічні інтереси змінюються з часом	Актуальність рекомендацій знижується	Постійне оновлення моделей, використання алгоритмів адаптації до змін
Обмежена кількість даних	Малий набір користувачів або контенту	Низька якість рекомендацій	Використання гібридних методів, що поєднують колаборативну та контентну фільтрацію

Не менш важливою вимогою є стійкість до маніпуляцій. Деякі користувачі чи продавці можуть намагатися впливати на рекомендації, створюючи штучні

оцінки або масово переглядаючи певний контент. Ефективна система повинна включати механізми для виявлення аномалій та фільтрації подібних маніпуляцій, щоб рекомендації залишалися чесними та корисними.

Окрім цього, система має бути універсальною, тобто підтримувати різні типи рекомендацій (персоналізовані, групові, загальні) та адаптуватися до змін у поведінці користувачів. Важливо, щоб рекомендаційні алгоритми могли функціонувати в різних масштабах і обробляти великі обсяги даних без значного падіння продуктивності.

Гнучкість і адаптивність рекомендаційних систем є важливими характеристиками, які впливають на їхню ефективність. Одна з ключових вимог, це стійкість до маніпуляцій, що означає здатність алгоритму розпізнавати та запобігати штучному впливу на рекомендації. Це актуально для платформ, де недобросовісні користувачі або продавці можуть намагатися маніпулювати рейтингами товарів чи популярністю контенту. Для вирішення цієї проблеми застосовуються методи виявлення аномальної активності, що дозволяють зберегти об'єктивність рекомендацій.

Ще одним важливим аспектом є масштабованість системи, адже вона повинна ефективно працювати з великими обсягами даних та зростаючою кількістю користувачів. Для цього використовуються розподілені обчислення, кешування та оптимізовані алгоритми, що забезпечують швидке формування рекомендацій навіть за високих навантажень.

Адаптивність алгоритмів гарантує їхню здатність до змін відповідно до нових інтересів користувачів. Якщо система базується лише на історичних даних, рекомендації можуть швидко втрачати актуальність. Використання алгоритмів машинного навчання, які постійно оновлюють моделі на основі нових дій користувача, допомагає підтримувати точність персоналізованих пропозицій.

Гнучкість налаштувань є ще одним важливим фактором. Надання користувачам можливості впливати на систему дозволяє коригувати рекомендації відповідно до їхніх вподобань. Наприклад, можливість приховувати певні категорії контенту або змінювати пріоритети у рекомендаціях покращує досвід взаємодії з платформою.

1.2 Класифікація рекомендаційних систем

Рекомендаційні системи можна класифікувати за методами обробки даних та принципами формування прогнозів. Вибір алгоритму залежить від наявних даних, особливостей користувацької бази та вимог до швидкодії. Сучасні технології, такі як Python та JavaScript, пропонують широкий набір інструментів для реалізації різних типів рекомендаційних систем.

Одним із найпоширеніших підходів є контентна фільтрація, що аналізує властивості об'єктів і на їх основі формує рекомендації. Наприклад, у Python бібліотека Scikit-learn дозволяє будувати подібні системи, використовуючи векторизацію текстових описів товарів за допомогою TF-IDF та пошук схожих об'єктів через cosine similarity. Для веб-додатків, що працюють у браузері, подібний аналіз можна реалізувати на JavaScript за допомогою TensorFlow.js, що дозволяє виконувати обчислення прямо на клієнтському боці [8].

Колаборативна фільтрація ґрунтується на взаємодіях між користувачами. Вона передбачає, що люди з подібними інтересами мають схожі переваги у виборі контенту. У Python для реалізації такого підходу часто використовують бібліотеку Surprise, яка містить алгоритми, зокрема Singular Value Decomposition (SVD), що дозволяє знаходити приховані взаємозв'язки між користувачами та продуктами. У JavaScript можна застосовувати Node.js + PostgreSQL, де історія дій зберігається у базі даних і обробляється серверною логікою для знаходження схожих користувачів або продуктів.

Гібридні системи поєднують переваги кількох підходів, комбінуючи аналіз характеристик товарів та поведінку користувачів. Наприклад, можна використовувати LightFM у Python, що підтримує як контентну, так і колаборативну фільтрацію. У веб-додатках можна інтегрувати GraphQL API, який обробляє запити на стороні сервера, поєднуючи дані з різних джерел для створення точніших рекомендацій [7].

Окрему нішу займають графові рекомендаційні системи, що будуються на аналізі зв'язків між об'єктами та користувачами. У Python їх можна реалізувати за допомогою Neo4j та NetworkX, що дозволяють будувати графові структури та

застосовувати алгоритми, такі як Personalized PageRank. У JavaScript клієнтська частина може візуалізувати графи за допомогою бібліотеки D3.js, що забезпечує інтерактивне відображення взаємозв'язків між елементами.

Таким чином, вибір технології та підходу до рекомендаційної системи залежить від цілей проєкту, доступних даних та необхідного рівня персоналізації. Сучасні інструменти Python і JavaScript дозволяють будувати масштабовані та гнучкі рішення для аналізу та прогнозування інтересів користувачів.

1.3 Використання графових моделей у рекомендаційних системах

Графова структура в рекомендаційних системах будується на основі вузлів, які можуть представляти користувачів, товари або категорії, а також ребер, що відображають взаємодії між ними. Наприклад, у випадку онлайн-магазину вузли можуть позначати покупців і товари, а ребра — їхні покупки або перегляди. Використання такої моделі дозволяє ефективно знаходити подібності між користувачами та прогнозувати, які товари можуть їх зацікавити.

Завдяки алгоритмам аналізу графів можна отримати точніші рекомендації. Одним із найпопулярніших підходів є метод випадкових блукань, коли система імітує переміщення по графу, поступово переходячи від одного вузла до іншого. Це дає змогу враховувати зв'язки не лише між безпосередньо взаємодіючими об'єктами, а й між тими, що знаходяться на більшій відстані в мережі.

Графові моделі також дозволяють вирішувати проблему холодного старту, коли система має обмежену інформацію про нового користувача або товар. У таких випадках можна використовувати знання про зв'язки в мережі, щоб знаходити релевантні рекомендації, навіть якщо немає історичних даних про взаємодію конкретного користувача [9].

У сучасних веб-сервісах для реалізації графових рекомендаційних систем широко використовуються технології, що працюють із графовими базами даних, такі як Neo4j, або бібліотеки для аналізу графів у Python, зокрема NetworkX та Deep Graph Library (DGL). Вони дозволяють будувати глибокі графові нейронні

мережі, які здатні навчатися на складних структурах даних та робити більш персоналізовані прогнози.

Графові алгоритми, такі як Personalized PageRank або Random Walk with Restart, дозволяють знаходити найбільш релевантні рекомендації, враховуючи структуру взаємозв'язків. Вони добре масштабуються на великі обсяги даних і дозволяють враховувати не лише безпосередні зв'язки, а й схожі об'єкти, що знаходяться далі в мережі [9].

Графи використовуються для моделювання взаємозв'язків між користувачами та продуктами, що дозволяє ефективно знаходити схожі товари, будувати персоналізовані рекомендації та аналізувати взаємодію всередині системи. У такій моделі вузли графа можуть представляти користувачів та товари, а ребра – зв'язки між ними, що відображають історію взаємодій, наприклад, перегляди, оцінки чи покупки (Див. рис. 1.3).

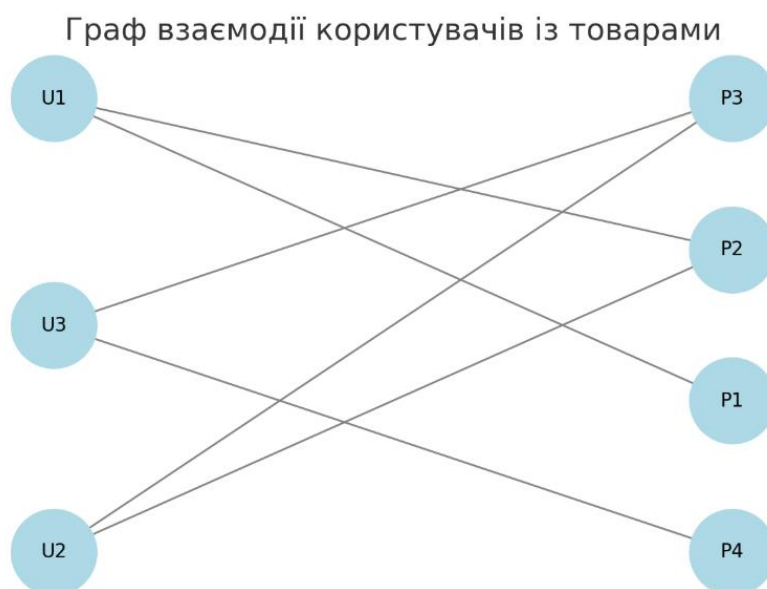


Рисунок 1.3 – Граф взаємодії користувачів з товарами

Вузли "U1", "U2" та "U3" представляють користувачів, а "P1", "P2", "P3" та "P4" – товари. Ребра показують, які товари переглядав або купував кожен користувач. Така модель допомагає визначати потенційні рекомендації на основі зв'язків між елементами.

1.4 Алгоритми рекомендацій у графових моделях

Графові моделі рекомендаційних систем використовують спеціалізовані алгоритми, які аналізують взаємозв'язки між вузлами та визначають найбільш релевантні об'єкти для користувачів. На відміну від класичних підходів, що базуються на матричних розкладаннях або статистичних методах, графові алгоритми дозволяють виявляти складні структури взаємодії між користувачами та контентом, що робить їх більш гнучкими та ефективними у великомасштабних системах.

Однією з ключових концепцій у графових рекомендаційних алгоритмах є пошук схожих об'єктів через аналіз зв'язків. Алгоритм може враховувати як прямі взаємодії користувача з об'єктами, так і непрямі залежності, що формуються через інших користувачів або подібні товари. Використання такого підходу дозволяє визначати зв'язки навіть між тими елементами, які раніше не мали спільних точок дотику [10].

Важливим аспектом графових алгоритмів є їхня здатність ранжувати вузли за ступенем важливості. Наприклад, певні товари можуть мати вищий рейтинг у системі, якщо вони часто зустрічаються у взаємодіях із користувачами, а рекомендації можуть враховувати ці закономірності, щоб пропонувати більш популярні або якісні варіанти. Водночас алгоритми можуть моделювати не тільки статичні зв'язки, а й враховувати зміни у поведінці користувачів у реальному часі, що дозволяє адаптуватися до актуальних інтересів аудиторії.

Глибокі графові нейронні мережі також знаходять своє застосування у побудові рекомендаційних моделей. Вони аналізують складні залежності між об'єктами та навчаються прогнозувати найбільш ймовірні варіанти вибору користувачів, використовуючи багаторівневі зв'язки. Такі методи дозволяють значно підвищити точність рекомендацій і працюють навіть у ситуаціях, коли традиційні алгоритми не дають достатньо якісних результатів [11].

Інтеграція графових алгоритмів у сучасні рекомендаційні системи дозволяє значно покращити точність і релевантність підбору контенту, оптимізувати швидкість роботи алгоритмів та забезпечити масштабованість для

роботи з великими обсягами даних. Використання цих методів у комерційних веб-платформах сприяє покращенню користувацького досвіду, збільшенню конверсії та підвищенню рівня задоволеності клієнтів.

1.5 Метрики оцінювання якості рекомендаційних алгоритмів

Від правильного вибору метрик залежить, наскільки система буде відповідати очікуванням користувачів і виконувати свої функції, зокрема підвищення залученості та конверсії.

Оцінка якості рекомендаційних алгоритмів зазвичай базується на аналізі точності передбачень, ранжування результатів і здатності системи адаптуватися до змінних вподобань користувачів. Якщо рекомендації не відповідають реальним інтересам людини, це може призвести до зниження довіри до системи та зменшення активності на платформі. Тому алгоритми повинні не лише пропонувати популярні товари чи контент, а й враховувати персональні вподобання користувачів [27].

Важливим аспектом є баланс між точністю та різноманітністю рекомендацій. Якщо алгоритм надто сильно орієнтується на попередню поведінку користувача, він може не пропонувати новий контент, обмежуючи можливості відкриття нового досвіду. З іншого боку, надмірна різноманітність без урахування інтересів користувача може знизити точність передбачень. Саме тому метрики мають враховувати як персоналізацію, так і різноманітність пропозицій [28].

Ще одним критерієм оцінювання є швидкість роботи алгоритму. У великих системах, де обробляються мільйони запитів на секунду, швидкодія є критично важливою. Навіть найбільш точні рекомендації будуть марними, якщо вони генеруються із затримкою, що погіршує користувацький досвід. Тому в реальних умовах потрібно аналізувати не лише якість рекомендацій, а й продуктивність алгоритму.

1.6 Аналіз існуючих рішень та інструментів для побудови рекомендаційних систем

Сучасні комерційні рішення в цій сфері базуються на гібридних підходах, поєднуючи різні алгоритми, що працюють із великими наборами даних у режимі реального часу. Серед інструментів, що дозволяють розробляти такі системи, особливе місце займають графові бази даних і бібліотеки для обробки графів. Вони забезпечують ефективний аналіз взаємозв'язків між користувачами, товарами та їхньою поведінкою. Наприклад, використання графових алгоритмів у поєднанні з нейронними мережами дозволяє отримувати більш складні й адаптивні рекомендації, які враховують навіть непрямі зв'язки між елементами.

Окрім графових підходів, важливу роль відіграють алгоритми машинного навчання. У Python для їхньої реалізації найчастіше використовують Scikit-learn, LightFM та TensorFlow, які підтримують як класичні методи колаборативної фільтрації, так і глибокі нейронні мережі. Наприклад, для побудови рекомендаційної системи на основі факторизації матриць можна використати бібліотеку Surprise, що дозволяє працювати з алгоритмами SVD та ALS [17].

Алгоритм SVD (Singular Value Decomposition) є потужним методом матричної факторизації, який широко використовується в рекомендаційних системах, зокрема в підході Collaborative Filtering. Алгоритм SVD оперує з матрицею взаємодій "користувачі-фільми", де рядки представляють користувачів, стовпці – фільми, а значення в комірках відображають взаємодію.

Алгоритм ALS (Alternating Least Squares) є ще одним методом матричної факторизації, який може бути використаний для реалізації Collaborative Filtering у розробленому проєкті рекомендаційної системи, особливо якщо розглядати альтернативи або розширення поточного використання SVD [27].

Подібно до SVD, ALS працює з матрицею взаємодій "користувачі-фільми". Його мета також полягає у розкладанні цієї матриці на дві низькорангові матриці. Основна відмінність ALS полягає в способі знаходження цих матриць. Замість одночасного розкладання, ALS використовує ітеративний підхід.

Графові бази даних є одним із найефективніших інструментів для побудови рекомендаційних систем, оскільки дозволяють аналізувати складні взаємозв'язки між користувачами та контентом. Наприклад, Neo4j є потужною графовою базою, яка підтримує алгоритми PageRank, Random Walk with Restart та Personalized PageRank [10].

Таблиця 1.3 – Порівняння графових алгоритмів у рекомендаційних системах

Алгоритм	Принцип роботи	Переваги	Недоліки
PageRank	Визначає важливість вузлів у графі, оцінюючи кількість і якість зв'язків	Добре підходить для визначення популярних об'єктів	Погано адаптується до змінних вподобань користувачів
Random Walk with Restart	Випадковий обхід графа з періодичним поверненням до стартового вузла	Враховує як глобальні, так і локальні зв'язки	Високе обчислювальне навантаження на великих графах
Personalized PageRank	Модифікація PageRank, що персоналізує результати для кожного користувача	Більш точні рекомендації порівняно зі стандартним PageRank	Потребує додаткових ресурсів для обчислення
Graph Neural Networks (GNNs)	Використовує глибоке навчання для аналізу зв'язків у графах	Здатні враховувати складні патерни взаємодій	Високі вимоги до обчислювальних ресурсів

У веб-додатках, побудованих на JavaScript, також доступні інструменти для реалізації рекомендацій. Наприклад, TensorFlow.js дозволяє виконувати машинне навчання на стороні клієнта, що зменшує затримки при формуванні рекомендацій. Серверні системи на Node.js можуть використовувати зв'язок із базами даних, такими як PostgreSQL або MongoDB, для збереження та обробки взаємодій користувачів із контентом.

Для роботи з графовими моделями у JavaScript можна використовувати D3.js, що дозволяє створювати інтерактивні візуалізації графів, або Graphology, яка містить базові алгоритми обробки графових структур. Наприклад, реалізація алгоритму PageRank у Node.js може здійснюватися через graphology-pagerank, що дозволяє швидко аналізувати взаємозв'язки між об'єктами.

Для реалізації рекомендаційних систем існує широкий вибір інструментів як у Python, так і в JavaScript. Python забезпечує потужні засоби для обробки великих даних, машинного навчання та графових алгоритмів, тоді як JavaScript дозволяє інтегрувати рекомендаційні системи безпосередньо у веб-застосунки, забезпечуючи швидкість обробки та візуалізацію взаємозв'язків.

1.7 Висновки до першого розділу

В першому розділі кваліфікаційної роботи освітнього рівня «Магістр» описано основні підходи до побудови рекомендаційних систем та їхню класифікацію. Контентна фільтрація дозволяє формувати рекомендації на основі характеристик об'єктів, тоді як колаборативна фільтрація аналізує взаємодію користувачів із контентом, шукаючи подібні патерни поведінки. Гібридні методи поєднують переваги обох підходів, забезпечуючи кращу точність рекомендацій. Окрему увагу було приділено графовим моделям, які завдяки структурі зв'язків між елементами дозволяють глибше аналізувати взаємодії та знаходити приховані закономірності у виборі користувачів.

Огляд існуючих рішень та інструментів продемонстрував, що для побудови сучасних рекомендаційних систем широко використовуються графові бази даних, бібліотеки машинного навчання та розподілені обчислення.

2 АНАЛІЗ МЕТОДІВ, СИСТЕМ ТА ПРОЄКТУВАННЯ РЕКОМЕНДАЦІЙНИХ СИСТЕМ НА ОСНОВІ ТЕОРІЇ ГРАФІВ

2.1 Методологічні основи побудови рекомендаційних систем

Рекомендаційні системи як технологічна концепція виникли у відповідь на необхідність інтелектуального фільтрування цифрового контенту для кожного окремого користувача. Їхнє завдання полягає у передбаченні потенційного інтересу користувача до певного об'єкта, який ще не був ним обраний або переглянутий. Методологічно такі системи реалізуються як предиктивні моделі, що ґрунтуються на попередньому досвіді, структурних властивостях контенту, поведінкових шаблонах або спільній активності користувачів.

У межах цього проєкту реалізація рекомендаційної системи базується на поєднанні декількох концептуально різних підходів. Один із них — контентно-орієнтована фільтрація яка ґрунтується на обчисленні схожості між фільмами за набором метаданих, таких як жанри, актори, ключові слова й опис. Реалізація здійснюється через функції зіставлення тегів та використання методу `similar_movies()` API платформи TMDb, що слугує головним джерелом фільмового контенту в системі. Такий підхід дозволяє створити зв'язки між об'єктами, які подібні, незалежно від користувача, що переглядає [30].

Другим базовим принципом є колаборативна фільтрація, що аналізує схожість між користувачами та їхніми діями. На відміну від контентного підходу, в цьому випадку алгоритм орієнтується не на сам фільм, а на структуру вподобань спільноти. Якщо кілька користувачів виявляють однакові інтереси, система може запропонувати одному з них ті фільми, які вже були високо оцінені іншими. У проєкті така логіка реалізована через алгоритм SVD, що працює з узагальненою матрицею взаємодій між користувачами і фільмами. Необхідною умовою для цього методу є наявність історії вподобань або лайків, які збираються через систему реакцій [31].

Важливим етапом формування рекомендацій стала гібридна модель, що поєднує переваги обох вищезазначених підходів. Вона дозволяє уникнути

обмежень кожного з них, зокрема "проблеми холодного старту" в колаборативному підході або надмірної лінійності в контентній фільтрації. У структурі вебсайту блоки рекомендацій формуються на базі комбінованих даних і за допомогою логіки, що забезпечує зважений вибір релевантного контенту.

Окремо варто згадати допоміжні методи: алгоритм трендів, який ґрунтується на даних про перегляди або оцінки за останній період, тегові рекомендації на основі жанрової класифікації, нечіткий пошук за допомогою Levenshtein Distance для обробки неточних запитів, а також динамічний алгоритм відображення, що адаптує контент на головній сторінці згідно з актуальністю та активністю. У перспективі система також враховує аналіз настроїв, який дозволяє враховувати емоційні оцінки фільмів на основі реакцій користувачів.

Усі ці методи взаємодіють в архітектурі вебзастосунку, де адміністрування реалізоване через закритий доступ /admin, а всі метадані фільмів надходять із TMDB API [33].

2.2 Системна модель рекомендаційної платформи

Функціонування рекомендаційної системи в межах вебплатформи передбачає взаємодію кількох логічно зв'язаних рівнів, кожен з яких виконує окрему роль у загальному процесі генерації персоналізованого контенту. В основі моделі лежить послідовне перетворення вхідної інформації про фільми, користувацьку активність та зовнішні метадані в структуровані рекомендації, що подаються у вебінтерфейсі.

Система функціонує як багаторівнева структура, де перший рівень відповідає за отримання даних — це, зокрема, інтеграція з TMDB API, що забезпечує регулярне поповнення локальної бази інформацією про фільми: назви, описи, жанри, актори, ключові слова, рейтинги. На цьому ж рівні фіксується взаємодія користувача з платформою — лайки, перегляди, пошукові запити, реакції. Усі ці дії записуються у відповідні таблиці бази даних і становлять основу поведінкової аналітики.

Другий рівень обробки включає алгоритмічне ядро системи. Тут відбувається аналіз накопиченої інформації — спочатку шляхом попередньої фільтрації, нормалізації та структурування даних, далі — через застосування відповідного алгоритму рекомендацій (контентного, колаборативного чи гібридного). Результати моделювання зберігаються тимчасово або передаються у відповідь на клієнтський запит через REST API. Завдяки цьому структура є гнучкою і дозволяє масштабувати систему, додаючи нові методи аналізу без зміни загальної архітектури [33].

Третій рівень — це візуалізація рекомендацій у браузері користувача. Інтерфейс сформований так, щоб окремі блоки рекомендацій могли відображати різні типи логіки — від персоналізованих до трендових або тематичних. Запити надсилаються до API сервера, а у відповідь приходять сформовані списки фільмів, які відразу інтегруються у сторінку без перезавантаження. Це забезпечується за рахунок асинхронної комунікації між фронтендом і бекендом (див. рис. 2.1).

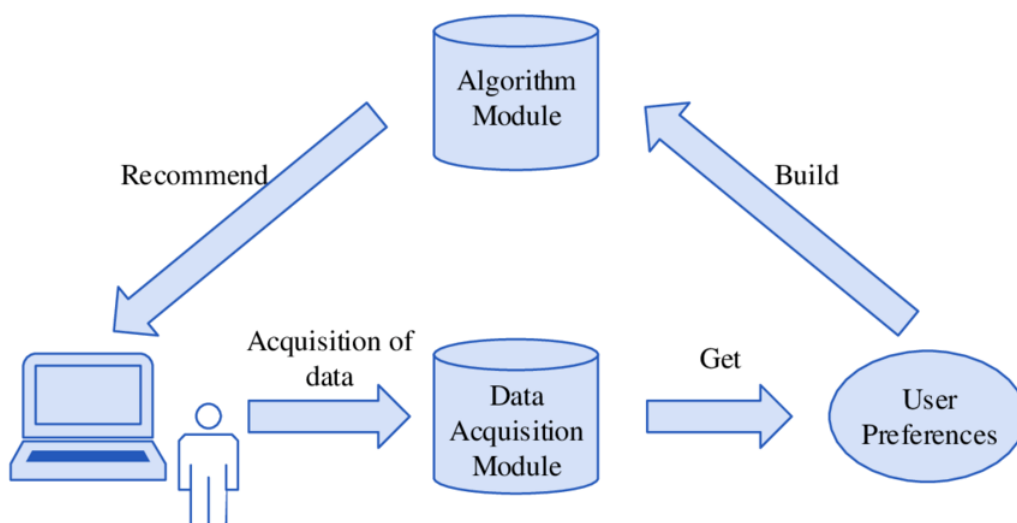


Рисунок 2.1 – Взаємодія компонентів рекомендаційної системи

Усі ці рівні пов'язані потоками даних, які рухаються в основному напрямку: від користувача до серверної частини (збір активності) і до кінця — у формі рекомендацій.

ліній або кривих. Товщина ребер або інтенсивність кольору вказує на ступінь схожості, де товстіші лінії або яскравіші кольори означають більшу схожість.

Ця візуалізація дозволяє швидко ідентифікувати кластери фільмів на основі спільних характеристик, сприяючи вивченню зв'язків між фільмами та відкриттю пов'язаних фільмів на основі вподобань глядачів або тематичних зв'язків. Загальна мережа графа візуалізує складну мережу зв'язків на основі взаємозв'язку фільмів через спільні характеристики, забезпечуючи динамічне розуміння контенту фільмів та потенційних рекомендацій.

Для глибшого аналізу взаємодії користувачів з контентом застосовується дводольний граф «користувачі-фільми». У цій структурі одна множина вузлів представляє користувачів, а інша – фільми. Ребро між вузлом користувача та вузлом фільму існує за умови, що даний користувач переглянув або оцінив цей фільм. Вага такого ребра може відображати надану користувачем оцінку. Ця модель є фундаментальною для алгоритмів матричної факторизації, які широко використовуються в Collaborative Filtering для виявлення латентних факторів, що визначають вподобання користувачів та характеристики фільмів. Проектування цього дводольного графа на односторонні графи (граф користувачів або граф фільмів) може відтворити структури, аналогічні описаним вище, надаючи додаткові можливості для аналізу та рекомендацій. (див. рис. 2.3)

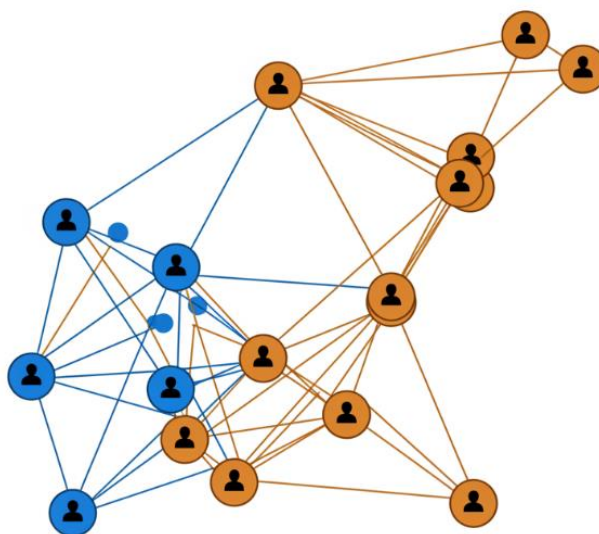


Рисунок 2.3 – Дводольний граф "користувачі-фільми"

Один набір вузлів, зображений синім кольором, представляє користувачів, а інший набір, зображений теплим помаранчевим кольором, представляє фільми. Ребра, що з'єднують користувачів із фільмами, з якими вони взаємодіяли, утворюють мережу взаємодій.

2.4 Інструменти та технології розробки

У випадку вебсайту з рекомендаціями фільмів технічна реалізація охоплює широкий спектр програмних рішень, які поділяються на інструменти серверної логіки, клієнтського інтерфейсу, бібліотеки для машинного навчання, API-зв'язки з зовнішніми джерелами та засоби для зберігання даних.

Розробка системи проводилася з орієнтацією на відкриті, гнучкі й масштабовані технології, які дозволяють швидко тестувати алгоритми та безперешкодно інтегрувати їх у вебінтерфейс. Вибір мови програмування, фреймворків, структури бази даних та сторонніх сервісів був зумовлений необхідністю забезпечити як інтелектуальну обробку даних, так і комфортну взаємодію користувача з контентом.

Окрему увагу було приділено побудові зв'язку з базою The Movie Database (TMDb) через API, що забезпечує повноцінний обмін даними про фільми [33].

TMDb - це база даних фільмів і телешоу, що містить інформацію про рейтинги, огляди, акторів і нові релізи, доступна для користувачів і розробників. Використовуючи велику базу даних TMDb, користувачі можуть досліджувати широкий спектр жанрів, відкривати нові назви та залишатися у курсі останніх випусків. Основна увага платформи на наданні точної та всеосяжної інформації робить її надійним вибором для тих, хто хоче дізнатися більше про свої улюблені фільми або відкрити для себе нові.

Така інтеграція є ключовою в архітектурі, адже дозволяє динамічно оновлювати каталог контенту, зберігаючи локальну базу в актуальному стані. Реалізація зв'язку відбувається на серверній стороні, звідки отримані метадані передаються в модулі фільтрації та персоналізації.

2.4.1 Серверні та клієнтські середовища (Flask/Django, React, API)

Запланована архітектура платформи передбачає чітке розділення логіки між серверною частиною, де здійснюється обробка даних та виконання алгоритмів, і клієнтською частиною, яка відповідає за взаємодію з користувачем і виведення рекомендацій у зручному інтерфейсі.

На стороні бекенду передбачається використання фреймворків Flask або Django, що є провідними рішеннями в екосистемі мови програмування Python. Обидва ці фреймворки підтримують реалізацію REST API, що є критично необхідним для обміну даними між сервером і клієнтською частиною застосунку. (див. рис. 2.4)

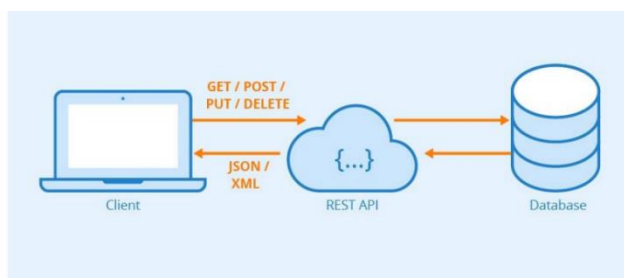


Рисунок 2.4 – Схема роботи REST API

Flask відзначається своєю легкістю й мінімалізмом, що є перевагою при побудові прототипів або невеликих сервісів із високою кастомізацією. Django, у свою чергу, забезпечує вищий рівень автоматизації завдяки вбудованій ORM, адміністративній панелі та модульній структурі, що ідеально підходить для комплексних, модульних систем.

У межах цього проєкту було прийнято рішення на користь використання Flask як серверного фреймворку замість Django, що обумовлено низкою технічних і концептуальних міркувань, характерних саме для реалізації системи рекомендацій із гнучкою веб-архітектурою.

Flask, як мікрофреймворк, надає розробнику більший контроль над структурою проєкту, що є важливою перевагою у випадках, коли система має нестандартну логіку обробки даних, зокрема — реалізацію кількох алгоритмів рекомендацій з різною методологічною основою. У проєкті, де основне

навантаження лягає на модулі аналітики, моделі на основі бібліотек машинного навчання, а також постійний обмін із зовнішнім API (TMDB), гнучкість Flask дозволяє швидко формувати легкі маршрути для обробки REST-запитів без перевантаження непотрібною надмірною структурою.

На відміну від Django, який орієнтований на швидкий розвиток класичних CRUD-застосунків із жорстко визначеною архітектурною моделлю, Flask є більш придатним для побудови високоспеціалізованого бекенду, де кожен компонент — від підключення до бази до виклику зовнішнього сервісу — налаштовується індивідуально. Це важливо у контексті побудови кастомізованої логіки рекомендацій, що не передбачає використання стандартних Django-інструментів, як-от `ModelForm` чи `TemplateView`.

Крім того, реалізація системи адміністративного керування не потребує повноцінної панелі Django admin, оскільки функціональність адмінки у цьому проєкті зводиться до базового інтерфейсу керування контентом (фільмами, категоріями, запитам до TMDB), що може бути реалізовано у Flask швидко й без надлишкових компонентів.

Вебсайт у межах цього проєкту планується реалізовувати за допомогою бібліотеки React як основного фреймворку для клієнтської частини. Цей вибір зумовлений можливістю компонувального підходу до побудови інтерфейсу, зручним управлінням станом додатку, високою продуктивністю та широкою підтримкою спільноти. React дозволить реалізувати асинхронне завантаження блоків рекомендацій, а також оновлення інтерфейсу без перезавантаження сторінки, що є ключовим для збереження користувацького комфорту.

Комунікація між фронтендом і сервером здійснюватиметься через REST API, де серверна частина відповідатиме за обробку запитів, виконання алгоритмів рекомендацій, доступ до бази даних і формування структурованих відповідей. Така архітектура забезпечує платформну незалежність, можливість багаторазового використання API на різних клієнтських пристроях, включаючи десктопні браузеры, мобільні застосунки або сторонні сервіси.

Особливу увагу буде приділено побудові окремого адміністративного інтерфейсу, доступ до якого здійснюватиметься через спеціальну адресу

(наприклад, /admin). Така панель дозволить керувати фільмами, оновлювати базу з TMDB, переглядати статистику користувачів, а також вручну втручатися в логіку ранжування у виняткових випадках. Це дозволяє підтримувати систему в актуальному стані й забезпечувати її гнучке налаштування без потреби зміни вихідного коду.

2.4.2 Бібліотеки для реалізації алгоритмів

На етапі проєктування системи персоналізованих рекомендацій передбачено використання ряду програмних бібліотек, які забезпечать реалізацію алгоритмів фільтрації, обробки даних, пошуку та аналітики в межах вебплатформи. Базовим середовищем реалізації обрано мову Python, що є широко застосовуваною у сфері прикладної статистики, машинного навчання та побудови серверної логіки для вебзастосунків.

Ключовим компонентом програмного ядра майбутньої системи стане бібліотека Surprise, яка спеціалізується на розробці алгоритмів рекомендацій на основі оцінок користувачів. Зокрема, планується використати модель SVD (сингулярного розкладу матриці) (див.рис.2.5).

$$\begin{array}{c}
 \begin{array}{|c|c|c|c|} \hline \square & \square & \square & \square \\ \hline \square & \square & \square & \square \\ \hline \square & \square & \square & \square \\ \hline \end{array} & = & \begin{array}{|c|c|c|c|} \hline \square & \square & \square & \square \\ \hline \square & \square & \square & \square \\ \hline \square & \square & \square & \square \\ \hline \end{array} & \begin{array}{|c|c|c|c|} \hline \square & \square & \square & \square \\ \hline \square & \square & \square & \square \\ \hline \square & \square & \square & \square \\ \hline \end{array} & \begin{array}{|c|c|c|c|} \hline \square & \square & \square & \square \\ \hline \square & \square & \square & \square \\ \hline \square & \square & \square & \square \\ \hline \end{array} \\
 \mathbf{M}_{m \times n} & = & \mathbf{U}_{m \times m} & \mathbf{\Sigma}_{m \times n} & \mathbf{V}^*_{n \times n} \\
 \\
 \begin{array}{|c|c|c|c|} \hline \square & \square & \square & \square \\ \hline \square & \square & \square & \square \\ \hline \square & \square & \square & \square \\ \hline \end{array} & \begin{array}{|c|c|c|c|} \hline \square & \square & \square & \square \\ \hline \square & \square & \square & \square \\ \hline \square & \square & \square & \square \\ \hline \end{array} & = & \begin{array}{|c|c|c|c|} \hline 1 & 0 & 0 & 0 \\ \hline 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 \\ \hline \end{array} \\
 \mathbf{U} & \mathbf{U}^* & = & \mathbf{I}_m \\
 \\
 \begin{array}{|c|c|c|c|} \hline \square & \square & \square & \square \\ \hline \square & \square & \square & \square \\ \hline \square & \square & \square & \square \\ \hline \end{array} & \begin{array}{|c|c|c|c|} \hline \square & \square & \square & \square \\ \hline \square & \square & \square & \square \\ \hline \square & \square & \square & \square \\ \hline \end{array} & = & \begin{array}{|c|c|c|c|} \hline 1 & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & 0 & 1 \\ \hline \end{array} \\
 \mathbf{V} & \mathbf{V}^* & = & \mathbf{I}_n
 \end{array}$$

Рисунок 2.5 – Модель сингулярного розкладу матриці

Модель SVD бере за основу матрицю користувачів та фільмів, у якій рядки — це користувачі, стовпці — фільми, а значення — це оцінки або індикатори взаємодії. У даному проєкті ці оцінки задаються умовно: кожен лайк інтерпретується як фіксована позитивна оцінка (наприклад, 1.0), а відсутність оцінки — як відсутність взаємодії. Матриця формується із даних таблиці MovieLike. SVD розкладає таку матрицю на три складові:

- U — матриця прихованих характеристик користувачів,
- Σ (sigma) — діагональна матриця ваг (власних значень),
- V^T — матриця прихованих характеристик фільмів.

Суть SVD полягає в наближенні початкової матриці взаємодій до добутку трьох компонентів: матриці характеристик користувачів, діагональної матриці вагових коефіцієнтів і матриці характеристик фільмів. Такий розклад, що математично представлений як $R \approx U \times \Sigma \times V^T$, дозволяє виявляти закономірності у вподобаннях, які залишаються прихованими при прямому аналізі вихідних даних. Це забезпечує можливість передбачення інтересу користувача до фільмів, які він ще не переглядав, шляхом моделювання його профілю у прихованому просторі ознак.

Завдяки факторизації матриці за допомогою SVD вдається зменшити розмірність даних, зберігаючи при цьому найважливішу інформацію про взаємодії. Це значно покращує ефективність моделі, оскільки працює не з усією великою та розрідженою матрицею рейтингових оцінок, а з її більш компактним представленням у вигляді латентних факторів. В результаті модель стає стійкішою до шуму в даних та здатною краще узагальнювати, що особливо важливо для рекомендацій у системах з великою кількістю користувачів і фільмів.

Під час реалізації такого підходу дані про взаємодії користувачів із фільмами конвертуються у формат, придатний для машинного навчання, із використанням класів Reader і Dataset з бібліотеки Surprise. Після генерації навчального набору виконується навчання моделі SVD на повній матриці. Це дозволяє отримувати передбачення рейтингу для кожного фільму, який ще не був переглянутий користувачем.

Лістинг 2.1 – Каркас програмного коду даної моделі

```

from surprise import Dataset, Reader, SVD
import pandas as pd

likes      =      MovieLike.objects.all().values_list('user_id',
'movie_id')

data = [(user, movie, 1.0) for user, movie in likes]

reader = Reader(rating_scale=(0, 1))

dataset      =      Dataset.load_from_df(pd.DataFrame(data,
columns=['userID', 'itemID', 'rating']), reader)

trainset = dataset.build_full_trainset()

algo = SVD()

algo.fit(trainset)

```

Результати прогнозування використовуються для сортування та подальшої генерації релевантного списку рекомендацій. Це рішення органічно інтегрується у Django-застосунок і дозволяє повноцінно реалізувати персоналізовану логіку у межах вебінтерфейсу.

2.5 Особливості інтеграції рекомендацій в інтерфейс користувача

Інтеграція системи персоналізованих рекомендацій у структуру вебінтерфейсу є не менш важливим завданням, ніж побудова самих алгоритмів. Рекомендації не повинні залишатися прихованим фоновим процесом — вони мають бути представлені так, щоб користувач інтуїтивно розумів їхнє призначення, довіряв логіці підбору контенту та активно взаємодівав із запропонованими елементами. Саме тому інтеграція алгоритмічних результатів у візуальну частину сайту потребує продуманих рішень у сфері UX-дизайну, особливо у випадках, коли йдеться про фільмовий контент, який подається у великому обсязі.

Одна з ключових умов ефективності – контекстне розміщення блоків рекомендацій. Вони мають з’являтися там, де користувач очікує додаткової пропозиції, наприклад, після перегляду опису фільму або завершення перегляду

стрічки. У таких випадках блок "Схожі фільми" або "Вам також може сподобатися" діє як логічне продовження досвіду, а не як окремий розділ, що потребує спеціального звернення уваги. Така інтеграція підвищує природність навігації та знижує когнітивне навантаження на користувача.

Інший важливий аспект — уніфікація візуального стилю рекомендацій із загальним дизайном сайту. Карточки фільмів, представлених у рекомендаціях, повинні мати ту ж структуру, що й інші елементи каталогу: зображення, назва, рейтинг, жанр. Водночас, корисним є додавання маркерів, які пояснюють логіку появи того чи іншого фільму в добірці (наприклад, "На основі жанру", "Популярне серед користувачів", "На основі ваших оцінок"). Це підвищує прозорість системи та сприяє формуванню довіри до її механізмів.

Оскільки вебплатформа орієнтується на широкий спектр пристроїв, значну роль відіграє адаптивність. Умови перегляду з десктопа і зі смартфона кардинально різняться, тому рекомендаційні блоки повинні коректно масштабуватися, перебудовуватися у каруселі або колонки, не перекриваючи основний контент. Така гнучкість досягається за рахунок використання фреймворків для адаптивної верстки (наприклад, Bootstrap або TailwindCSS) у поєднанні з логікою умовного рендерингу у клієнтських скриптах.

Крім розміщення, важливо враховувати час виводу рекомендацій. Надмірна кількість динамічного контенту одразу після завантаження сторінки може негативно впливати на швидкодію сайту. Тому доцільно реалізувати механізми відкладеного завантаження (lazy loading), які дозволяють підвантажувати блоки рекомендацій асинхронно, після основного контенту або під час скролінгу.

2.6 Архітектура зберігання та обробки даних у системі рекомендацій фільмів

Архітектура зберігання та обробки даних розробленої системи рекомендацій фільмів побудована на трьох основних рівнях: рівні зберігання, рівні обробки на сервері та рівні виведення на клієнтській стороні. Кожен рівень

виконує чітко визначені функції, забезпечуючи ефективне функціонування системи, що використовує комбінацію графових моделей та статистичних алгоритмів для надання персоналізованих рекомендацій.

Для забезпечення функціональності пошуку реалізовані алгоритми точного та нечіткого пошуку. Алгоритм динамічного відображення контенту відповідає за ранжування фільмів відповідно до їхньої актуальності або популярності. Крім того, серверна частина здійснює аналіз відгуків та реакцій користувачів для визначення настроїв щодо фільмів, що використовується у відповідному рекомендаційному алгоритмі. Доступ до адміністративної панелі (/admin) також обробляється на серверній стороні, забезпечуючи інтерфейс для керування контентом та користувачами.

На рівні зберігання даних використовується реляційна база даних, оптимізована для зберігання метаданих фільмів, отриманих через API сервісу TMDB. Це включає структуровану інформацію про назви, жанри, акторів, режисерів, описи, ключові слова та унікальні ідентифікатори TMDB, а також URL-адреси постерів. Окрема увага приділяється індексації ключових полів для забезпечення швидкого пошуку та фільтрації, необхідних для реалізації Content-Based Filtering, зокрема для порівняння тегів (жанрів) та пошуку схожості за ключовими словами. Крім того, база даних зберігає інформацію про користувачів та їхню історію взаємодій з фільмами, включаючи перегляди та оцінки, що є основою для алгоритмів Collaborative Filtering, які аналізують поведінку інших користувачів зі схожими смаками. Для підтримки рекомендацій на основі тегів (жанрів) у базі даних також зберігається інформація про категоризацію фільмів. Дані про пошукові запити користувачів фіксуються для забезпечення функціональності точного та нечіткого пошуку (з використанням алгоритму Levenshtein Distance) та потенційного майбутнього аналізу пошукових тенденцій. Результати аналізу відгуків та реакцій користувачів щодо фільмів, необхідні для алгоритму рекомендацій на основі аналізу настроїв, також зберігаються в структурованому форматі (див. рис. 2.6).

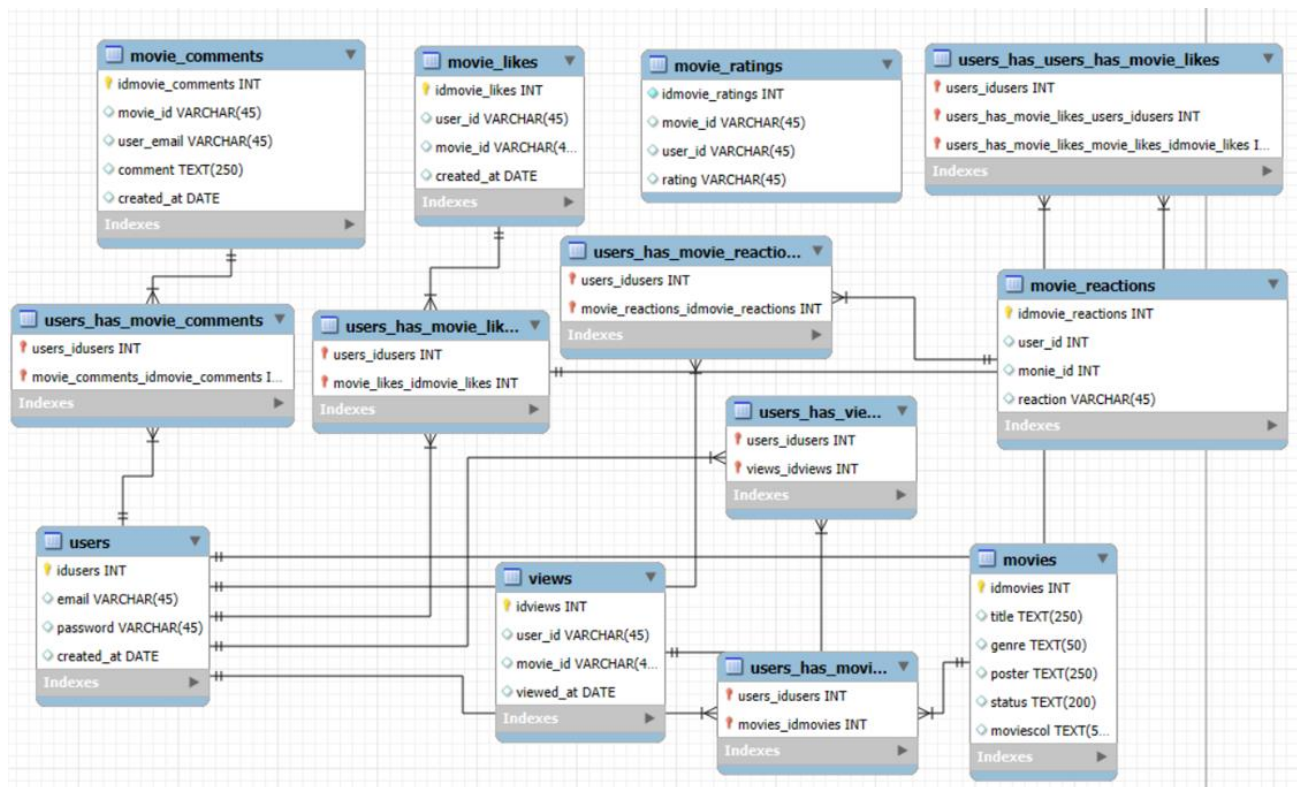


Рисунок 2.6 – Концептуальна схема бази даних веб-сайту

Рівень обробки даних реалізований на серверній стороні з використанням фреймворку Flask або Django. Цей рівень відповідає за ключові операції, включаючи отримання та інтеграцію даних з API сервісу TMDB. Для забезпечення актуальності інформації про фільми здійснюється періодична синхронізація з TMDB, під час якої нові фільми додаються до бази даних, а існуючі записи оновлюються. З метою оптимізації продуктивності та зменшення навантаження на зовнішній API застосовується механізм локального кешування отриманих метаданих. Основна функціональність серверної частини полягає у реалізації різноманітних рекомендаційних алгоритмів. Для Content-Based Filtering здійснюється порівняння метаданих фільмів, включаючи аналіз спільних жанрів, схожості ключових слів та використання функціональності `similar_movies()` API TMDB. Collaborative Filtering реалізується шляхом аналізу історії переглядів та оцінок користувачів для виявлення схожих смаків та формування рекомендацій на основі вподобань цієї групи. Гібридна модель поєднує результати Content-Based та Collaborative Filtering для підвищення точності рекомендацій.

2.7 Висновки до другого розділу

В другому розділі кваліфікаційної роботи досліджено методологічну основу проекту, якою є комбінація контентно-орієнтованої та колаборативної фільтрації, доповнена гібридним підходом для підвищення точності рекомендацій. Система використовує метадані фільмів, отримані через TMDB API, а також дані про взаємодію користувачів (перегляди, оцінки, реакції) для побудови рекомендацій.

Системна модель передбачає багаторівневу архітектуру, де перший рівень відповідає за збір та зберігання даних, другий – за їх обробку та застосування рекомендаційних алгоритмів, а третій – за візуалізацію рекомендацій у вебінтерфейсі. Ключовими елементами системи є графові моделі, зокрема граф фільмів на основі схожості контенту та дводольний граф "користувачі-фільми", які використовуються для аналізу зв'язків між об'єктами та користувачами.

Інструментарій розробки включає Python як основну мову програмування, фреймворк Flask для реалізації серверної частини та REST API, бібліотеку React для розробки інтерактивного клієнтського інтерфейсу, а також бібліотеку Surprise для реалізації алгоритмів колаборативної фільтрації, зокрема моделі SVD. Інтеграція з TMDB API забезпечує постійне оновлення каталогу фільмів актуальними метаданими.

Архітектура зберігання та обробки даних передбачає використання реляційної бази даних для структурованого зберігання інформації про фільми, користувачів та їхні взаємодії. Серверна частина відповідає за отримання даних з TMDB, реалізацію рекомендаційних алгоритмів, обробку пошукових запитів та керування адміністративною панеллю. Клієнтська частина забезпечує відображення рекомендацій та взаємодію з користувачем.

3 РОЗРОБКА ВЕБ-САЙТУ З РЕКОМЕНДАЦІЙНОЮ СИСТЕМОЮ

3.1 Постановка технічного завдання

Проект вебплатформи з рекомендаційною системою фільмів потребує чітко сформульованого технічного завдання, що визначає загальні вимоги до функціоналу, архітектури, інтерфейсної взаємодії та інструментів реалізації. Розробка передбачає створення повноцінного вебзастосунку, який забезпечуватиме персоналізовані рекомендації фільмів на основі активності користувача, інтелектуального пошуку, зручної навігації, а також можливості керування вмістом через адміністративну панель.

Архітектура застосунку побудована на основі фреймворку Django для серверної частини, що забезпечує обробку запитів, реалізацію алгоритмів персоналізації, доступ до бази даних, авторизацію, а також обробку даних з API TMDB. Вебінтерфейс реалізується за допомогою React, що дає змогу створити динамічний і зручний інтерфейс із підтримкою асинхронної взаємодії з сервером. Комунікація між фронтендом і бекендом забезпечується через REST API, що дозволяє гнучко передавати дані й оновлювати сторінки без повного перезавантаження.

Основними функціональними модулями сайту є:

- Система рекомендацій — автоматизований блок, який надає індивідуальні добірки фільмів користувачам. Рекомендації реалізуються через алгоритми колаборативної фільтрації (SVD), контентного аналізу, гібридні моделі та алгоритм нечіткого пошуку. Джерелом даних для навчання моделей слугує таблиця вподобань користувачів, що оновлюється в реальному часі.

- Пошуковий інтерфейс — реалізований через комбінований механізм exact і fuzzy search. Використовується бібліотека RapidFuzz для обробки неточних запитів, що дозволяє враховувати орфографічні помилки або часткові збіги при пошуку фільмів.

- Користувацький інтерфейс — передбачає доступ до головної сторінки з новими релізами, рекомендаційних блоків, сторінок фільмів з описом і

трейлерами, функцій оцінювання та перегляду історії. Окрема увага приділяється адаптивності для різних типів пристроїв, що реалізується через стилізацію у TailwindCSS.

— Адміністративна панель — окремий маршрут `/admin`, через який адміністратор має можливість додавати або редагувати інформацію про фільми, оновлювати дані з TMDb API, переглядати статистику активності користувачів. Адмінінтерфейс інтегрується у логіку Django із базовим рівнем автентифікації.

Усі компоненти системи інтегруються в єдину модульну архітектуру, що ґрунтується на вже створеній базі даних. Її структура забезпечує повну сумісність із функціональними модулями сайту та підтримує масштабування без порушення цілісності системи. Така архітектура дозволяє легко оновлювати або замінювати алгоритми персоналізації, зберігаючи при цьому стабільність роботи платформи навіть за умови зростання навантаження. Подальша реалізація вебфункціоналу здійснюється з урахуванням цієї централізованої структури зберігання та обробки даних.

3.2 Архітектура та функціональні компоненти системи рекомендацій

Система рекомендацій на сайті фільмів є складним програмним комплексом, який поєднує механізми збору даних, алгоритмічної обробки та виведення результатів у зрозумілому користувачеві форматі. Архітектура цього рішення побудована на модульному підході з чітким розподілом відповідальностей між логічними компонентами. Це дозволяє забезпечити масштабованість, легкість оновлення та адаптацію до змін у структурі даних або алгоритмах.

3.2.1 Загальна архітектура системи

Сайт реалізований як класичний веб-додаток з клієнтсько-серверною архітектурою. Серверна частина відповідальна за обробку даних, виконання алгоритмів рекомендацій, взаємодію з зовнішніми сервісами (зокрема, TMDb

API), а також за забезпечення авторизації через адміністративний інтерфейс. Клієнтська частина забезпечує візуальне представлення результатів, реалізацію пошуку, інтерактивні елементи й підтримку користувацьких сценаріїв [44].

Оснoву даних про фільми становить зовнішній програмний інтерфейс — The Movie Database API (TMDB), з якого здійснюється регулярне завантаження метаданих: назви фільмів, жанри, ключові слова, опис, актори, рейтинг тощо. Ці дані зберігаються у локальній базі, що дозволяє зменшити кількість запитів до API та забезпечити автономність виводу контенту.

3.2.2 Серверна логіка та реалізація рекомендацій

Сервер реалізований на основі фреймворків, сумісних із Python, з використанням внутрішньої бази даних та зв'язку з TMDB API. Алгоритми рекомендацій запускаються на сервері та використовують локально кешовану інформацію про фільми та користувацьку активність. Результати обробки передаються до фронтенду у вигляді REST-запитів.

Окремі ендпоінти реалізують логіку для кожного з підходів:

— Контентно-орієнтований фільтр (Content-Based) використовує пошук за метаданими (жанри, ключові слова, режисери) з порівнянням схожості, зокрема через cosine similarity.

— Колаборативна модель (Collaborative Filtering) функціонує на основі локальної історії дій користувачів або попередньо збережених шаблонів схожих вподобань.

— Гібридна модель об'єднує ці два методи, дозволяючи адаптувати результат відповідно до конкретного користувача.

Фронтенд реалізовано з використанням JavaScript (або фреймворку React.js), де динамічне оновлення контенту дозволяє інтерактивно оновлювати блоки з рекомендаціями без перезавантаження сторінки. Наприклад, на сторінці перегляду фільму виводяться "Схожі фільми", що формуються за запитом до ендпоінту `/api/recommend/similar`. Окремі блоки відповідають за "Популярне",

"На основі переглядів" чи "Рекомендовано вам", кожен з яких підключений до відповідного алгоритмічного модуля (див. рис. 3.1).

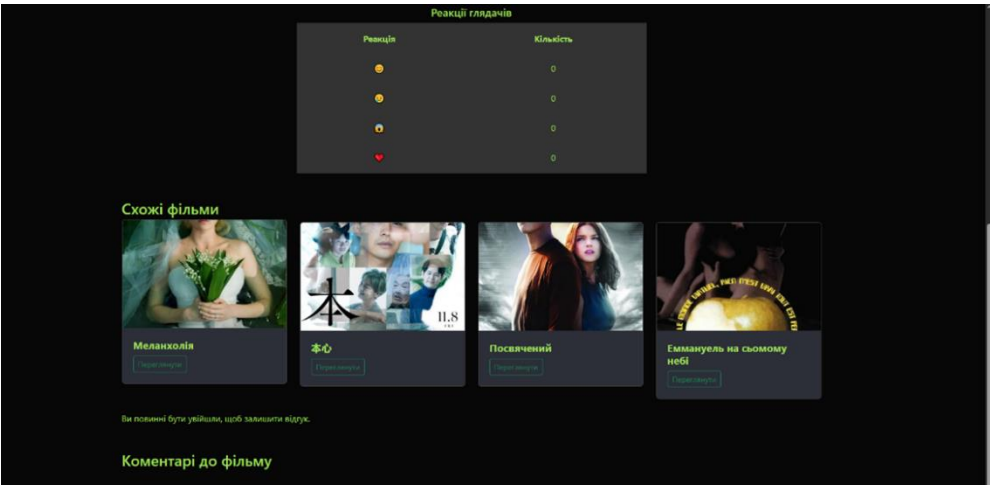


Рисунок 3.1 – Блоки з рекомендаціями

Доступ до адміністративної панелі здійснюється за адресою /admin, де реалізовано базову систему авторизації (логін і пароль: admin/admin). Через неї адміністратор має змогу ініціювати оновлення каталогу фільмів з TMDb, переглядати логи взаємодій користувачів, а також вручну керувати деякими компонентами рекомендацій (наприклад, підсилювати популярність окремих фільмів або вручну призначати теги) (див. рис. 3.2).

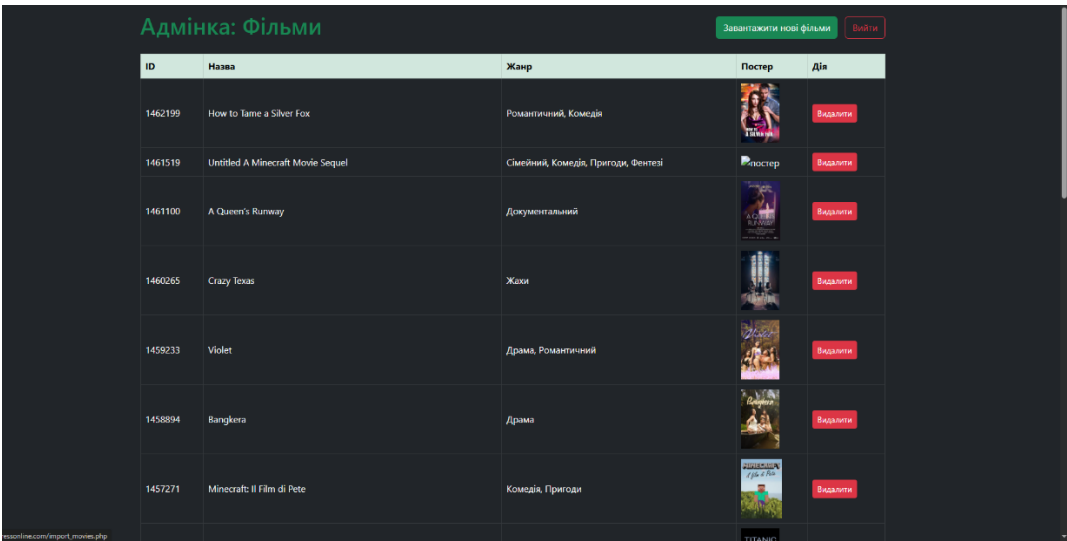


Рисунок 3.2 – Адміністративна панель

Основою сайту виступає головна сторінка, яка одночасно виконує роль навігаційного центру та вітрини контенту. (див. рис. 3.4)

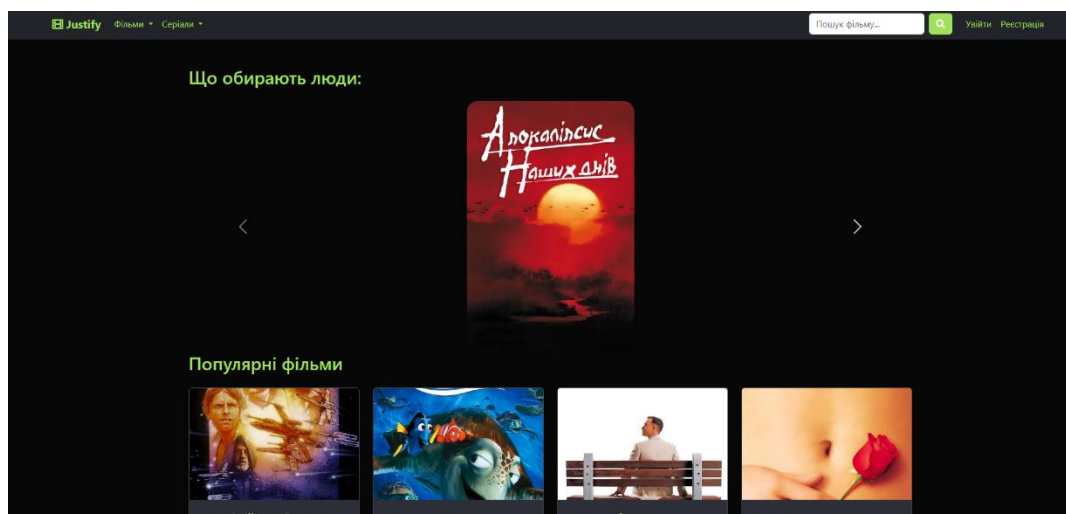


Рисунок 3.4 – Головна сторінка сайту

Вона автоматично наповнюється актуальними добірками фільмів, інформація про які надходить як із локальної бази, так і через API зовнішнього джерела TMDB. Використання серверної пагінації дозволяє рівномірно розподіляти контент на кілька сторінок, забезпечуючи комфортну навігацію для користувача.

З головної сторінки здійснюється перехід на сторінку окремого фільму (див. рис. 3.5).

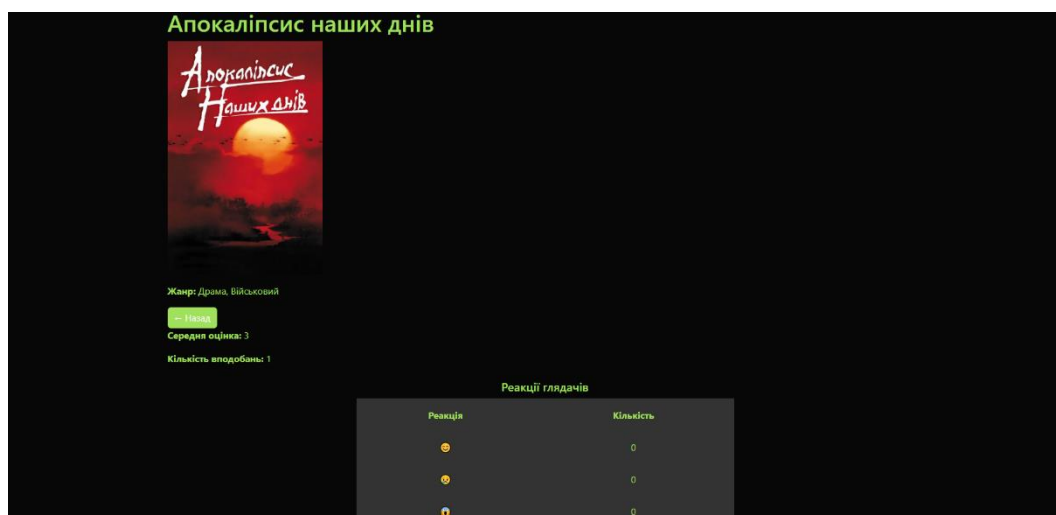


Рисунок 3.5 – Сторінка фільму

Тут зосереджено повний опис фільму: сюжет, жанрова приналежність, акторський склад, дата випуску, трейлер тощо. Особливу увагу приділено інтерактивним елементам: кнопкам уподобань, збереженню в обране, а також відображенню схожих фільмів, що обчислюються на основі контентних ознак. Таким чином реалізується початковий рівень рекомендацій, без необхідності явно звертатися до окремої сторінки.

Розширення можливостей взаємодії забезпечується через інтеграцію пошукової системи. Вона дозволяє як точно, так і наближене (нечітке) знаходження фільмів за назвою, включаючи варіанти з помилками чи неповними запитами (див. рис. 3.6).

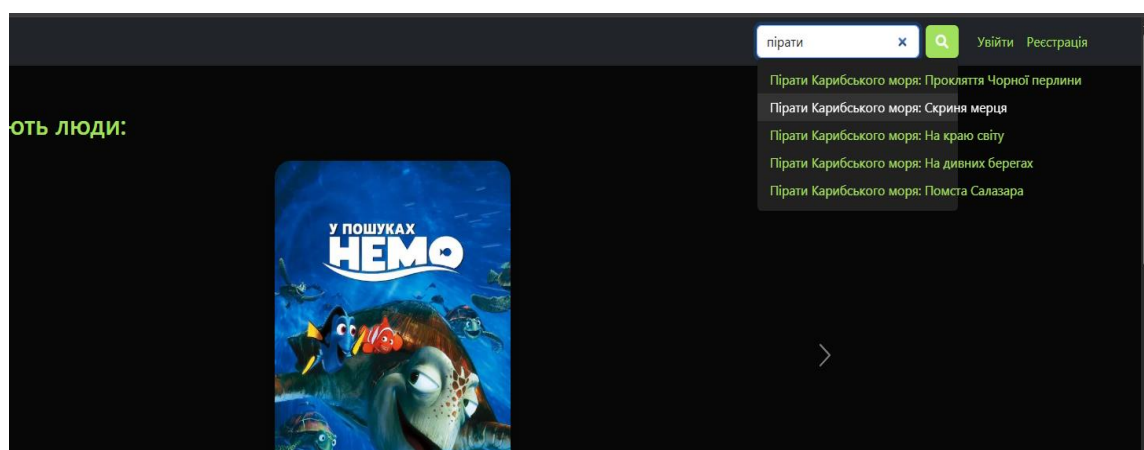


Рисунок 3.6 – Можливості пошуку на веб-сайті

На рівні серверної логіки використовується бібліотека RapidFuzz, що дозволяє виконувати ефективне порівняння введеного запиту зі значеннями в базі даних та повертати релевантні результати, впорядковані за ступенем схожості.

Доступ до повноцінного функціоналу сайту надається після проходження авторизації або реєстрації. Обидві форми реалізовані відповідно до архітектурних стандартів Flask і підтримують основні принципи безпеки (див. рис. 3.7).

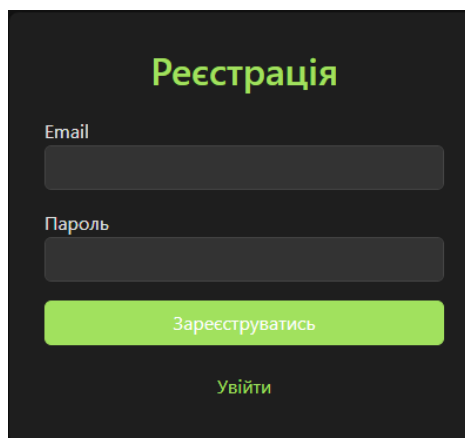
The image shows a registration form on a dark background. At the top, the word 'Реєстрація' is written in a light green font. Below it, there are two input fields: the first is labeled 'Email' and the second is labeled 'Пароль'. Both fields are empty and have a light gray border. Below the password field is a large, solid light green button with the text 'Зареєструватись' in a darker green font. At the bottom of the form, there is a link 'Увійти' in a light green font.

Рисунок 3.7 – Авторизація/Реєстрація на сайті

Після автентифікації користувач отримує персоналізований доступ, який включає можливість зберігати вподобання, формувати історію переглядів і, що найголовніше, отримувати рекомендації, згенеровані відповідно до його активності.

3.4 Реалізація алгоритмів рекомендацій

У межах розробки функціональної частини платформи важливим етапом стало впровадження алгоритмів, які відповідають за формування персоналізованих рекомендацій. Саме алгоритмічна складова визначає інтелектуальність системи, її здатність адаптуватися до інтересів користувачів, аналізувати історію переглядів, виявляти приховані зв'язки між фільмами та пропонувати релевантний контент у відповідний момент часу.

З технічного погляду, реалізація алгоритмів відбувається у серверному середовищі на основі мови Python із залученням бібліотек машинного навчання та текстової аналітики. Алгоритми інтегруються у вебзастосунок через окремі маршрути Flask, а результати їхньої роботи передаються у шаблони HTML, формуючи користувацький інтерфейс рекомендацій у режимі реального часу.

Алгоритми в проєкті різноманітні: від класичних моделей колаборативної та контентно-орієнтованої фільтрації до евристичних механізмів на основі тегів, популярності, поведінкових патернів і нечіткого пошуку. Це дозволяє сформувати гібридну структуру, де кожен з підходів виконує свою функцію в

системі. Основна увага зосереджена на моделі SVD, яка дозволяє здійснювати прогнозування взаємодій навіть за відсутності повних даних. У поєднанні з іншими методами це створює багаторівневий механізм рекомендацій, що адаптивно реагує на потреби користувача [33].

3.4.1 Content-Based Filtering (фільтрація на основі вмісту)

Алгоритм ґрунтується на аналізі характеристик самих фільмів: жанри, ключові слова, актори, опис, режисери. Система знаходить схожість між фільмами за контентними ознаками і пропонує ті, які найбільш схожі на вже переглянуті або оцінені користувачем.

Використання контентних ознак допомагає формувати персоналізовані рекомендації навіть тоді, коли інформації про поведінку користувача замало або її взагалі немає, що робить систему більш гнучкою і універсальною.

Технічна реалізація: У Python для цього використовуються бібліотеки `pandas` (для обробки таблиць), `scikit-learn` (для векторизації даних), а також `TfidfVectorizer` або `CountVectorizer`, які формують числові вектори з тексту (опису, ключових слів). Потім застосовується `cosine_similarity`, яка обчислює ступінь подібності між фільмами.

3.4.2 Collaborative Filtering (колаборативна фільтрація)

Цей підхід використовує поведінку інших користувачів. Система визначає, з якими користувачами поточний має схожі вподобання, і пропонує йому те, що сподобалося цим користувачам. У Python використовують бібліотеку `Surprise` — спеціалізований фреймворк для побудови колаборативних систем. Застосовуються алгоритми SVD (сингулярне розкладення), KNN (найближчі сусіди), а також матричне факторизування [34].

Текст алгоритму подано в лістингу 2.1

Лістинг 2.1 – Назва лістингу

```
def get_surprise_recommendations(user_id, top_n=5):
    likes = MovieLike.objects.all().values_list('user_id',
'movie_id')

    data = [(user, movie, 1.0) for user, movie in likes]

    reader = Reader(rating_scale=(0, 1))
    dataset = Dataset.load_from_df(pd.DataFrame(data,
columns=['userID', 'itemID', 'rating']), reader)
    trainset = dataset.build_full_trainset()

    algo = SVD()
    algo.fit(trainset)

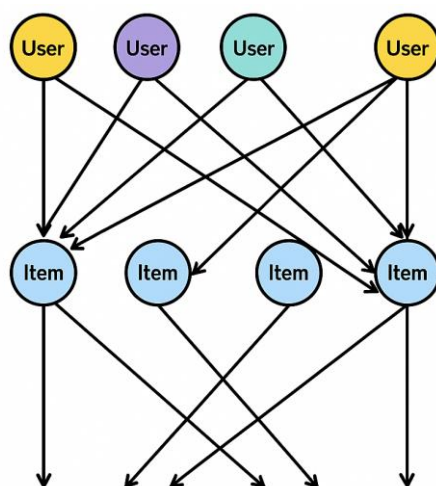
    watched =
MovieLike.objects.filter(user_id=user_id).values_list('movie_id',
flat=True)

    all_movies = Movie.objects.exclude(id__in=watched)

    predictions = []
    for movie in all_movies:
        pred = algo.predict(user_id, movie.id)
        predictions.append((movie, pred.est))

    predictions.sort(key=lambda x: x[1], reverse=True)
    return [movie for movie, _ in predictions[:top_n]]
```

На відміну від контентної фільтрації, колаборативна фільтрація не використовує безпосередньо інформацію про характеристики об'єктів (наприклад, фільмів), а вивчає схожість між користувачами або між елементами на основі їхніх спільних уподобань. Наведена функція є частиною реалізації алгоритму колаборативної фільтрації (див. рис. 3.8).



Collaborative Filtering

Рисунок 3.8 – Алгоритм колаборативної фільтрації

Зокрема, з використанням бібліотеки Surprise, яка широко застосовується для побудови рекомендаційних систем у Python. Реалізація базується на методі SVD (Singular Value Decomposition) — одному з найпоширеніших алгоритмів матричної факторизації, що використовується для прогнозування уподобань користувача на основі поведінки інших користувачів.

Функція `get_surprise_recommendations` приймає два параметри: ідентифікатор користувача (`user_id`) і кількість рекомендованих об'єктів (`top_n`). Вона спочатку формує набір даних на основі вподобань користувачів, зібраних із моделі MovieLike. Усі вподобання інтерпретуються як позитивні оцінки з фіксованим значенням 1.0, оскільки система не працює з реальними рейтингами, а лише з фактом позитивної взаємодії.

Далі формується датафрейм, що використовується для побудови об'єкта Dataset з бібліотеки Surprise, з подальшою генерацією повного навчального набору (`build_full_trainset`). Це забезпечує повне охоплення наявних взаємодій між користувачами та фільмами в процесі тренування моделі.

Після цього ініціалізується модель SVD, яка проходить навчання на сформованому тренувальному наборі. Після завершення навчання здійснюється прогнозування: із загального переліку фільмів відбираються лише ті, які ще не були вподобані даним користувачем. Для кожного з цих фільмів обчислюється передбачувана оцінка (предикція) на основі моделі, після чого список

результатів сортується за значенням прогнозованого інтересу у спадаючому порядку.

Підсумком виконання функції є повернення списку об'єктів фільмів, які з найбільшою ймовірністю можуть зацікавити користувача, що відповідає суті персоналізованої колаборативної рекомендаційної системи.

3.4.3 Інші алгоритми системи рекомендацій

Окрім основних моделей, реалізованих на основі контентного та колаборативного аналізу, у структурі сайту використовуються й допоміжні алгоритми, що виконують специфічні завдання — доповнюють головну логіку рекомендацій, формують динамічний контент та покращують пошукову зручність. Хоча ці алгоритми не є ядром персоналізованої моделі, їх інтеграція значною мірою підвищує загальну якість взаємодії з користувачем та функціональність сервісу [34].

Одним із таких механізмів є Trending Algorithm, який відповідає за формування блоку популярного контенту. Алгоритм не передбачає персоналізації, однак базується на колективній поведінці аудиторії, відображаючи найпопулярніші фільми за певний період. Це реалізується шляхом агрегації кількості переглядів, оцінок або збережень у базі даних, часто з урахуванням часової динаміки. Дані також можуть надходити з зовнішнього джерела — TMDb API, що дозволяє розширити підбір фільмів за глобальними трендами. Такий підхід є ефективним для нових або неавторизованих користувачів [33].

Інша функціональна складова — Tag-Based Recommendation, яка формує рекомендації на основі категоріальних ознак, таких як жанри чи тематичні ключові слова. Алгоритм аналізує найбільш частотні категорії переглянутих або вподобаних фільмів і пропонує схожий контент за жанровими тегами. Реалізація здійснюється через SQL-запити або ORM-фільтри, де в моделі фільму зберігаються прив'язані жанри у вигляді масивів або таблиць багатьох до

багатьох. Така логіка часто використовується для попередньої фільтрації у гібридних моделях.

Особливу роль у покращенні інтерфейсу взаємодії виконує Fuzzy Search — нечіткий пошук, що дозволяє обробляти користувацькі запити з помилками або варіаціями написання. Алгоритм реалізується за допомогою метрик рядкової схожості, зокрема Levenshtein Distance, що оцінює різницю між введеним запитом і назвами фільмів у базі. В Python це забезпечується бібліотеками fuzzywuzzy або RapidFuzz. Подібна технологія особливо корисна для мобільних користувачів або в умовах мультимовного пошуку.

Для організації головної сторінки застосовується Dynamic Content Display Algorithm, який ранжує контент на основі критеріїв релевантності: дати випуску, популярності чи обраного режиму перегляду. Це забезпечує динамічну зміну сітки фільмів залежно від активності користувача або новизни контенту. Алгоритм реалізується на рівні запитів до бази, де дані сортуються за відповідними полями.

Ще один допоміжний метод — Sentiment Analysis Based Recommendation, що передбачає аналіз відгуків або реакцій користувачів на фільми. Мета — визначити загальний емоційний настрій до фільму, використовуючи засоби обробки природної мови. У Python для цього застосовуються бібліотеки TextBlob, VADER або сучасні трансформерні моделі з платформи Hugging Face. Отримані оцінки настроїв можуть бути використані як додатковий параметр при формуванні рекомендацій.

3.5 Інтеграція та візуалізація рекомендацій на веб-сайті

Процес інтеграції рекомендаційної системи у веб-платформу передбачає не лише технічну реалізацію алгоритмів, а й продуману візуальну подачу контенту. Саме спосіб, у який результати рекомендацій відображаються на сайті, безпосередньо впливає на ефективність взаємодії користувача із системою. У даному проєкті ключовим є забезпечення плавного, логічного й адаптивного

представлення персоналізованих блоків, що відповідають за релевантні підказки, схожі фільми або нові популярні стрічки.

Інтеграція на рівні клієнтської частини була реалізована через компоненти, розроблені на JavaScript, з використанням сучасного фреймворку для фронтенд-розробки. Дані з рекомендаційних алгоритмів передаються через REST API у форматі JSON, після чого вони динамічно підвантажуються в інтерфейс без потреби перезавантаження сторінки. Це забезпечує гнучку взаємодію та дозволяє змінювати структуру рекомендацій залежно від поведінки користувача у реальному часі.

Особливу увагу було приділено адаптивності візуального оформлення. Рекомендаційні блоки виводяться в окремих секціях — таких як “Схожі фільми”, “Рекомендовано для вас”, “Популярне зараз”, “На основі ваших уподобань”. Кожен із цих модулів має чітко визначену логіку підбору контенту, яка пов’язана з конкретним алгоритмічним підходом. Наприклад, блок “Схожі фільми” формується переважно на основі контентно-орієнтованої фільтрації, а “Рекомендовано для вас” — результат колаборативної моделі, що використовує попередню поведінку користувача (див. рис. 3.9).

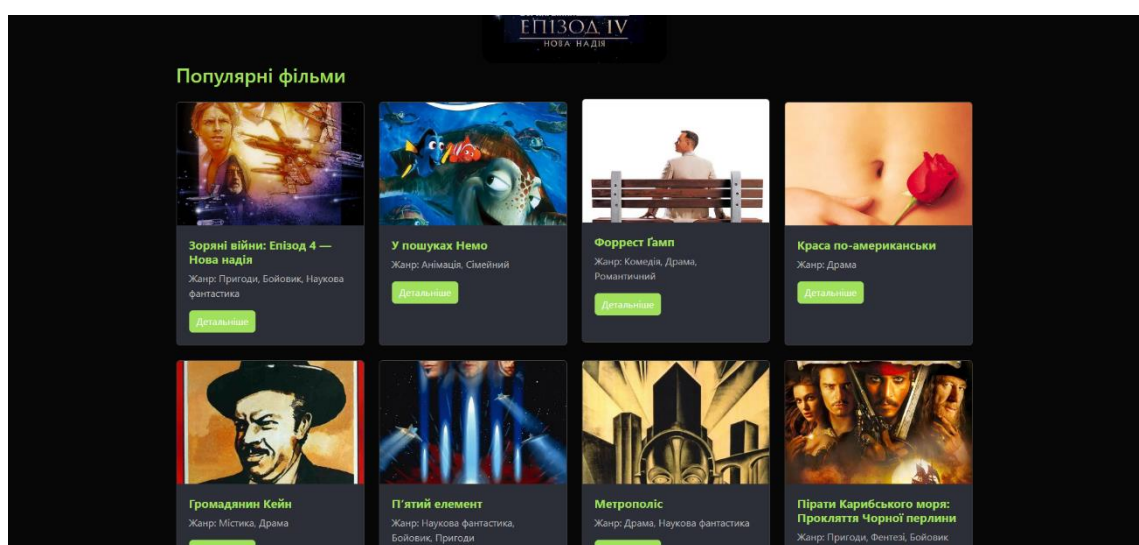


Рисунок 3.9 – Рекомендаційні блоки на веб сайті

Важливим аспектом інтеграції є оптимізація запитів до серверної частини. Щоб уникнути надмірного навантаження на систему, реалізовано механізм

кешування результатів на стороні сервера, а також обмеження частоти повторного обчислення рекомендацій. Крім того, було впроваджено lazy-loading зображень і фільмів, що дозволяє поступово завантажувати дані під час прокручування сторінки, забезпечуючи швидкість та безперервність інтерфейсу.

Для неавторизованих користувачів система пропонує загальні рекомендації на основі загальної популярності або трендів. У той час як після входу користувача в обліковий запис активується механізм персоналізації, з урахуванням історії його переглядів, реакцій і оцінок. Система також дозволяє враховувати зміну уподобань у динаміці, забезпечуючи контекстно-чутливе оновлення відображуваного контенту.

Візуальна інтеграція системи рекомендацій не є лише функціональною частиною програмного забезпечення, а виконує ключову роль у створенні індивідуального кінонавігатора для кожного користувача, підтримуючи принципи доступності, адаптивності та персоналізованої взаємодії.

3.6 Тестування роботи сайту та перевірка рекомендацій

Після завершення основних етапів реалізації проєкту важливим завданням стало проведення тестування вебсайту та функціональної перевірки всіх інтегрованих компонентів. Особлива увага приділялася стабільності роботи сторінок, правильності маршрутизації, взаємодії між клієнтською частиною та серверною логікою, а також ефективності роботи алгоритмів рекомендацій. Уся система тестувалася у контрольованому середовищі із заздалегідь підготовленими даними, що дозволило послідовно перевірити окремі модулі та їхню взаємодію.

Функціональність основних сторінок — зокрема головної, сторінки фільму, пошуку, реєстрації та авторизації — тестувалася вручну шляхом моделювання дій користувача. Було перевірено відображення фільмів, коректність завантаження інформації з бази даних, правильність роботи форм, відображення повідомлень про помилки та переадресацію після входу або виходу з облікового запису. Тестування також охоплювало процес оновлення

вмісту сайту через адміністративну панель, додавання нових фільмів і коректну взаємодію з TMDb API.

Окрему фазу тестування становила перевірка роботи алгоритмів персоналізації. З метою контролю правильності рекомендацій було створено кілька тестових облікових записів користувачів з різними вподобаннями. Кожен із них здійснював перегляд і оцінку фільмів певної жанрової спрямованості, після чого на основі цих дій система формувала рекомендації. Було зафіксовано, що запропоновані результати дійсно демонструють відповідність попереднім уподобанням користувача, що вказує на правильну роботу колаборативної та контентно-орієнтованої моделей.

У процесі виведення персоналізованих добірок також перевірялося, чи не потрапляють до результатів фільми, які вже були оцінені користувачем. Крім того, тестувалася коректність сортування результатів за передбаченою оцінкою та швидкість відповіді сервера при обробці запитів, що викликають запуск алгоритмів. За результатами перевірки було підтверджено стабільну роботу алгоритмічного блоку, навіть при одночасному зверненні з декількох облікових записів.

3.7 Висновок до третього розділу

В третьому розділі кваліфікаційної роботи описано процес розробки вебплатформи з рекомендаційною системою фільмів, починаючи з постановки технічного завдання та закінчуючи тестуванням готового продукту. Було детально розглянуто архітектуру застосунку, що базується на фреймворках Django (бек-енд) та React (фронт-енд), а також комунікацію між ними через REST API.

Особливу увагу приділено функціональним компонентам системи, серед яких виділено систему рекомендацій, пошуковий інтерфейс (з реалізацією exact та fuzzy search), користувацький інтерфейс (з адаптивним дизайном на TailwindCSS) та адміністративну панель (на базі Django).

Розділ також глибоко занурюється в архітектуру та реалізацію самої системи рекомендацій. Описано клієнт-серверну архітектуру, використання TMDB API як основного джерела даних та локальне кешування для оптимізації. Детально розглянуто серверні алгоритми рекомендацій: контентно-орієнтований фільтр (з використанням cosine similarity), колаборативна модель (на основі SVD, реалізована за допомогою бібліотеки Surprise) та гібридний підхід, що об'єднує ці методи.

Окремо розглянуто реалізацію основних сторінок сайту (головна сторінка, сторінка фільму, пошук, авторизація/реєстрація) з використанням Flask та Jinja2 для бек-енду та Bootstrap 5 для фронт-енду.

Значну частину розділу присвячено опису реалізації алгоритмів рекомендацій, включаючи Content-Based Filtering (з використанням pandas, scikit-learn та TF-IDF), Collaborative Filtering (з детальним описом алгоритму SVD на основі бібліотеки Surprise), а також допоміжних алгоритмів, таких як Trending Algorithm, Tag-Based Recommendation, Fuzzy Search (з використанням RapidFuzz), Dynamic Content Display Algorithm та Sentiment Analysis Based Recommendation.

Процес інтеграції та візуалізації рекомендацій на вебсайті описано з акцентом на динамічне підвантаження даних через REST API, адаптивність дизайну та оптимізацію запитів (кешування, lazy-loading).

4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1 Обовязки роботодавця щодо створення безпечних і нешкідливих умов праці

Згідно з чинним законодавством роботодавець зобов'язаний створити на робочому місці в кожному структурному підрозділі належні умови праці відповідно до нормативно-правових актів, а також забезпечити дотримання вимог законодавства щодо прав працівників з охорони праці [50].

З цією метою роботодавець забезпечує функціонування системи управління охороною праці, а саме:

- створює відповідні служби і призначає посадових осіб, які забезпечують вирішення конкретних питань охорони праці, затверджує інструкції про їх обов'язки, права та відповідальність за виконання покладених на них функцій, а також контролює їх додержання;

- розробляє за участю сторін колективного договору і реалізує комплексні заходи для досягнення встановлених нормативів та підвищення існуючого рівня охорони праці;

- забезпечує виконання необхідних профілактичних заходів відповідно до обставин, що змінюються;

- впроваджує прогресивні технології, досягнення науки і техніки, засоби механізації та автоматизації виробництва, вимоги ергономіки, позитивний досвід з охорони праці тощо;

- забезпечує належне утримання будівель і споруд, виробничого обладнання та устаткування, моніторинг за їх технічним станом;

- забезпечує усунення причин, що призводять до нещасних випадків, професійних захворювань, та здійснення профілактичних заходів, визначених комісіями за підсумками розслідування цих причин;

- організовує проведення аудиту охорони праці, лабораторних досліджень умов праці, оцінку технічного стану виробничого обладнання та устаткування, атестацій робочих місць на відповідність нормативно-правовим

актам з охорони праці в порядку і строки, що визначаються законодавством, та за їх підсумками вживає заходів до усунення небезпечних і шкідливих для здоров'я виробничих факторів;

— розробляє і затверджує положення, інструкції, інші акти з охорони праці, що діють у межах підприємства (далі - акти підприємства), та встановлюють правила виконання робіт і поведінки працівників на території підприємства, у виробничих приміщеннях, на будівельних майданчиках, робочих місцях відповідно до нормативно-правових актів з охорони праці, забезпечує безоплатно працівників нормативно-правовими актами та актами підприємства з охорони праці;

— здійснює контроль за дотриманням працівником технологічних процесів, правил поводження з машинами, механізмами, устаткуванням та іншими засобами виробництва, використанням засобів колективного та індивідуального захисту, виконанням робіт відповідно до вимог з охорони праці;

— організовує пропаганду безпечних методів праці та співробітництво з працівниками у галузі охорони праці;

— вживає термінових заходів для допомоги потерпілим, залучає за необхідності професійні аварійно-рятувальні формування у разі виникнення на підприємстві аварій та нещасних випадків.

Роботодавець зобов'язаний забезпечити за свій рахунок придбання, комплектування, видачу та утримання засобів індивідуального захисту відповідно до нормативно-правових актів з охорони праці та колективного договору, а у разі передчасного зношення цих засобів не з вини працівника, замінити їх за свій рахунок [51].

Особлива увага має приділятися виявленню та усуненню причин, що можуть призвести до нещасних випадків, професійних захворювань, здійсненню профілактичних заходів з метою недопущення аварії на виробництві. Для цього проводяться лабораторні дослідження умов праці, аналізується технічний стан виробничого обладнання та устаткування, здійснюється атестація робочих місць на відповідність їх нормативно-правовим актам з охорони праці, за підсумками

якої роботодавець розробляє та впроваджує заходи усунення небезпечних і шкідливих для здоров'я виробничих факторів [50].

Роботодавець через створену ним службу з охорони праці, комісію з питань охорони праці здійснює контроль за додержанням працівниками вимог виробничої санітарії, гігієни праці, техніки безпеки, використання засобів колективного та індивідуального захисту, виконання робіт згідно з розробленими і затвердженими на підприємстві положеннями, інструкціями та іншими актами з охорони праці.

У свою чергу, працівники, виконуючи свої трудові обов'язки, повинні дотримуватись трудової і технічної дисципліни, підвищувати продуктивність та якість праці.

Згідно з Законодавством України про охорону праці, зокрема статтею 18 Закону України «Про охорону праці», роботодавець зобов'язаний організувати навчання, інструктаж і перевірку знань з питань охорони праці для всіх працівників. Також це передбачено типовим положенням про порядок проведення навчання і перевірки знань з питань охорони праці, затвердженим наказом Держгірпромнагляду № 15 від 26.01.2005 року а потім, Державній службі України з питань праці (Держпраці) згідно з розпорядженням Кабінету Міністрів України № 1021-р від 30 вересня 2015 року.

Основним документом, який регламентує взаємовідносини між трудовим колективом і роботодавцем, є колективний договір. Цей договір розробляється роботодавцем та профспілковою організацією підприємства і затверджується на зборах (конференції) трудового колективу.

Для запланованих заходів з охорони праці роботодавець зобов'язаний виділити цільові кошти та необхідні матеріальні ресурси, витратити які на інші цілі заборонено. Порядок використання цих коштів визначається у колективному договорі і контролюється трудовим колективом.

До обов'язків роботодавця входить своєчасне проведення загальнообов'язкового державного соціального страхування працівників від нещасного випадку на виробництві та професійного захворювання, вживання термінових заходів для допомоги потерпілим, у т.ч. залучення за необхідності

професійних аварійно-рятувальних формувань, вести облік і розслідування нещасних випадків, професійних захворювань і аварій на виробництві.

Роботодавець також може за рахунок власних коштів здійснювати додаткові виплати потерпілим працівникам і членам їх сімей відповідно до колективного або трудового договору.

4.2 Правило техніки безпеки при експлуатації обладнання

На підприємствах різного типу зокрема, в офісних приміщеннях для роботи широко застосовується комп'ютерне та офісне електронне обладнання (ПК, принтери, сервери, мережеві пристрої тощо). Безпечна експлуатація такого обладнання регламентується чинними нормативно-правовими актами, зокрема ДСТУ EN 60204-1:2015 «Безпечність машин. Електрообладнання машин. Загальні вимоги», ДСТУ ІЕС 60364-7-705:2006 щодо електробезпеки, а також вимогами Правил охорони праці під час використання комп'ютерної техніки в офісах. Ці документи встановлюють основні вимоги до монтажу, експлуатації, заземлення, організації електроживлення та запобігання електротравмам працівників [52].

До експлуатації цього устаткування допускаються особи, навчені за програмою технічного мінімуму, що пройшли інструктаж на робочому місці та стажування. Вони повинні бути забезпечені інструкціями безпечної експлуатації обслуговуваного устаткування і ознайомлені під розписку з їх змістом.

Небезпека електромеханічного устаткування для обслуговуючого персоналу обумовлена наявністю в ньому електроприводу і робочих органів, які рухаються з великою швидкістю, а також і можливістю розлітання частинок оброблюваних сировини і продуктів.

Перед ввікненням машин в роботу необхідно перевіряти стан заземлення (занулення), цілісність ізоляції кабелю, надійність кріплення змінних механізмів, наявність і справність огорож, відсутність яких-небудь предметів у завантажувальних пристроях і робочих камерах.

Перед початком роботи з комп'ютерним обладнанням (ПК, принтерами, мережевими пристроями тощо) слід переконатися в його справності: перевірити цілісність кабелів живлення, відсутність пошкоджень на корпусах і роз'ємах, а також справну роботу вентиляції. Роботу з технікою варто починати тільки після увімкнення обладнання у звичайному режимі без навантаження (холостий запуск), щоб виключити сторонні шуми або збої.

У разі появи сторонніх шумів, підвищеного нагрівання корпусу або кабелів, запаху гару чи збоїв у роботі пристроїв необхідно негайно вимкнути обладнання за допомогою кнопки живлення або витягнути вилку з розетки та повідомити відповідальну особу.

Після завершення роботи комп'ютерне обладнання слід коректно вимкнути через програмні засоби («Завершення роботи»), після чого вимкнути живлення (через кнопку на корпусі або від'єднання від мережі). Обладнання необхідно очистити від пилу та забруднень відповідними засобами (серветками, щітками, антистатичними засобами).

Для запобігання перегріву комп'ютерної техніки, особливо серверів, блоків живлення, процесорів, принтерів тощо, необхідно забезпечити належну вентиляцію. Рекомендується використовувати локальні вентиляційні або охолоджувальні системи, зокрема витяжні вентилятори або кондиціонери, а також регулярно очищувати вентиляційні отвори.

Комп'ютерне обладнання, як і інші електроприлади, становить потенційну небезпеку через наявність джерел електричного струму та нагрівальних елементів. Тому важливо дотримуватись правил електробезпеки та експлуатації згідно з технічною документацією та чинними нормативними актами.

Перед ввімкненням будь-якого апарату в роботу необхідно перевірити наявність і стан запобіжних захисних засобів і сигнальних пристроїв, заземлення (занулення) корпусу, а також цілісність ізоляції кабелю і елементів штепсельного роз'єму. Для кожного виду електронагрівальних апаратів розроблені особливі вимоги безпеки під час експлуатації [49].

Експлуатація таких апаратів має бути припинена у разі відмови запобіжних захисних засобів і сигнальних пристроїв, неконтрольованого підвищення

температури електронагрівників, а деяких з них – у разі пониження рівня води нижче допустимого значення, надмірного підвищення тиску пари [49].

Після закінчення роботи електронагрівальний апарат вимикають перемикачем, що є на його корпусі, і вимикають з електричної мережі. Потім видаляють з нього залишки води або продукту.

Технічне обслуговування і ремонт електромеханічного і електронагрівального устаткування проводять за договором механіки ремонтно-монтажних комбінатів, сервісних організацій.

Отже, безпечна експлуатація комп'ютерного та офісного електронного обладнання вимагає дотримання встановлених норм і правил, що регламентуються відповідними нормативно-правовими актами. Застосування технічних засобів контролю, своєчасне технічне обслуговування та належна організація робочого процесу дозволяють знизити ризики електротравм і технічних збоїв в офісних умовах.

4.3 Забезпечення захисту населення від впливу іонізуючих випромінювань

Іонізуюче випромінювання — одна з найбільш серйозних загроз для здоров'я людини у сучасному технологічному світі. У зв'язку з розвитком атомної енергетики, використанням джерел іонізуючого випромінювання у промисловості, медицині та науці, питання забезпечення радіаційного захисту населення набуває першочергового значення для державної політики в галузі охорони здоров'я, екології та національної безпеки.

Право кожної особи, яка перебуває на території України, на захист від впливу іонізуючого випромінювання закріплене у статті 3 Закону України «Про захист людини від впливу іонізуючого випромінювання». Це право реалізується через комплекс превентивних і компенсаційних заходів, спрямованих на обмеження доз опромінення, інформування населення, державний контроль за джерелами випромінювання та регламентоване нормування ризиків [46].

Оснoву радіаційного захисту становить система дозових обмежень. Для осіб з населення встановлено ліміт ефективної дози опромінення в розмірі 1 мЗв на рік, а для професійного персоналу — до 20 мЗв на рік у середньому, з можливістю досягнення 50 мЗв у межах окремого року. Окремо враховуються особливі категорії: вагітні жінки, молодь віком до 18 років, здобувачі освіти, медичний персонал, а також працівники, що виконують тимчасові завдання з джерелами випромінювання. Для кожної з цих груп встановлюються конкретні межі доз навантаження на критичні органи з урахуванням радіочутливості тканин та довгострокових ризиків [46].

Ключову роль у захисті населення відіграють державні норми радіаційної безпеки — НРБУ-97, що регламентують класифікацію осіб за категоріями опромінення (А, Б, В) і встановлюють допустимі рівні доз. До лімітів не включаються природні джерела фону, медичні обстеження, техногенні фонові джерела та аварійні дози. Водночас, у разі настання радіаційних інцидентів передбачено процедури укриття, йодної профілактики та тимчасової евакуації населення, залежно від очікуваних доз навантаження на щитовидну залозу або ефективної дози тіла [47].

Особливу увагу приділено трудовій діяльності з джерелами іонізуючого випромінювання. Юридичні особи та ФОП, які здійснюють таку діяльність, зобов'язані організовувати систематичний контроль за радіаційним фоном, вести облік індивідуальних доз, інформувати персонал, розробляти заходи безпеки та планувати дії у разі радіаційної аварії. До контролю також входять медичні огляди, звітність до державного дозиметричного реєстру та дотримання умов ліцензування.

Важливою складовою є захист під час медичного опромінення, що підпорядковується принципам оптимізації й обґрунтування. Це означає, що будь-яке втручання повинно бути не тільки виправданим, а й максимально безпечним з урахуванням ефективності альтернативних методів. Інформація про дози, отримані під час медичних процедур, підлягає збереженню протягом 50 років, а пацієнт має право бути поінформованим про ризики і навіть відмовитися від процедури, за винятком випадків епідеміологічної небезпеки.

Суттєвим елементом системи захисту є механізм компенсації та відшкодування шкоди, передбачений статтями 19 і 20 Закону. У разі перевищення дозових лімітів з незалежних від особи причин, їй надається грошова компенсація. У разі заподіяння шкоди здоров'ю, життю чи майну внаслідок іонізуючого випромінювання, передбачено обов'язок винного суб'єкта здійснити повне відшкодування. Такі випадки підлягають судовому розгляду з урахуванням доказів дозового перевищення [48].

Забезпечення ефективного захисту населення від іонізуючого випромінювання неможливе без урахування міжнародного досвіду та співпраці із профільними організаціями, які встановлюють глобальні стандарти у цій сфері. Одними з ключових суб'єктів міжнародного регулювання є Міжнародне агентство з атомної енергії (МАГАТЕ), Всесвітня організація охорони здоров'я (ВООЗ), Європейська комісія, а також Міжнародна комісія з радіаційного захисту (ICRP).

МАГАТЕ відіграє провідну роль у формуванні загальнообов'язкових стандартів безпеки, координує зусилля країн-членів у розробці політик захисту та забезпечує технічну підтримку у впровадженні відповідного законодавства. Документи МАГАТЕ, зокрема "Основні міжнародні стандарти безпеки" (GSR Part 3), містять вимоги до захисту здоров'я і безпеки працівників, населення та навколишнього середовища від шкідливого впливу іонізуючого випромінювання.

ВООЗ, у свою чергу, акцентує увагу на питаннях громадського здоров'я та сприяє гармонізації підходів до медичного опромінення населення. Її рекомендації враховуються при розробці політик оптимізації доз опромінення в системах охорони здоров'я.

Значний внесок у розвиток доктрини радіаційного захисту здійснює Міжнародна комісія з радіаційного захисту (ICRP), яка формулює науково обґрунтовані принципи, зокрема, один із основоположних — принцип ALARA (As Low As Reasonably Achievable). Цей принцип полягає у мінімізації доз опромінення до настільки низьких рівнів, наскільки це можливо і доцільно,

враховуючи економічні та соціальні фактори. ALARA є основою для багатьох національних політик та практик у сфері радіаційного захисту.

Відповідно, Україна як учасник міжнародного співтовариства активно інтегрує положення міжнародних нормативних документів у національне законодавство, зокрема через адаптацію Державних норм радіаційної безпеки (НРБУ-97) до сучасних вимог. Такий підхід дозволяє забезпечити відповідність системи радіаційного захисту України сучасним європейським та світовим стандартам, а також гарантувати ефективну взаємодію у випадку транскордонних радіаційних інцидентів або надзвичайних ситуацій.

Ефективне функціонування системи радіаційного захисту неможливе без належного рівня освітньої підготовки та інформованості населення. Знання основних принципів поводження з джерелами іонізуючого випромінювання, розуміння механізмів впливу на організм та обізнаність щодо алгоритмів дій у надзвичайних ситуаціях є важливими елементами національної безпеки та захисту здоров'я громадян.

Інформаційно-просвітницька діяльність охоплює широке коло завдань — від базових шкільних курсів із фізики та основ безпеки життєдіяльності до спеціалізованих програм професійної підготовки для медичних працівників, рятувальних служб, персоналу атомної енергетики та інших критичних галузей. Особливу увагу слід приділяти розробці навчальних модулів, які адаптовані до вікових особливостей та професійних ризиків відповідних аудиторій.

У закладах освіти рекомендовано включати в навчальні програми тематику, пов'язану з радіаційною безпекою, зокрема з історичним аналізом радіаційних аварій (таких як Чорнобиль чи Фукусіма), основами дозиметрії, захисними методами та першочерговими діями у разі радіаційної небезпеки. Застосування інтерактивних методів навчання, таких як симуляції аварійних ситуацій, електронні курси та тренінги, сприяє підвищенню рівня сприйняття та засвоєння матеріалу.

Медичні працівники мають володіти не лише знаннями щодо безпечного використання діагностичного та терапевтичного обладнання, а й бути здатними надавати роз'яснення пацієнтам щодо радіаційних ризиків, особливо у випадках

застосування процедур, пов'язаних із високим рівнем опромінення. Для цього запроваджуються системи післядипломної освіти, періодичної сертифікації та участі у тематичних конференціях.

З метою формування стійкої культури безпеки також проводяться громадські інформаційні кампанії, видаються брошури, буклети, створюються теле- та радіопрограми, що висвітлюють питання радіаційного захисту у доступній для широкої аудиторії формі. Розвиток цифрових платформ дає змогу оперативно поширювати важливу інформацію у випадку реальної або потенційної радіаційної загрози.

У контексті воєнного часу або загрози терористичних актів із використанням радіоактивних матеріалів, підвищення обізнаності населення набуває ще більшого значення. Навчене населення здатне зменшити ступінь паніки, діяти більш організовано та знизити рівень потенційного ураження.

Загалом, система забезпечення радіаційної безпеки в Україні є комплексною, багаторівневою і передбачає як нормативно-правове регулювання, так і науково обґрунтовану стратегію технічного контролю. З огляду на наявність радіаційних ризиків, зокрема внаслідок Чорнобильської катастрофи, в умовах воєнного часу або розвитку ядерної енергетики, забезпечення ефективного захисту населення від впливу іонізуючих випромінювань залишається одним із пріоритетних напрямів державної політики у сфері охорони здоров'я та безпеки довкілля [47].

4.4 Висновки до четвертого розділу

У четвертому розділі було розглянуто питання про охорону праці, що мають ключове значення в забезпеченні належних умов для збереження життя, здоров'я та працездатності працівників і населення. Зокрема, було досліджено обов'язки роботодавця щодо створення безпечних і нешкідливих умов праці, які передбачають організацію системного управління охороною праці, забезпечення працівників інструкціями та засобами індивідуального захисту, проведення інструктажів і навчань, контроль за дотриманням правил техніки безпеки, а

також вжиття заходів з профілактики виробничого травматизму та професійних захворювань.

Окрему увагу було приділено питанням техніки безпеки під час експлуатації комп'ютерного та електронного офісного обладнання. Згідно з чинними нормативно-правовими актами, безпечна робота з таким устаткуванням вимагає дотримання вимог електробезпеки, технічної справності обладнання, належної організації вентиляції, проведення регулярного технічного обслуговування та усунення потенційних ризиків, пов'язаних із перегрівом або пошкодженням електричних компонентів.

Крім того, розділ містив аналіз правових та організаційних засад забезпечення захисту населення від іонізуючого випромінювання. Було акцентовано на важливості дотримання нормативних обмежень ефективної дози опромінення для різних категорій осіб, запровадженні превентивних заходів, контролі за джерелами випромінювання, а також ролі державних норм радіаційної безпеки у мінімізації негативного впливу радіації на здоров'я населення.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи освітнього рівня «Магістр» було досягнуто поставлену мету — розроблено вебплатформу з вбудованою рекомендаційною системою, яка поєднує сучасні алгоритмічні підходи до персоналізації контенту та ефективно реалізована за допомогою фреймворку Flask і супутніх технологій. В першому розділі кваліфікаційної роботи освітнього рівня «Магістр»:

- розглянуто загальні принципи побудови систем персоналізованих рекомендацій;
- охарактеризовано сучасні алгоритми формування рекомендацій, зокрема графові моделі, класифікаційні підходи та оцінювальні метрики;
- проаналізовано методи реалізації алгоритмів у контексті веброзробки;
- обґрунтовано доцільність використання гібридного підходу в рамках платформи з фільмами;
- досліджено програмні засоби та бібліотеки для побудови рекомендаційних систем;
- сформовано концептуальну модель системи з урахуванням практичної реалізації на Python.

В другому розділі кваліфікаційної роботи:

- описано технічні засади архітектури сайту, його функціональних блоків і механізмів взаємодії між компонентами;
- проаналізовано системні вимоги, середовище реалізації, структуру бази даних і логіку обробки запитів;
- подано узагальнену характеристику серверних і клієнтських рішень, з акцентом на інструменти Flask, Bootstrap та REST API.

В третьому розділі кваліфікаційної роботи:

- реалізовано ключові сторінки сайту, включаючи головну, картку фільму, систему пошуку та модулі автентифікації;
- запропоновано архітектуру рекомендаційної підсистеми з алгоритмами SVD, контентного аналізу, популярності та тегів;

- спроектовано адміністративний інтерфейс із можливістю керування контентом та перегляду статистики;

- протестовано функціональність платформи на прикладах різних користувацьких сценаріїв, підтверджено коректність рекомендацій та стабільність роботи системи в цілому.

У розділі «Охорона праці та безпека в надзвичайних ситуаціях» проаналізовано правові та організаційні основи забезпечення безпечних і нешкідливих умов праці на підприємствах, зокрема обов'язки роботодавця щодо створення відповідного виробничого середовища. Описано ключові правила техніки безпеки при експлуатації обладнання, зосереджено увагу на дотриманні інструкцій, попередженні аварійних ситуацій та систематичному контролю за технічним станом робочих засобів. Також висвітлено питання забезпечення захисту населення від впливу іонізуючих випромінювань — розглянуто дозові ліміти, законодавчі норми, принципи регламентації дозових навантажень, а також заходи захисту у разі аварійного радіаційного впливу.

ПЕРЕЛІК ДЖЕРЕЛ

1. Harchenko Alexandr, Bodnarchuk Ihor, Halay Iryna, Yatcyshyn Vasyl. Software Architecture Design on the Base of Method of Hierarchic Optimization. Proceeding of VIIIth International Conference on Perspective Technologies and Methods in MEMS Design. Сторінки. 39-40
2. Олександр Харченко, Ігор Боднарчук. Моделі оптимізації архітектури програмних систем. Редакційно-видавничий відділ Чернігівського державного технологічного університету
3. ЛВ Волинець, НА Гарматюк, ВА Готович. Великі за обсягом набори біомедичних даних та машинне навчання. Матеріали XII Міжнародної науково-практичної конференції молодих учених та студентів „Актуальні задачі сучасних технологій “. Сторінки 370-371
4. Ihor Bodnarchuk, Yuriy Skorenkyu, Taras Kramar, Oleksii Duda, Vyacheslav Nykytyuk. Конференція ІТТАР. Сторінки 414-425
5. Інформація про персоналізацію контенту та рівні [Електронний ресурс]. – Режим доступу до ресурсу: <https://www.imena.ua/blog/personalization/> // Дата доступу до ресурсу: 20.05.2025р.
6. Голенко М. Ю., магістрант, гр. ІПЗм-20-1, Вакалюк Т. А., д-р пед. наук, проф., професор кафедри іпз, Державний університет «Житомирська політехніка». Тема: область застосування та види рекомендаційних систем
7. Особливості залучення аудиторії [Електронний ресурс]. – Режим доступу до ресурсу: <https://www.kiwiagency.com.ua/blog/chto-takoe-engagement-rate-i-kak-ego-opredelit.html> // Дата доступу до ресурсу: 20.05.2025р.
8. Zhurov V., Yasniy O. (2018) Rozrobka systemy upravlinnia kontentom za dopomohoiu pryrodnoi movy [Development of control system by natural language]. Materials of the V Scientific and Technical Conference "Information models, systems and technologies" (Tern., 1-2 February 2018), pp. 3
9. Zhurov V., Yasniy O. (2018) Rozrobka systemy upravlinnia kontentom za dopomohoiu pryrodnoi movy [Development of control system by natural language].

Materials of the V Scientific and Technical Conference "Information models, systems and technologies" (Tern., 1-2 February 2018), pp. 31

10. 10 Product Recommendation Techniques to Improve UX and Conversions [Електронний ресурс] - Режим доступу до ресурсу: <https://conversionxl.com/blog/productrecommendations/> // Дата доступу до ресурсу: 20.05.2025р.

11. Від концепції до реалізації: як побудувати ефективну рекомендаційну систему на основі машинного навчання за два дні [Електронний ресурс]. – Режим доступу до ресурсу: <https://dou.ua/forums/topic/47504/> // Дата доступу до ресурсу: 20.05.2025р.

12. Техніка безпеки під час експлуатації обладнання [Електронний ресурс]. – Режим доступу до ресурсу: <https://oppb.com.ua/news/tehnika-bezpeky-pid-chas-ekspluataciyi-obladnannya-v-remontnyh-maysternyah> // Дата доступу до ресурсу: 20.05.2025р.

13. Hybrid Recommender Systems: Survey and Experiments. [Електронний ресурс] - Режим доступу до ресурсу: <https://pdfs.semanticscholar.org/5880/b9bc3f75f4649b8ec819c3f983a14fca9927.pdf> // Дата доступу до ресурсу: 20.05.2025р.

14. Forbes - How Much Data Do We Create Every Day [Електронний ресурс] – Режим доступу до ресурсу: <https://www.forbes.com/sites/bernardmarr/2018/05/21/how-much-data-do-we-createevery-day-the-mind-blowing-stats-everyone-should-read/#3f7a3fe460ba> // Дата доступу до ресурсу: 20.05.2025р.

15. Як інтегрувати графічні бази даних у вже існуючу ІТ-інфраструктуру: <https://careers.epam.ua/blog/how-to-integrate-graph-databases-into-existing-it-infrastructure> // Дата доступу до ресурсу: 20.05.2025р.

16. Рекомендаційні системи, що базуються на графах [Електронний ресурс]. – Режим доступу до ресурсу: https://habr.com/ru/companies/epam_systems/articles/526748/ // Дата доступу до ресурсу: 20.05.2025р.

17. Алгоритми та структури даних — від «десь чув» до «ефективно

застосовую» [Електронний ресурс]. – Режим доступу до ресурсу: <https://dou.ua/forums/topic/40645/> // Дата доступу до ресурсу: 20.05.2025р.

18. Переваги та недоліки python [Електронний ресурс] – режим доступу: <https://blog.ithillel.ua/articles/perevagi-i-nedoliki-movi-python> // Дата доступу до ресурсу: 20.05.2025р.

19. Робота з моделями на Django [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.djangoproject.com/en/4.2/topics/db/models/> // Дата доступу до ресурсу: 20.05.2025р.

20. Wiki [Електронний ресурс]. – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/%D0%93%D0%BE%D0%BB%D0%BE%D0%B2%D0%BD%D0%B0_%D1%81%D1%82%D0%BE%D1%80%D1%96%D0%BD%D0%BA%D0%B0 // Дата доступу до ресурсу: 20.05.2025р.

21. Тернопільський національний технічний університет імені Івана Пулюя. [Електронний ресурс] – <https://tntu.edu.ua/?p=uk/main> // Дата доступу до ресурсу: 20.05.2025р.

22. Стаття про Django [Електронний ресурс] – Режим доступу до ресурсу: <https://wezom.com.ua/ua/blog/razrabotka-sajtov-na-python-django> // Дата доступу до ресурсу: 20.05.2025р.

23. Робота з моделями на Django [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.djangoproject.com/en/4.2/topics/db/models/> // Дата доступу до ресурсу: 20.05.2025р.

24. Visual Studio Code docs [Електронний ресурс] – Режим доступу до ресурсу: <https://code.visualstudio.com/docs> // Дата доступу до ресурсу: 20.05.2025р.

25. Python docs [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.python.org/3/> // Дата доступу до ресурсу: 20.05.2025р.

26. С. В. Ватаманеску. Про застосування графів у комп'ютерних інформаційних технологіях / С. В. Ватаманеску (Статті. Донецький національний університет імені Василя Стуса, м. Вінниця)

27. Francesco Ricci, Lior Rokach, Bracha Shapira. Recommender Systems Handbook / Ricci, Lior Rokach, Bracha Shapira 2011 – с. 842 (Довідник по рекомендаційних системах).

28. Персональні рекомендації та як вони працюють [Електронний ресурс] : <https://brander.ua/blog/personalni-rekomendacii-yak-se-prasyue>. // Дата доступу до ресурсу: 20.05.2025р.

29. Аналіз поведінки користувача на веб-сайті [Електронний ресурс] : <https://www.promodo.ua/blog/5-prostih-krokv-z-analizu-povedinki-pokupciv-na-sayti>. // Дата доступу до ресурсу: 20.05.2025р.

30. Gene H. Golub, Charles F. Van Loan. Matrix Computations / Gene H. Golub, Charles F. Van Loan, 2016 – с. 518 (Математичні методи роботи з матрицями).

31. Персональні рекомендації та як вони працюють [Електронний ресурс] : <https://brander.ua/blog/personalni-rekomendacii-yak-se-prasyue>. // Дата доступу до ресурсу: 20.05.2025р.

32. Аналіз поведінки користувача на веб-сайті [Електронний ресурс] : <https://www.promodo.ua/blog/5-prostih-krokv-z-analizu-povedinki-pokupciv-na-sayti>. // Дата доступу до ресурсу: 20.05.2025р.

33. Luis Serrano. Grokking Machine Learning / Luis Serrano, 2021 – с. 512

34. Кросплатформність у SMM: як створити унікальний контент для кожної соцмережі [Електронний ресурс]. – Режим доступу до ресурсу: <https://textum.com.ua/blog/krosplatformnist-u-smm-yak-stvoryty-unikalnyj-kontent-dlya-kozhnoyi-soczmerzhi/> // Дата доступу до ресурсу: 20.05.2025р.

35. Як працює колаборативна фільтрація [Електронний ресурс]. – Режим доступу до ресурсу: https://www.vpnunlimited.com/ua/help/cybersecurity/collaborative-filtering?srsId=AfmBOorIFBnOCglRKwGKDqqEnsMrJHtxpdM_-F3XlWAXgK-YJ0ZVs4Ez // Дата доступу до ресурсу: 20.05.2025р.

36. Retailrocket recommender system dataset [Електронний ресурс] – Режим доступу до ресурсу: <https://www.kaggle.com/retailrocket/ecommercedataset#events.csv> // Дата доступу

до ресурсу: 20.05.2025р.

37. Andrew Havens. Beginners guide to creating a REST API [Электронный ресурс]. – Режим доступа: <http://www.andrewhavens.com/posts/20/beginners-guide-to-creating-a-rest-api/> // Дата доступа до ресурсу: 20.05.2025р.

38. McLeod D. Collaborative Filtering for Information Recommendation Systems [Электронный ресурс] / D. McLeod, A. Y. Chen // Research Reports. – 2009. – Режим доступа до ресурсу: http://research.create.usc.edu/cgi/viewcontent.cgi?article=1101&context=nonpublished_reports. // Дата доступа до ресурсу: 20.05.2025р.

39. Koren Y. Matrix Factorization Techniques for Recommender Systems / Y. Koren, B. Robert, V. Chris. // Journal Computer. – 2009. – №42 (8). – pp. 30–37.

40. Dunning T. Practical Machine Learning: Innovations in Recommendation / T. Dunning, E. Friedman, A. Felfernig // O'Reilly Media, Inc. , 2014. – 56 с.

41. Lucene documentation [Электронный ресурс] – Режим доступа до ресурсу: https://lucene.apache.org/core/7_5_0/index.html // Дата доступа до ресурсу: 20.05.2025р.

42. Elasticsearch documentation [Электронный ресурс] – Режим доступа до ресурсу: <https://www.elastic.co/guide/index.html> // Дата доступа до ресурсу: 20.05.2025р.

43. Elasticsearch Spark support [Электронный ресурс] – Режим доступа до ресурсу: <https://www.elastic.co/guide/en/elasticsearch/hadoop/current/spark.html> // Дата доступа до ресурсу: 20.05.2025р.

44. HDFS User Guide [Электронный ресурс] – Режим доступа до ресурсу: <https://hadoop.apache.org/docs/r2.4.1/hadoop-project-dist/hadoopdfs/HdfsUserGuide.html> // Дата доступа до ресурсу: 20.05.2025р.

45. Hadoop tutorial [Электронный ресурс] – Режим доступа до ресурсу: <https://www.tutorialspoint.com/hadoop/> // Дата доступа до ресурсу: 20.05.2025р.

46. Mahout documentation [Электронный ресурс] – Режим доступа до ресурсу: <https://mahout.apache.org/> // Дата доступа до ресурсу: 20.05.2025р.

47. Building a Correlated Cross-Occurrence (CCO) Recommenders with the Mahout CLI [Электронный ресурс] – Режим доступа до ресурсу:

<http://mahout.apache.org/users/recommender/intro-cooccurrence-spark.html> // Дата доступу до ресурсу: 20.05.2025р.

48. Архітектура рекомендацій: як дати користувачеві соцмережі те, що йому сподобається [Електронний ресурс]. – Режим доступу до ресурсу: <https://habr.com/ru/companies/magnus-tech/articles/753706/> // Дата доступу до ресурсу: 20.05.2025р.

49. Техніка безпеки під час експлуатації обладнання [Електронний ресурс]. – Режим доступу до ресурсу: <https://oppb.com.ua/news/tehnika-bezpeky-pid-chas-ekspluataciyi-obladnannya-v-remontnyh-maysternyah> // Дата доступу до ресурсу: 20.05.2025р.

50. Закон України про захист людини від впливу іонізуючого випромінювання. Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/15/98-%D0%B2%D1%80#Text> // Дата доступу до ресурсу: 20.05.2025р.

51. Правове регулювання захисту людини від впливу іонізуючого випромінювання. Режим доступу до ресурсу: https://legalaid.wiki/index.php/%D0%9F%D1%80%D0%B0%D0%B2%D0%BE%D0%B2%D0%B5_%D1%80%D0%B5%D0%B3%D1%83%D0%BB%D1%8E%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F_%D0%B7%D0%B0%D1%85%D0%B8%D1%81%D1%82%D1%83_%D0%BB%D1%8E%D0%B4%D0%B8%D0%BD%D0%B8_%D0%B2%D1%96%D0%B4_%D0%B2%D0%BF%D0%BB%D0%B8%D0%B2%D1%83_%D1%96%D0%BE%D0%BD%D1%96%D0%B7%D1%83%D1%8E%D1%87%D0%BE%D0%B3%D0%BE_%D0%B2%D0%B8%D0%BF%D1%80%D0%BE%D0%BC%D1%96%D0%BD%D1%8E%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F_%D0%92%D1%96%D0%B4%D1%88%D0%BA%D0%BE%D0%B4%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F_%D1%88%D0%BA%D0%BE%D0%B4%D0%B8 // Дата доступу до ресурсу: 20.05.2025р.

52. Про захист людини від впливу іонізуючого випромінювання. Режим доступу до ресурсу: https://protocol.ua/ua/pro_zahist_lyudini_vid_vplivu_ionizuyuchogo_viprominyuvannya/ // Дата доступу до ресурсу: 20.05.2025р.

53. Створення безпечних і нешкідливих умов праці. Режим доступу до ресурсу: <https://oppb.com.ua/news/stvorenniya-bezpechnyh-i-neshkidlyvyh-umov-praci> // Дата доступу до ресурсу: 20.05.2025р.

54. Створення безпечних та нешкідливих умов праці. Режим доступу до ресурсу:

[https://legalaid.wiki/index.php/%D0%A1%D1%82%D0%B2%D0%BE%D1%80%D0%B5%D0%BD%D0%BD%D1%8F_%D0%B1%D0%B5%D0%B7%D0%BF%D0%B5%D1%87%D0%BD%D0%B8%D1%85_%D1%96_%D0%BD%D0%B5%D1%88%D0%BA%D1%96%D0%B4%D0%BB%D0%B8%D0%B2%D0%B8%D1%85_%D1%83%D0%BC%D0%BE%D0%B2_%D0%BF%D1%80%D0%B0%D1%86%D1%96_%D0%94%D0%B5%D1%80%D0%B6%D0%B0%D0%B2%D0%BD%D0%B8%D0%B9_%D0%BD%D0%B0%D0%B3%D0%BB%D1%8F%D0%B4_%D0%B7%D0%B0_%D0%BE%D1%85%D0%BE%D1%80%D0%BE%D0%BD%D0%BE%D1%8E_%D0%BF%D1%80%D0%B0%D1%86%D1%96#:~:text=%D0%A0%D0%BE%D0%B1%D0%BE%D1%82%D0%BE%D0%B4%D0%B0%D0%B2%D0%B5%D1%86%D1%8C%20%D0%B7%D0%BE%D0%B1%D0%BE%D0%B2%D1%8F%D0%B7%D0%B0%D0%BD%D0%B8%D0%B9%20%D0%B2%D0%B6%D0%B8%D0%B2%D0%B0%D1%82%D0%B8%20%D0%B7%D0%B0%D1%85%D0%BE%D0%B4%D1%96%D0%B2%20%D0%B4%D0%BB%D1%8F%20%D0%B7%D0%B0%D0%B1%D0%B5%D0%B7%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%BD%D1%8F%20%D0%B1%D0%B5%D0%B7%D0%BF%D0%B5%D0%BA%D0%B8%20%D1%96%20%D0%B7%D0%B0%D1%85%D0%B8%D1%81%D1%82%D1%83,\)%20\(%D1%81%D1%82%D0%B0%D1%82%D1%82%D1%8F%20158%20%D0%9A%D0%97%D0%BF%D0%9F\).](https://legalaid.wiki/index.php/%D0%A1%D1%82%D0%B2%D0%BE%D1%80%D0%B5%D0%BD%D0%BD%D1%8F_%D0%B1%D0%B5%D0%B7%D0%BF%D0%B5%D1%87%D0%BD%D0%B8%D1%85_%D1%96_%D0%BD%D0%B5%D1%88%D0%BA%D1%96%D0%B4%D0%BB%D0%B8%D0%B2%D0%B8%D1%85_%D1%83%D0%BC%D0%BE%D0%B2_%D0%BF%D1%80%D0%B0%D1%86%D1%96_%D0%94%D0%B5%D1%80%D0%B6%D0%B0%D0%B2%D0%BD%D0%B8%D0%B9_%D0%BD%D0%B0%D0%B3%D0%BB%D1%8F%D0%B4_%D0%B7%D0%B0_%D0%BE%D1%85%D0%BE%D1%80%D0%BE%D0%BD%D0%BE%D1%8E_%D0%BF%D1%80%D0%B0%D1%86%D1%96#:~:text=%D0%A0%D0%BE%D0%B1%D0%BE%D1%82%D0%BE%D0%B4%D0%B0%D0%B2%D0%B5%D1%86%D1%8C%20%D0%B7%D0%BE%D0%B1%D0%BE%D0%B2%D1%8F%D0%B7%D0%B0%D0%BD%D0%B8%D0%B9%20%D0%B2%D0%B6%D0%B8%D0%B2%D0%B0%D1%82%D0%B8%20%D0%B7%D0%B0%D1%85%D0%BE%D0%B4%D1%96%D0%B2%20%D0%B4%D0%BB%D1%8F%20%D0%B7%D0%B0%D0%B1%D0%B5%D0%B7%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%BD%D1%8F%20%D0%B1%D0%B5%D0%B7%D0%BF%D0%B5%D0%BA%D0%B8%20%D1%96%20%D0%B7%D0%B0%D1%85%D0%B8%D1%81%D1%82%D1%83,)%20(%D1%81%D1%82%D0%B0%D1%82%D1%82%D1%8F%20158%20%D0%9A%D0%97%D0%BF%D0%9F).) // Дата доступу до ресурсу: 20.05.2025р.

55. Ben Shaw. Web Development with Django: Learn to build modern web applications with a Python-based framework / Ben Shaw, 2021 – с. 826

56. Francesco Ricci, Lior Rokach, Bracha Shapira. Recommender Systems Handbook / Ricci, Lior Rokach, Bracha Shapira 2011 – с. 842 (Довідник по рекомендаційних системах).

ДОДАТКИ

Тези першої конференції

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Тернопільський національний технічний університет імені Івана Пулюя (Україна)
Університет імені П'єра і Марії Кюрі (Франція)
Маріборський університет (Словенія)
Технічний університет у Кошице (Словаччина)
Вільнюський технічний університет ім. Гедимінаса (Литва)
Наукове товариство ім. Т.Шевченка

АКТУАЛЬНІ ЗАДАЧІ СУЧАСНИХ ТЕХНОЛОГІЙ

Збірник
тез доповідей

**XIII Міжнародної науково-практичної
конференції молодих учених та студентів**
11-12 грудня 2024 року



УКРАЇНА
ТЕРНОПІЛЬ – 2024

БАЗИ ПЛАТФОРМИ FIT

74. **Ю.Р. Курчак, О.Р. Кучма**
МОДЕЛЮВАННЯ ТА АНАЛІЗ СИСТЕМИ РЕКОМЕНДАЦІЙ НА ВЕБ-САЙТІ
НА ОСНОВІ ТЕОРІЇ ГРАФІВ І СТАТИСТИЧНИХ АЛГОРИТМІВ
75. **Я. В. Мозговий, О. О. Базиль**
ПРОЦЕДУРНА ГЕНЕРАЦІЯ В ІГРАХ: МОЖЛИВОСТІ ДЛЯ СТВОРЕННЯ
НЕСКІНЧЕННИХ СВІТІВ
76. **Я.І. Музичишин**
МЕТОДИ РОЗПІЗНАВАННЯ ВІЛЬНИХ ПАРКОМІСЬ
77. **Яцишин В.В., Демиденко А.О., Яцишин Вік. В.**
ПІДХОДИ ДО ВИЯВЛЕННЯ АНОМАЛІЙ ТРАФІКУ У КОМП'ЮТЕРНИХ
МЕРЕЖАХ

УДК 004.4

Ю.Р. Курчак, О.Р. Кучма

(Тернопільський національний технічний університет імені Івана Пулюя, Україна)

МОДЕЛЮВАННЯ ТА АНАЛІЗ СИСТЕМИ РЕКОМЕНДАЦІЙ НА ВЕБ-САЙТІ НА ОСНОВІ ТЕОРІЇ ГРАФІВ І СТАТИСТИЧНИХ АЛГОРИТМІВ

Y.R. Kurchak, O.R. Kuchma

MODELING AND ANALYSIS OF A WEBSITE RECOMMENDATION SYSTEM BASED ON GRAPH THEORY AND STATISTICAL ALGORITHMS

В наші дні веб-сайти потребують ефективних систем рекомендацій для забезпечення персоналізованого користувацького досвіду, підвищення лояльності аудиторії та збільшення конверсії. У цьому контексті використання теорії графів і статистичних алгоритмів відкриває нові можливості для моделювання взаємодії між користувачами та об'єктами системи.

Теорія графів дозволяє представити структуру даних у вигляді вузлів (користувачів і об'єктів) та ребер (взаємодій, таких як оцінки, перегляди чи покупки). Графова модель сприяє виявленню кластерів схожих об'єктів і груп користувачів, що підвищує точність рекомендацій. Статистичні алгоритми, зокрема методи колаборативної фільтрації та метрики подібності, забезпечують аналіз схожості між вузлами, використовуючи такі показники, як косинусна подібність або кореляція [1].

На веб-сайті статистичні алгоритми, такі як колаборативна фільтрація та метрики подібності, аналізують дані про взаємодію користувачів із контентом для визначення схожості між вузлами – користувачами або об'єктами. Цей процес починається зі збору даних, наприклад, оцінок, переглядів чи історії покупок, які формують базу даних про взаємодії. Ці дані представляються у вигляді матриці, де користувачі та об'єкти пов'язані через рівень взаємодії.

Таблиця 1. Матриця взаємодії користувача з елементами на веб-сайті

Користувач/Елемент	Елемент 1	Елемент 2	Елемент 3	Елемент 4
Користувач 1	5	4	0	0
Користувач 2	0	0	4	0
Користувач 3	3	5	0	4
Користувач 4	0	0	0	5

Матриця представляє взаємодію користувачів із товарами на веб-сайті інтернет-магазину. Рядки відображають користувачів, а стовпці – доступні товари. Значення в клітинках показують оцінки, які користувачі дали товарам, або їхню відсутність, якщо взаємодії не було. Оцінки відображають рівень задоволення чи зацікавленості користувача конкретним товаром.

Ця матриця використовується для аналізу поведінки користувачів і подібностей між ними. Система рекомендацій може аналізувати схожі оцінки, щоб виявляти спільні вподобання користувачів або зв'язки між товарами, які часто отримують подібні оцінки. На основі цього формується персоналізований підхід до кожного користувача, де система прогнозує, які товари можуть бути для нього цікавими [4].

Результати такого аналізу дозволяють не лише рекомендувати товари, які користувачі ще не бачили, але й заповнювати прогалини у взаємодіях. Завдяки цьому підходу забезпечується ефективна персоналізація рекомендацій.

Запропонована модель дозволяє оптимізувати систему рекомендацій шляхом використання графових алгоритмів для кластеризації, що зменшує складність обробки великих даних. Додатково, використання статистичних алгоритмів адаптує рекомендації до індивідуальних потреб користувача, підвищуючи релевантність пропозицій [5].

Розробка системи рекомендацій може здійснюватися різними методами, залежно від типу даних і потреб системи. Одним із підходів є аналіз взаємодій користувачів із товарами, що дозволяє виявляти схожості між користувачами або товарами. Інший підхід зосереджується на характеристиках самих товарів, щоб рекомендувати схожі за змістом або атрибутами об'єкти [2].

Поєднання цих підходів створює гібридні системи, які забезпечують більшу точність рекомендацій. Використання сучасних математичних і алгоритмічних методів, таких як факторизація матриць або аналіз графів, допомагає ефективно обробляти великі обсяги даних і знаходити приховані зв'язки між користувачами та товарами.

Додатково, впровадження машинного та глибокого навчання дозволяє адаптувати систему до змін у поведінці користувачів, враховуючи історію їхніх взаємодій і контекстуальні фактори. Такі методи роблять системи рекомендацій більш персоналізованими й релевантними для кожного користувача.

Системи рекомендацій є важливим інструментом у сучасному цифровому середовищі, адже вони забезпечують не лише зручність користувача, а й підвищують ефективність бізнес-процесів [2].



Рисунок 1. Алгоритми персональних рекомендацій [3]

У контексті аналізу взаємодій користувачів і товарів вони відкривають можливості для більш глибокого розуміння поведінки користувачів, виявлення трендів і прогнозування майбутніх уподобань. Інтеграція передових методів аналізу даних, таких як графові моделі, машинне навчання та алгоритми кластеризації, дозволяє створювати адаптивні, масштабовані системи, які ефективно працюють навіть у великих базах даних [3].

Література

6. С. В. Ватаманеску. Про застосування графів у комп'ютерних інформаційних технологіях / С. В. Ватаманеску (Статті. Донецький національний університет імені Василя Стуса, м. Вінниця)
7. Francesco Ricci, Lior Rokach, Bracha Shapira. Recommender Systems Handbook / Ricci, Lior Rokach, Bracha Shapira 2011 – с. 842 (Довідник по рекомендаційних системах).

8. Персональні рекомендації та як вони працюють [Електронний ресурс]: <https://brander.ua/blog/personalni-rekomendacii-yak-se-prasyue>. Доступ до ресурсу: 28.11.2024
9. Аналіз поведінки користувача на веб-сайті [Електронний ресурс]: <https://www.promodo.ua/blog/5-prostih-krokv-z-analizu-povedinki-pokupciv-na-sayti>. Доступ до ресурсу: 28.11.2024
10. Gene H. Golub, Charles F. Van Loan. Matrix Computations / Gene H. Golub, Charles F. Van Loan, 2016 – с. 518 (Математичні методи роботи з матрицями).

Тези другої конференції

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
імені ІВАНА ПУЛЮЯ
НАУКОВЕ ТОВАРИСТВО ім. ШЕВЧЕНКА

МАТЕРІАЛИ

ХІІ НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ

«ІНФОРМАЦІЙНІ МОДЕЛІ,
СИСТЕМИ ТА ТЕХНОЛОГІЇ»



18–19 грудня 2024 року

ТЕРНОПІЛЬ
2024

К. Костюк РОЗРОБКА МЕТОДОЛОГІЇ ДЛЯ ОЦІНКИ РИЗИКІВ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ ДЛЯ АГРОХОЛДИНГУ K. Kostyuk DEVELOPMENT OF A METHODOLOGY FOR INFORMATION SECURITY RISK ASSESSMENT FOR AN AGRICULTURAL HOLDING	46
Ю. Р. Курчак МОДЕЛЮВАННЯ ТА АНАЛІЗ СИСТЕМИ РЕКОМЕНДАЦІЙ НА ВЕБ-САЙТІ НА ОСНОВІ ТЕОРІЇ ГРАФІВ І СТАТИСТИЧНИХ АЛГОРИТМІВ Y. R. Kurchak MODELING AND ANALYSIS OF A WEBSITE RECOMMENDATION SYSTEM BASED ON GRAPH THEORY AND STATISTICAL ALGORITHMS	47
В. А. Лабчук ДОСЛІДЖЕННЯ МОЖЛИВОСТЕЙ ЗАСТОСУВАННЯ MICROSOFT LINQ ДЛЯ ЗАДАЧ КОНСОЛІДАЦІЇ ДАНИХ V. A. Labchuk RESEARCHMENT OF THE DATA CONSOLIDATION TASKS' OPPORTUNITIES WITH THE HELP OF USING MICROSOFT LINQ TECHNOLOGY	48
В. А. Лабчук ПЕРЕВАГИ ТА НЕДОЛІКИ LINQ В ПОРІВНЯННІ З SQL-КОДОМ V. A. Labchuk PROS AND CONS OF USING LINQ INSTEAD OF RAW-SQL	49
С. В. Литвиненко, М. Є. Фриз МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ ТАРГЕТОВАНОЇ РЕКЛАМИ S. V. Lytvynenko, M. E. Friz MATHEMATICAL SUPPORT OF TARGETED ADVERTISING	50
О. Ловчук, В. Дубельт, А. Лисий KUBERNETES В ОБЧИСЛЮВАЛЬНІЙ ІНФРАСТРУКТУРІ РОЗУМНИХ МІСТ O. Lovchuk, V. Dubelt, A. LYSYI KUBERNETES IN THE SMART CITIES COMPUTING INFRASTRUCTURE	51
О. Ловчук, Р. Катрич, В. ДУДА АКТУАЛЬНІСТЬ ХМАРНОГО МАСШТАБУВАННЯ ТА KUBERNETES O. Lovchuk, R. Katrych, V. Duda THE RELEVANCE OF CLOUD SCALATION AND KUBERNETES	52
П. Луговський, І. Фалендиш ВИБІР ІНФОРМАТИВНИХ ОЗНАК ДЛЯ ЗАДАЧІ ВИЯВЛЕННЯ ШКІДЛИВОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ В ANDROID P. Lugovskyi, I. Falendysh SELECTION OF INFORMATIVE FEATURES FOR THE TASK OF ANDROID MALWARE DETECTION	53
В. Мазур МЕТОДИ ЗАХИСТУ ВІД КІБЕРАТАК НА ОСНОВІ ШТУЧНОГО ІНТЕЛЕКТУ V. Mazur AI-BASED METHODS FOR PROTECTION AGAINST CYBERATTACKS	54
М. Маланчук, Г. Козбур ОГЛЯД ВІТЧИЗНЯНИХ АВТОМАТИЗОВАНИХ СИСТЕМ ДЛЯ НАДАННЯ ПЕРШОЇ МЕДИЧНОЇ ДОПОМОГИ З ВИКОРИСТАННЯМ ЧАТ-БОТІВ M. Malanchuk, H. Kozbur REVIEW OF DOMESTIC AUTOMATED SYSTEMS FOR FIRST AID USING CHATBOTS	55

УДК 004.4

Ю. Р. Курчак

(Тернопільський національний технічний університет імені Івана Пулюя, Україна)

МОДЕЛЮВАННЯ ТА АНАЛІЗ СИСТЕМИ РЕКОМЕНДАЦІЙ НА ВЕБ-САЙТІ НА ОСНОВІ ТЕОРІЇ ГРАФІВ І СТАТИСТИЧНИХ АЛГОРИТМІВ

UDC 004.4

Y. R. Kurchak

MODELING AND ANALYSIS OF A WEBSITE RECOMMENDATION SYSTEM BASED ON GRAPH THEORY AND STATISTICAL ALGORITHMS

Веб-сайти потребують інтеграції ефективних систем рекомендацій для персоналізації взаємодії користувачів. У цьому проєкті основою є Python та його фреймворки Flask або Django, що забезпечують швидку розробку, гнучкість і потужні можливості інтеграції алгоритмів машинного навчання [1]. База даних буде на основі PostgreSQL або MySQL, що забезпечить зберігання та доступ до великих обсягів даних.

Система працює за рахунок обробки даних про взаємодії користувачів із контентом. Дані, такі як перегляди, оцінки чи історія покупок, зберігаються в базі та аналізуються [3]. Основою є методи колаборативної фільтрації, що порівнюють користувачів або об'єкти для пошуку схожостей. Для виявлення груп схожих об'єктів та користувачів використовуються графові алгоритми, наприклад, PageRank чи Louvain для кластеризації [2].

Гібридний підхід поєднує аналіз поведінкових даних із характеристиками контенту, що дозволяє покращити точність рекомендацій і зменшити проблему «холодного старту» [4]. Особливий акцент зроблено на впровадженні методів факторизації матриць, для пошуку прихованих взаємозв'язків між користувачами та контентом. Алгоритми реалізуються за допомогою бібліотек, зокрема Scikit-learn, Pandas і NumPy.

Ключовими перевагами системи є її здатність підвищувати зручність користувачів і збільшувати конверсії завдяки точним рекомендаціям. Разом із тим, виклики включають обробку великих обсягів даних та забезпечення конфіденційності користувачів. Для вирішення цих завдань буде використано масштабовані обчислення за допомогою хмарних сервісів, таких як AWS або Google Cloud.

Проєкт демонструє, як інтеграція сучасних алгоритмів, таких як машинне навчання, графові моделі та кластеризація, дозволяє створити ефективну рекомендаційну систему. Завдяки цьому веб-сайт стає не лише інструментом доступу до контенту, а й платформою для глибшого розуміння потреб користувачів та їхньої поведінки [5].

Література

1. Ben Shaw. Web Development with Django: Learn to build modern web applications with a Python-based framework / Ben Shaw, 2021 – с. 826
2. Francesco Ricci, Lior Rokach, Bracha Shapira. Recommender Systems Handbook / Ricci, Lior Rokach, Bracha Shapira 2011 – с. 842 (Довідник по рекомендаційних системах).
3. Персональні рекомендації та як вони працюють [Електронний ресурс]: <https://brander.ua/blog/personalni-rekomendacii-yak-se-pracyue>. Доступ до ресурсу: 28.11.2024
4. Аналіз поведінки користувача на веб-сайті [Електронний ресурс]: <https://www.promodo.ua/blog/5-prostih-krokv-z-analizu-povedinki-pokupciv-na-sayti>. Доступ до ресурсу: 28.11.2024
5. Luis Serrano. Grokking Machine Learning / Luis Serrano, 2021 – с. 512.