Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет інформаційно-аналітичних технологій та менеджменту
(повна назва)

Кафедра прикладної математики
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Розробка рекомендаційної системи для аудіоконтенту з застосуванням методів машинного навчання без учителя
(тема)

Виконав:
здобувач _2_ року навчання, групи _САУм-23-1
Ільницький В.Б.
(прізвище, ініціали)
Спеціальність
124 Системний аналіз
(код і повна назва спеціальності)
Тип програми _освітньо-професійна
(освітньо-професійна або освітньо-наукова)
Освітня програма
Системний аналіз і управління
(повна назва освітньої програми)
Керівник проф. Сидоров М.В.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри ПМ _____ Сидоров М.В.
(підпис) (прізвище, ініціали)

2025 р.

Харківський національний університет радіоелектроніки

Факультет _інформаційно-аналітичних технологій та менеджменту
Кафедра _прикладної математики
Рівень вищої освіти _другий (магістерський)
Спеціальність _124 Системний аналіз
(код і повна назва)
Тип програми _освітньо-професійна
(освітньо-професійна або освітньо-наукова)
Освітня програма _Системний аналіз і управління
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри ПМ _____
(підпис)
“ _25_ ” _листопада_ 2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві _Ільницькому Владиславу Богдановичу
(прізвище, ім’я, по батькові)

1. Тема роботи _Розробка рекомендаційної системи для аудіоконтенту з застосуванням методів машинного навчання без учителя

затверджена наказом по університету від 22 листопада 2024 р. № 1228 Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 6 січня 2025 р.

3. Вихідні дані до роботи _модель рекомендаційної системи для аудіоконтенту

4. Перелік питань, що потрібно опрацювати в роботі
1. Системний аналіз предметної області
2. Вибір і обґрунтування методу розв’язання
3. Програмна реалізація
4. Результати обчислювального експерименту
5. Аналіз можливих застосувань

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій

1. Актуальність теми роботи
2. Постановка задачі
3. Системний аналіз предметної області
4. Результати обчислювального експерименту

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Підбір та вивчення технічної літератури за темою роботи	25 листопада – 1 грудня 2024 р.	виконано
2	Вибір та обґрунтування методу	2 – 8 грудня 2024 р.	виконано
3	Розробка алгоритму і програми	9 – 22 грудня 2023 р.	виконано
4	Проведення аналітичних досліджень та розрахунків	23 – 29 грудня 2024 р.	виконано
5	Робота над текстом пояснювальної записки	30 грудня 2024 р. – 9 січня 2025 р.	виконано
6	Представлення роботи на рецензію в ЕК	10 січня 2025 р.	виконано

Дата видачі завдання _25_ листопада 2024 р.

Здобувач _____
(підпис)

Керівник роботи _____ проф. Сидоров М.В.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 61 с., 6 табл., 29 рис., 1 дод., 12 джерел.

ГІБРИДНА РЕКОМЕНДАЦІЙНА СИСТЕМА, КОЛАБОРАТИВНА ФІЛЬТРАЦІЯ, МАШИННЕ НАВЧАННЯ БЕЗ УЧИТЕЛЯ, ФІЛЬТРАЦІЯ НА ОСНОВІ ВМІСТУ.

Об’єкт дослідження – рекомендаційні системи для аудіоконтенту.

Мета роботи – створення рекомендаційної системи, яка здатна визначати вподобання користувачів та пропонувати відповідний аудіоконтент.

Методи дослідження – колаборативна фільтрації та фільтрація на основі вмісту.

Було створено багатошарову архітектуру рекомендаційної системи, яка забезпечує персоналізовані рекомендації, демонструючи ефективність інтеграції різних підходів і сучасних технологій. Система здатна працювати з великими обсягами даних.

Отримані результати доцільно застосовувати для поліпшення рекомендаційних платформ, зокрема в стрімінгових сервісах, онлайн-магазинах та інших сферах, що вимагають персоналізованого підходу.

Загалом, сфери застосування рекомендаційних систем це цифрові платформи, які працюють з великим обсягом користувацьких даних і надають рекомендації.

Дослідження створює основу для вдосконалення рекомендаційних систем, підвищуючи їхню точність і продуктивність, а також адаптуючи їх до динаміки сучасного інформаційного середовища.

Гібридний підхід забезпечує високу ефективність рекомендацій. Пропонується подальший розвиток технологій для поліпшення адаптивності та продуктивності системи, розширення її застосування у різних галузях

ABSTRACT

Introductory note: 61 pages, 6 tables, 29 figures, 1 appendix, 12 sources.

COLLABORATIVE FILTERING, CONTENT-BASED FILTERING, HYBRID RECOMMENDER SYSTEM, UNSUPERVISED MACHINE LEARNING.

Object of research – recommender systems for audio content.

Purpose of work – to develop a recommender system capable of determining user preferences and providing personalized audio content.

Methods of research – collaborative filtering and content-based filtering.

A multi-layered architecture of a recommender system was created, ensuring personalized recommendations and demonstrating the effectiveness of integrating various approaches and modern technologies. The system is capable of handling large volumes of data.

The results are recommended for improving recommendation platforms, particularly in streaming services, online stores, and other areas that require a personalized approach.

Overall, the application domains for recommender systems include digital platforms that process large amounts of user data and deliver recommendations.

This research lays the foundation for enhancing recommender systems, improving their accuracy and productivity, and adapting them to the dynamics of the modern information environment.

The hybrid approach ensures high recommendation efficiency. Further development of technologies is proposed to enhance the system’s adaptability and performance and to expand its applications across various industries.

ЗМІСТ

ВСТУП

Актуальність теми. Рекомендаційні системи стали невід'ємною частиною сучасного цифрового світу, відіграючи ключову роль у персоналізації користувацького досвіду та оптимізації взаємодії між людьми та інформацією. У світі, де обсяг доступних даних постійно зростає, ці системи виступають незамінними посередниками, допомагаючи користувачам орієнтуватися в морі інформації та знаходити релевантний контент.

Сьогодні рекомендаційні системи широко застосовуються в різноманітних сферах: від електронної комерції та стрімінгових сервісів до соціальних мереж та освітніх платформ. Вони не лише покращують користувацький досвід, пропонуючи персоналізований контент, але й відіграють важливу роль у бізнес-стратегіях, заохочуючи користувачів проводити більше часу на платформі та частіше здійснювати цільові дії, такі як покупки чи перегляд контенту.

Актуальність дослідження та розробки рекомендаційних систем зумовлена кількома факторами. По-перше, зростаюча кількість даних та інформаційне перевантаження користувачів створюють потребу в ефективних механізмах фільтрації та персоналізації контенту. По-друге, постійно зростаючі очікування користувачів щодо персоналізованого досвіду вимагають все більш складних та точних алгоритмів рекомендацій. По-третє, рекомендаційні системи мають значний вплив на формування споживчих звичок та інформаційного поля користувачів, що піднімає важливі етичні питання та проблеми інформаційних бульбашок.

У контексті технологічного розвитку, поява нових методів машинного навчання та збільшення обчислювальних потужностей відкривають нові можливості для вдосконалення рекомендаційних систем. Інтеграція різних підходів, а також використання глибокого навчання, дозволяють створювати більш точні та ефективні рекомендації.

Виходячи з цього, актуальним є розробка рекомендаційної системи для аудіоконтенту з застосуванням методів машинного навчання без учителя.

Мета і завдання кваліфікаційної роботи. Метою кваліфікаційної роботи є створення рекомендаційної системи, яка здатна визначати вподобання користувачів та пропонувати відповідний аудіоконтент. Для досягнення поставленої мети необхідно виконати наступні завдання:

– провести аналіз сучасного стану методів розробки рекомендаційних систем та ознайомитися з особливостями архітектури та реалізації гібридних систем;

– застосувати гібридний метод до розв’язання задачі рекомендаційної системи та скласти відповідний алгоритм;

– програмно реалізувати модель гібридної системи на мові програмування Python за допомогою бібліотеки scikit-learn;

– провести навчання моделі на відповідному датасеті;

– провести низку обчислювальних експериментів та проаналізувати їх результати.

Об’єктом дослідження є рекомендаційні системи для аудіоконтенту.

Предметом дослідження є методи машинного навчання без учителя, застосовані у рекомендаційних системах.

Методи дослідження. У кваліфікаційній роботі використовується гібридний метод, який складається з колаборативної фільтрації та фільтрації на основі вмісту.

Публікації. Результати, отримані у роботі, було представлено на ІІІ Міжнародній науково-практичній конференції англійською мовою «Навчання і викладання: у світі після війни» (м. Харків, 8 листопада 2024 р.) [1].

1 СИСТЕМНИЙ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧ ДОСЛІДЖЕННЯ

1.1 Системний аналіз рекомендаційної системи для аудіоконтенту

1.1.1 Вербальна модель системи

Об’єкт аналізу – «Рекомендаційна система для аудіоконтенту».

Предмет аналізу – «Застосування методів машинного навчання без учителя».

Точка зору: дослідник.

Ціль: отримання релевантних рекомендацій.

Для того, щоб глибше проаналізувати систему в її зовнішньому контексті, науковці часто вдаються до використання моделі під назвою «чорна скриня». Ця модель є незамінною у вивченні складних систем, оскільки вона зосереджує увагу на зовнішніх аспектах системи і її кордонах, не вдаючись до аналізу внутрішніх процесів. Однак цей підхід може призвести до того, що значущість деяких внутрішніх механізмів може бути недооцінена [2, 3].

В моделі «чорної скриньки» ключовими елементами є входи та виходи. На прикладі нашої системи, вхідними даними є аудіоконтент у вигляді пісні, тоді як вихідними даними є список пісень, які можуть зацікавити користувача. Ця модель зображена на рисунку 1.1.

Детальний опис впливу зовнішнього середовища на систему є вкрай важливим. Він допомагає зрозуміти, які фактори можуть впливати на ефективність роботи системи і виявити потенційні проблеми та недоліки, з якими система може стикнутися. Зовнішнє середовище системи можна визначити як комплекс елементів, які оточують систему і взаємодіють з нею як безпосередньо, так і опосередковано. На рисунку 1.2 представлено схему «система – зовнішнє середовище».

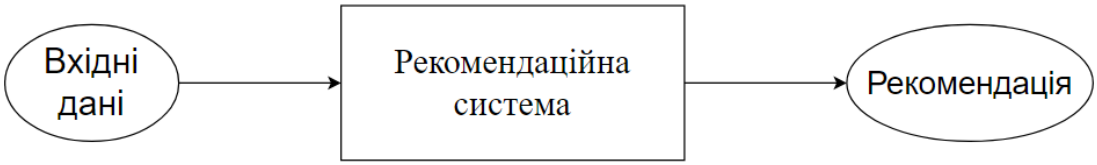

Рисунок 1.1 – Модель типу «чорна скринька»

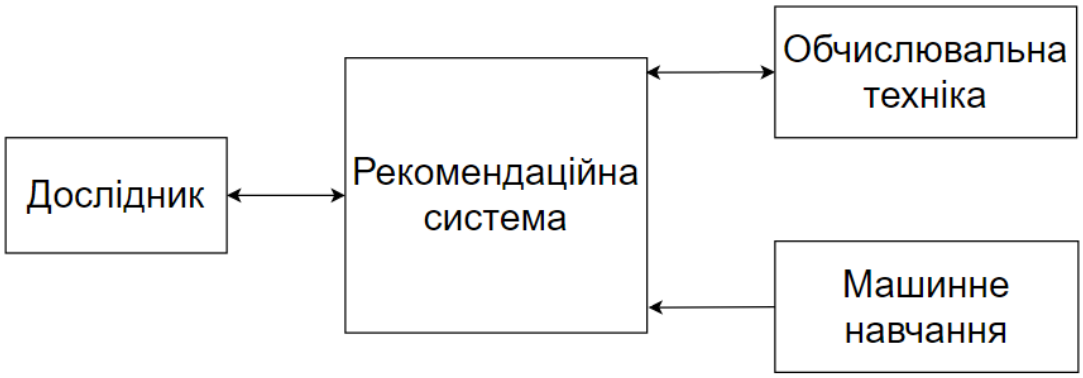

Рисунок 1.2 – Схема «система – зовнішнє середовище»

1.1.2 Функціональна модель системи

Методологія IDEF0 застосовується для функціонального моделювання різноманітних процесів і об’єктів. Основна концепція IDEF0 заснована на аналізі кожної системи як набору взаємопов’язаних та взаємодіючих компонентів. Це дозволяє візуалізувати та аналізувати процеси, які мають місце в рамках системи.

На першому етапі розробки моделі створюється контекстна діаграма, що ілюструє загальну функціональність системи та її взаємодію з зовнішнім середовищем. Така діаграма дає змогу оцінити, як система представлена в контексті ширшої системи взаємин.

Вхідними даними для системи є перелік пісень, які подобаються користувачу. Алгоритми машинного навчання без вчителя використовуються як основний механізм керування процесами всередині системи. Програмне забезпечення та дослідник виступають як допоміжні механізми, що забезпечують необхідну підтримку для функціонування системи. Результатом роботи системи є список пісень, які можуть зацікавити користувача. Контекстна діаграма процесу дослідження рекомендаційної системи для аудіоконтенту показана на рисунку 1.3.

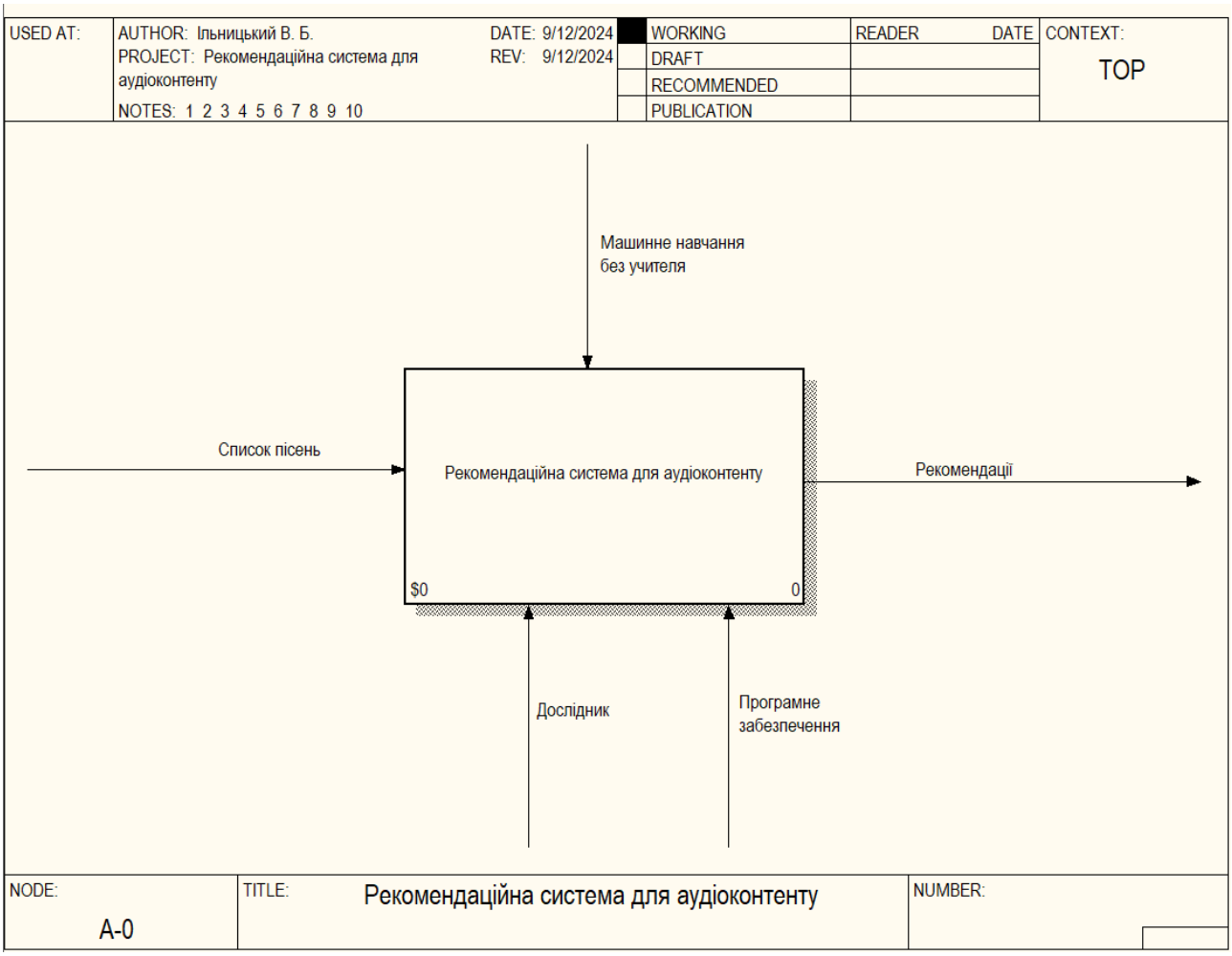

Рисунок 1.3 – Контекстна діаграма (рівень A-0)

Наступний етап використання методології IDEF0 полягає у декомпозиції, що включає детальний розподіл контекстної діаграми на більш глибокі та деталізовані частини. Цей процес має вирішальне значення, оскільки він допомагає виявити внутрішню структуру функціонального блоку та подальшу сегмента-

цію на дрібніші частини. Як результат, ми отримуємо глибокий і детальний опис системи на кожному рівні декомпозиції, що забезпечує високоякісний результат завдяки точній деталізації.

Для аналізу рекомендаційної системи аудіоконтенту ми використовуємо наступний підхід: вибір алгоритмів машинного навчання без учителя, налаштування параметрів цих алгоритмів, розробка програмного коду за допомогою бібліотеки scikit-learn та запуск процесу тренування моделі. На рисунку 1.4 показано деталізацію контекстної діаграми, що демонструє всі ключові компоненти процесу та сприяє досягненню необхідного рівня деталізації.

На рисунку 1.5 зображено декомпозицію роботи «Вибір алгоритмів машинного навчання без учителя».

На рисунку 1.6 зображено декомпозицію роботи «Налаштування параметрів алгоритмів».

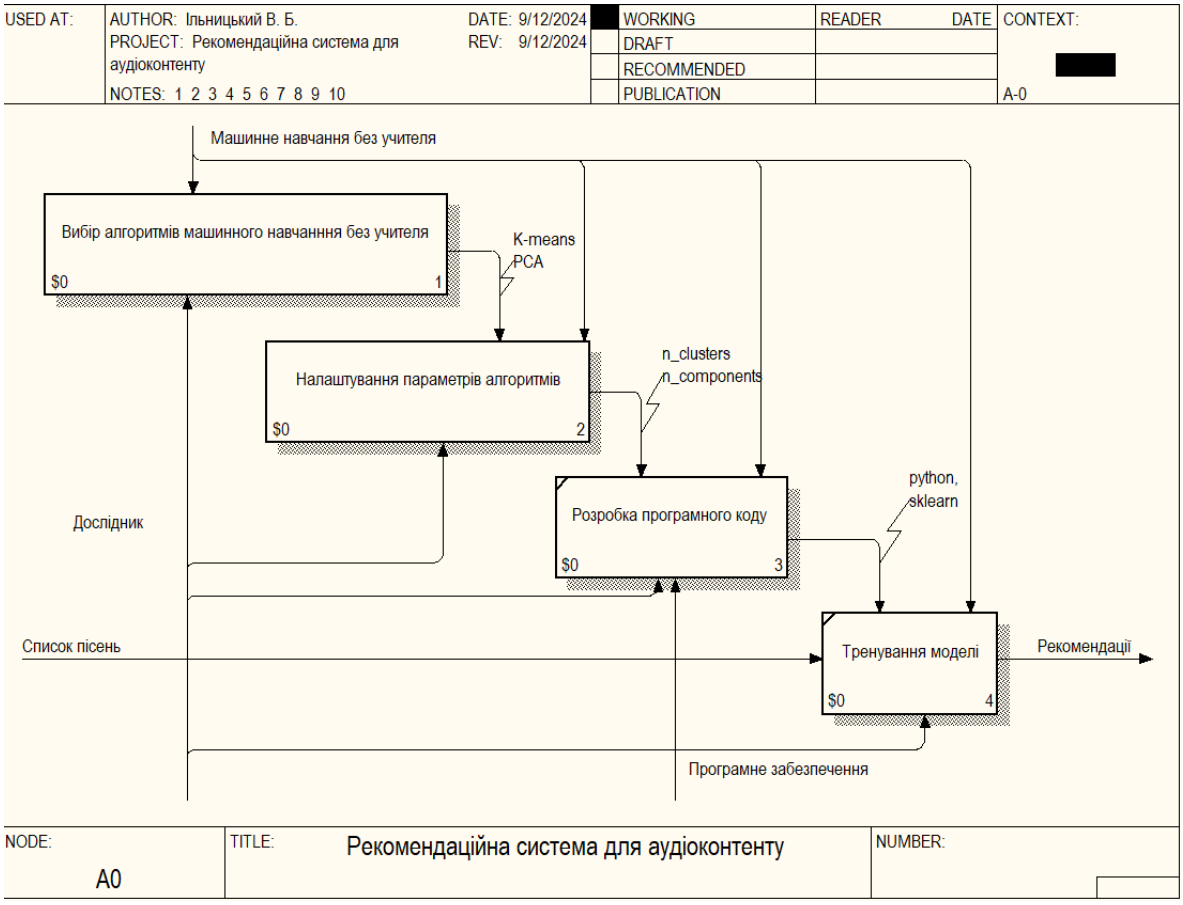

Рисунок 1.4 – Декомпозиція роботи «Рекомендаційна система для аудіоконтенту»: рівень A0

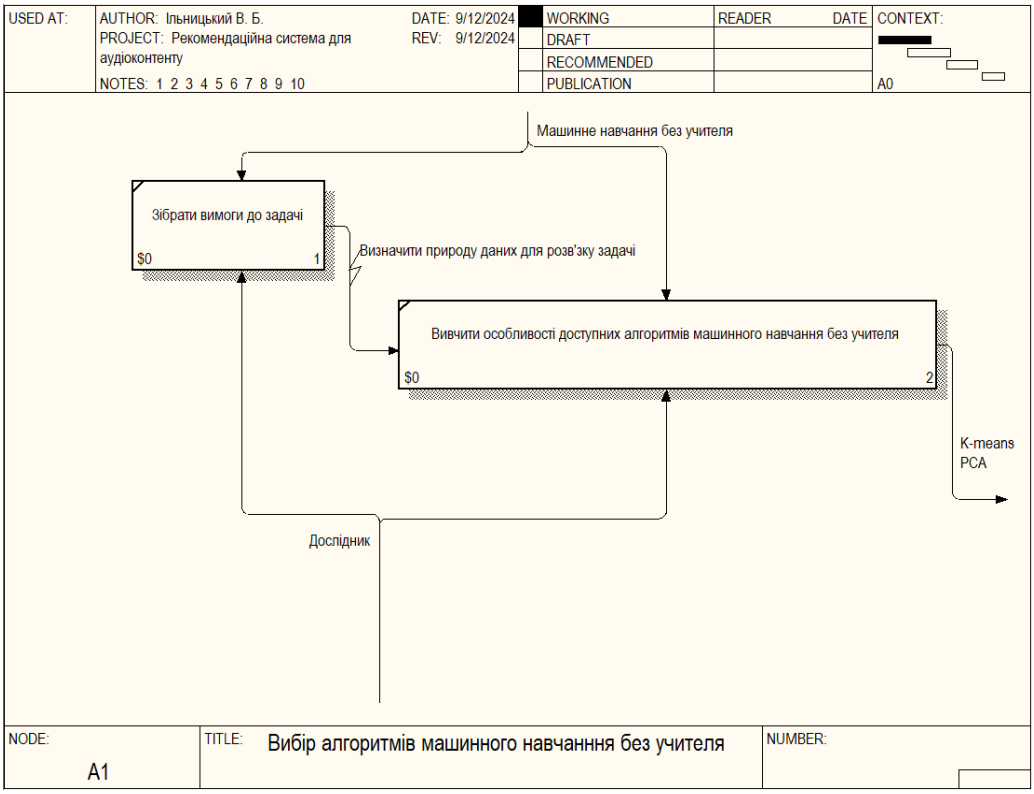

Рисунок 1.5 – Декомпозиція роботи «Вибір алгоритмів машинного навчання без учителя» рівень А1

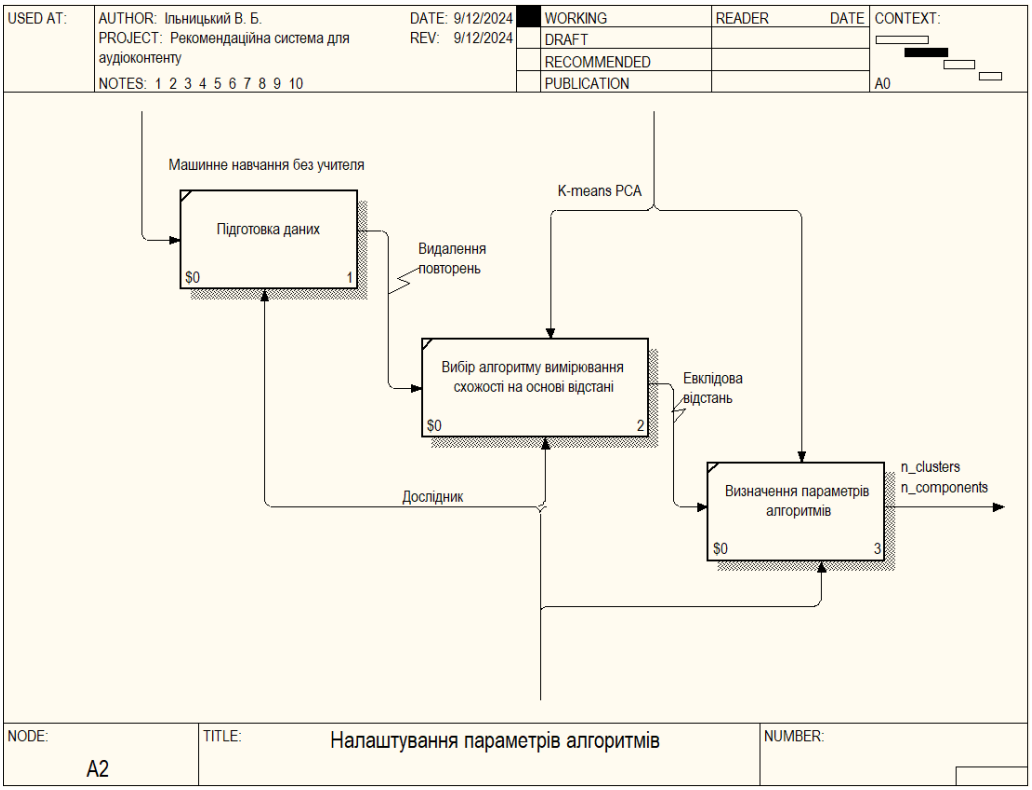

Рисунок 1.6 – Декомпозиція роботи «Налаштування параметрів алгоритмів» рівень А1

1.2 Аналіз сценаріїв побудови рекомендаційної системи

1.2.1 Модель аналізу системи

Задача побудови рекомендаційної системи полягає у визначенні найбільш релевантних предметів для користувача та заснована на його попередніх вподобаннях і поведінці. Це складна задача оптимізації, адже кількість можливих рекомендацій зростає експоненціально зі збільшенням кількості предметів та користувачів. Таким чином, якщо база даних містить велику кількість предметів. Традиційні методи рекомендацій, які базуються на повному переборі всіх можливостей, стають непридатними через величезні обсяги часу, необхідні для обробки.

Для вибору методу побудови рекомендаційної системи можна використати метод аналізу ієрархій. Основна ідея цього методу полягає у розбитті складної задачі на більш прості підзадачі (декомпозиція) та представленні їх у вигляді ієрархій. Процес вирішення задачі можна організувати наступним чином:

– формулювання проблеми прийняття рішень;

– розбиття проблеми на підзадачі;

– встановлення критеріїв для вибору рекомендацій;

– аналіз альтернативних варіантів рекомендацій;

– розробка ієрархічної репрезентації проблеми;

– побудова матриць переваг та проведення експертної оцінки;

– обчислення локального вектора переваг;

– оцінка суджень експертів;

– визначення вектора глобальних пріоритетів;

– на основі аналізу результатів вибір методу для вирішення проблеми.

Критерії вибору методу побудови рекомендаційної системи включають:

– критерій 1 (К1): релевантність рекомендацій;

– критерій 2 (К2): швидкість видачі рекомендацій;

– критерій 3 (К3): обчислювальна складність системи.

Проведемо аналіз альтернатив вирішення проблеми.

Застосування машинного навчання дозволяє рекомендаційним системам вчитися зі збільшенням кількості даних, покращуючи точність та релевантність рекомендацій. Алгоритми безперервно аналізують поведінку користувачів, оптимізуючи прогнози для кожного з них [4].

Один з найпопулярніших підходів до побудови рекомендаційних систем – колаборативна фільтрація. Цей підхід використовує аналіз поведінки користувачів та виявленні схожості між ними. Він дозволяє рекомендувати аудіоконтент, який сподобався схожим користувачам. Основними перевагами є здатність виявляти новий контент без необхідності його заздалегідь аналізувати. Однак, цей метод має недоліки, такі як «холодний старт» для нових користувачів та обмеження в рекомендаціях при недостатньому обсязі даних [5, 6].

Другий за популярністю підхід – фільтрація на основі вмісту. Цей підхід базується на аналізі характеристик самого аудіоконтенту, таких як жанр, артисти, бітрейт тощо. Система визначає контент, схожий на той, який користувач вже оцінив позитивно, і рекомендує його. Подібність може ґрунтуватися на атрибутах товарів, які подобаються користувачеві [7, 8].

Отже, в той час як колаборативна фільтрація використовує рейтинги інших користувачів і цільового користувача, в рекомендаційних системах фільтрації на основі вмісту використовуються рейтинги і атрибути пісень, які сподобалися цільовому користувачеві. Таким чином, інші користувачі не відіграють великої ролі в системах на основі контенту, оскільки вони використовують інше джерело даних для рекомендацій. Перевага цього методу полягає у можливості рекомендувати контент новим користувачам без історії їхніх вподобань. Проте, цей метод може бути обмеженим, якщо контент має недостатньо описових характеристик або є суб’єктивними в оцінці [9].

Гібридні системи поєднують колаборативну фільтрацію та фільтрацію на основі вмісту, намагаючись об’єднати їхні переваги та мінімізувати недоліки. Такий підхід може забезпечити більш точні та релевантні рекомендації, а також краще впоратись з проблемою «холодного старту». Гібридні системи можуть

використовувати різні техніки, такі як вагові коефіцієнти, злиття прогнозів або послідовне застосування різних методів.

Також варто зазначити, що додавання контекстуальної інформації, такої як час доби, локація користувача, або навіть психологічний стан, може допомогти рекомендаційним системам зробити більш точні та відповідні пропозиції.

Кожен із цих сценаріїв має свої сильні та слабкі сторони. Вибір конкретного підходу залежить від специфіки задачі, доступних даних та цілей системи. Аналіз цих сценаріїв дозволяє зрозуміти, які методи найкраще підходять для вирішення конкретних завдань у рамках розробки рекомендаційної системи для аудіоконтенту.

Таким чином, маємо наступну множину альтернатив:

– альтернатива 1 (A1): колаборативна фільтрація;

– альтернатива 2 (A2): фільтрація на основі вмісту;

– альтернатива 3 (A3): гібридна система.

На рисунку 1.7 зображена ієрархічна модель проблеми вибору.

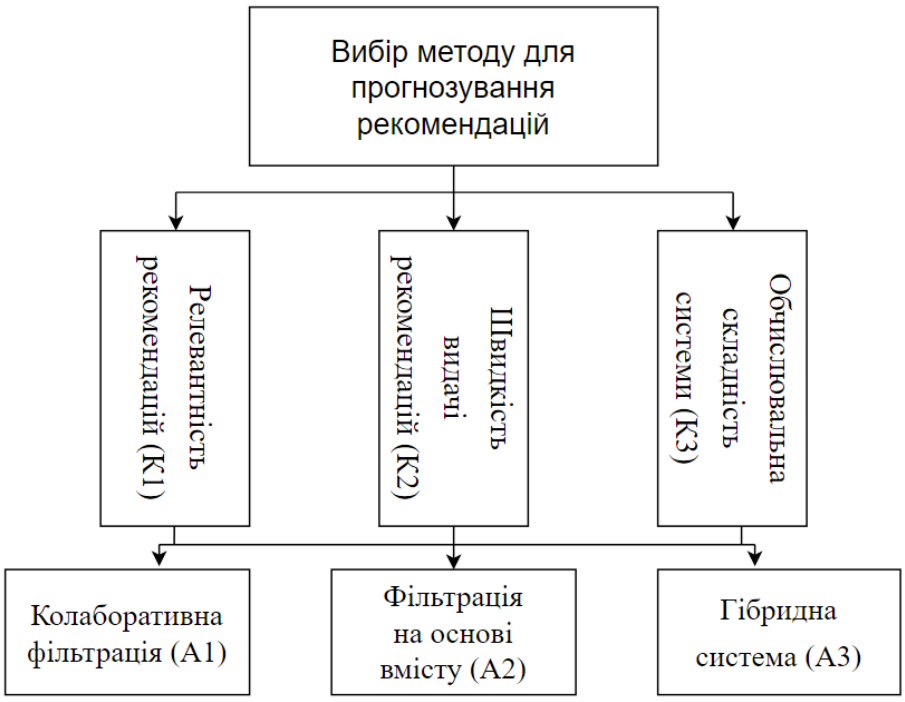

Рисунок 1.7 – Ієрархічна модель аналізу проблеми

1.2.2 Оцінювання вектора пріоритетів незадоволеностей методом аналізу ієрархій

Спочатку потрібно сформулювати матрицю парних порівнянь критеріїв та альтернатив за кожним з критеріїв. Шляхом експертного опитування отримали дані, які наведено у таблиці 1.1.

Таблиця 1.1 – Матриця парних порівнянь критеріїв

Критерії оцінювання	К1	К2	К3	Оцінка компонентів	Вектор пріоритетів
К1	1	$\frac{1}{4}$	$\frac{1}{6}$	0,345	0,081
К2	4	1	$\frac{1}{4}$	1,000	0,236
К3	6	4	1	2,884	0,681
Усього				4,231	1

Знаходимо власний вектор $\lambda_{\max}$ за формулою $\lambda_{\max} \approx \sum_{j=1}^{3} y_j p_j$, де $y_j = \sum_{i=1}^{3} a_{ij}$ – сума елементів j-го стовпчика. Отримуємо, що $\lambda_{\max} \approx 3{,}107$. Індекс узгодженості $CI = \frac{\lambda_{\max} - 3}{3-1} = 0{,}053$, випадковий індекс ($RI$) для матриць третього порядку дорівнює $0{,}58$. Тоді відносна узгодженість дорівнює $CR = \frac{CI}{RI} = \frac{0{,}053}{0{,}58} = 0{,}092$. Оскільки відносна узгодженість менша за $0{,}2$, то це свідчить про задовільність логічних суджень експерта.

Формуємо матриці попарних порівнянь альтернатив за кожним критерієм.

Таблиця 1.2 – Матриця парних порівнянь альтернатив за першим критерієм

К1	А1	А2	А3	Оцінка компонентів	Вектор пріоритетів
А1	1	2	7	2,410	0,651
А2	$\frac{1}{2}$	1	$\frac{1}{2}$	0,629	0,170
А3	$\frac{1}{7}$	2	1	0,658	0,178
Усього				3,698	1

Знаходимо, що $\lambda_{\max} \approx 3,43$. Індекс узгодженості $CI = \frac{\lambda_{\max} - 3}{3-1} = 0,217$. Тоді відносна узгодженість дорівнює $CR = \frac{CI}{RI} = \frac{0,216}{0,58} = 0,375$. Відносна узгодженість менша за $0,2$, що свідчить про задовільність логічних суджень експерта.

Таблиця 1.3 – Матриця парних порівнянь альтернатив за другим критерієм

К1	А1	А2	А3	Оцінка компонентів	Вектор пріоритетів
А1	1	2	$\frac{1}{5}$	0,736	0,196
А2	$\frac{1}{2}$	1	$\frac{1}{3}$	0,550	0,146
А3	5	3	1	2,466	0,657
Усього				3,753	1

Знаходимо, що $\lambda_{\max} \approx 3,163$. Індекс узгодженості $CI = \frac{\lambda_{\max} - 3}{3-1} = 0,081$. Тоді відносна узгодженість дорівнює $CR = \frac{CI}{RI} = 0,14$. Відносна узгодженість менша за $0,2$, що свідчить про задовільність логічних суджень експерта.

Таблиця 1.4 – Матриця парних порівнянь альтернатив за третім критерієм

К1	А1	А2	А3	Оцінка компонентів	Вектор пріоритетів
А1	1	2	$\frac{1}{3}$	0,873	0,249
А2	$\frac{1}{2}$	1	$\frac{1}{3}$	0,550	0,157
А3	3	3	1	2,080	0,593
Усього				3,504	1

Знаходимо, що $\lambda_{max} = 3,05$. Індекс узгодженості $CI = \frac{\lambda_{max} - 3}{3-1} = 0,02$. Тоді відносна узгодженість дорівнює $CR = \frac{CI}{RI} = \frac{0,02}{0,58} = 0,04$. Відносна узгодженість менша за $0,2$, що свідчить про задовільність логічних суджень експерта.

1.2.3 Модель вирішення проблеми

Отримавши вектори локальних переваг можемо побудувати вектор глобальних переваг. Для цього складемо матрицю, де стовпці є векторами локальних пріоритетів. У таблиці 1.5 наведено результати розрахунків.

Таблиця 1.5 – Узагальнені пріоритети альтернатив

К / А	К1	К2	К3	Узагальнені пріоритети
А1	0,651	0,196	0,249	0,081
А2	0,170	0,146	0,157	0,236
А3	0,178	0,657	0,593	0,681

Тоді вектор глобальних пріоритетів матиме вигляд:

$$\vec{p} = \begin{pmatrix} 0,651 & 0,196 & 0,249 \\ 0,170 & 0,146 & 0,157 \\ 0,178 & 0,657 & 0,593 \end{pmatrix} \cdot \begin{pmatrix} 0,081 \\ 0,236 \\ 0,681 \end{pmatrix} =$$

$$= (0,269;\ 0,155;\ 0,574)^T .$$

Розрахуємо індекс узгодженості та відношення узгодженості для всієї ієрархії:

$$CI = CI^K + (\vec{p}^K, CI^K) =$$

$$= 0,053 + 0,217 \cdot 0,081 + 0,081 \cdot 0,236 + 0,02 \cdot 0,681 = 0,10,$$

$$RI = RI^K + RI^A = 0,58 + 0,58 = 1,16,$$

$$CR = \frac{CI}{RI} = \frac{0,10}{1,16} = 0,089.$$

Оскільки ми є особою, що приймає рішення, то на основі вектора глобальних пріоритетів можемо зробити наступний висновок: для розв'язання задачі розробки рекомендаційної необхідно обрати третю альтернативу – гібридну систему.

1.3 Змістовна та формальна постановки задачі

1.3.1 Змістовна постановка задачі

Задача розробки рекомендаційної системи виникає при необхідності забезпечення користувачів персоналізованими пропозиціями, які відповідають їхнім індивідуальним інтересам та вподобанням. Система повинна аналізувати існуючі дані про поведінку та вподобання користувачів, використовуючи інфо-

рмацію про їхні попередні дії, такі як перегляди або оцінки.

В основі рекомендаційної системи лежить математична модель, яка може бути виражена через матрицю користувач-предмет, де кожен елемент матриці відображає певну взаємодію між користувачем і об’єктом. Заповнення пропусків у цій матриці за допомогою прогнозування невідомих значень є однією з основних задач системи.

Виклик полягає в тому, щоб знайти спосіб ефективно обробляти ці дані та забезпечити високу точність прогнозування рекомендацій, які будуть релевантними та цікавими кожному користувачу. Система має враховувати різноманіття інтересів серед великої кількості користувачів та бути здатною адаптуватися до змін у їхніх перевагах з часом.

1.3.2 Формальна постановка задачі

Обраний гібридний метод складається з двох підзадач, які розв’язуються послідовно.

Постановка задачі для фільтрації на основі контенту має наступний вигляд.

Нехай X – множина об’єктів. Задана тренувальна вибірка об’єктів. $X_n = \{\vec{x}^{(i)}\}_{i=1}^{n} \subset X$ та функція відстані між об’єктами $d(\vec{x}, \vec{x}')$.

Розглядається задача навчання без учителя – мітки $y^{(i)}$ об’єктів $\vec{x}^{(i)} \in X_n$ не задані.

Необхідно побудувати алгоритм a, що дозволяє розбити вибірку на множини, які не перетинаються – кластери, так, щоб кожен кластер складався з об’єктів, близьких за метрикою d, а об’єкти різних кластерів суттєво відрізнялись.

Це означає, що кожному об’єкту $\vec{x}^{(i)} \in X_n$ необхідно поставити у відповідність мітку кластера $\vec{y}^{(i)} \in Y$, де Y – деяка множина номерів (міток) кластерів.

Алгоритм кластеризації – це функція $a: X \to Y$, яка будь-якому об'єкту $\vec{x}$ ставить у відповідність номер кластера $y \in Y$.

Множина Y у деяких задачах відома заздалегідь, однак частіше вона невідома і ставиться задача визначити оптимальну кількість кластерів.

В ході кластеризації об'єкти групуються, виходячи з подібності ознак. Під схожістю об'єктів розуміється близькість їх один до одного у сенсі обраної метрики.

Постановка задачі колаборативної фільтрації має наступний вигляд.

Нехай R – матриця взаємодій розміром $m \times n$, де m – кількість користувачів, n – кількість пісень. Тоді $R_{ij} = 1$ – користувач i вподобав пісню j і $R_{ij} = 0$ – користувач i не вподобав пісню j або не взаємодіяв з нею.

Визначаємо косинусну схожість між користувачами i та j. Будуємо матрицю схожості користувачів S, де кожен елемент матриці S відповідає схожості між користувачами i та j.

Для рекомендацій потрібно обчислити оцінку для кожної пісні, яка не взаємодіяла з користувачем u. Оцінка для пісні j користувача u обчислюється як зважене середнє взаємодій інших користувачів з цією піснею.

1.4 Постановка задач дослідження

Провівши системний аналіз предметної області – задачі розробки рекомендаційних систем, можна дійти висновку, що для ефективного формування персоналізованих пропозицій користувачам рекомендується використовувати комбіновані системи рекомендацій, а саме гібридні системи.

Отже, метою кваліфікаційної роботи є впровадження гібридного підходу, що поєднує фільтрацію на основі вмісту та колаборативну фільтрацію, для підвищення точності та релевантності рекомендацій. Для досягнення поставленої мети необхідно виконати наступні завдання:

– провести аналіз сучасного стану методів розробки рекомендаційних систем та ознайомитися з особливостями архітектури та реалізації гібридних систем;

– застосувати гібридний метод до розв’язання задачі рекомендаційної системи та скласти відповідний алгоритм;

– програмно реалізувати модель гібридної системи на мові програмування Python за допомогою бібліотеки scikit-learn;

– провести навчання моделі на відповідному датасеті;

– провести низку обчислювальних експериментів та проаналізувати їх результати.

2 ВИБІР ТА ОБҐРУНТУВАННЯ МЕТОДУ РОЗВ'ЯЗАННЯ

2.1 Фільтрація на основі вмісту

Для застосування фільтрації на основі вмісту зазвичай використовують метод кластеризації k-means. Цей метод дозволяє групувати об'єкти (наприклад, фільми, книги, музику тощо) на основі їхніх характеристик або властивостей.

Кластеризація k-means інакше називається алгоритмом машинного навчання без учителя. Приписка «без учителя» означає, що ми не даємо алгоритму еталонних прикладів того, як повинні виглядати правильні пари входу-виходу. Цей алгоритм також є параметричним, оскільки для його роботи потрібен параметр k [10].

Основна ідея полягає в тому, що об'єкти, які мають схожі характеристики, згруповуються разом у кластери. Кількість кластерів k задається заздалегідь, і алгоритм k-means намагається мінімізувати внутрішньокластерну варіабельність, тобто зробити об'єкти в межах одного кластера максимально схожими одне на одного, а об'єкти з різних кластерів – максимально різними.

Процес кластеризації за допомогою k-means включає наступні кроки:

а) задати кількість кластерів k;

б) із початкової множини даних навмання обрати k спостережень, які будуть центрами початкових кластерів (початковими центроїдами);

в) для кожного зразка визначити, до якого з центроїдів він розташований ближче за все, і віднести його до цього кластера;

г) оновити центроїди сформованих кластерів за формулою:

$$\vec{\mu}^{(j)} = \frac{1}{\left|C_j\right|} \sum_{\vec{x}^{(i)} \in C_j} \vec{x}^{(i)}, \; j \in 1,2,...,k,$$

де $\left|C_j\right|$ – кількість об’єктів, що потрапили у j-й кластер C_j;

$\vec{\mu}$ – центроїда;

г) повторювати кроки в) і г), поки вміст кластерів не перестане змінюватися від ітерації до ітерації або не буде досягнутий заданий допуск або максимальна кількість ітерацій.

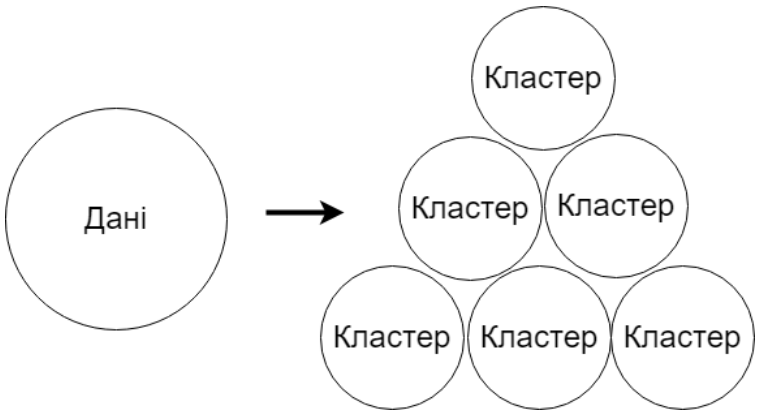

Рисунок 2.1 – Дані після кластеризації

Після завершення кластеризації, система може рекомендувати користувачеві об’єкти з того ж кластера, до якого належать об’єкти, які вже сподобалися цьому користувачеві. Це дозволяє зробити рекомендації більш персоналізованими і точними.

Фільтрація на основі вмісту з використанням k-means є ефективним способом знаходження схожих об’єктів і може бути застосована в різних областях, де є потреба у групуванні та рекомендації вмісту.

Метод k-means може бути поданий як задача ітеративної мінімізації внутрішньокластерної суми квадратичних помилок SSE (2.1):

$$SSE = \sum_{i=1}^{n} \sum_{j=1}^{k} \omega_{ij} \left\| \vec{x}^{(i)} - \vec{\mu}^{(j)} \right\|_2^2, \tag{2.1}$$

де $\vec{\mu}^{(j)}$ – центроїд кластера C_j.

Якщо $\vec{x}^{(i)} \in C_j$, то $\omega_{ij} = 1$, а інакше $\omega_{ij} = 0$.

Метод k-means має свої переваги та недоліки. До переваг можна віднести швидкість та простоту реалізації, а також зрозумілість та прозорість алгоритму.

Однак метод має і недоліки. По-перше, необхідно вручну задавати кількість кластерів k. По-друге, результат сильно залежить від вибору початкових центроїдів. Також метод працює повільно на великих наборах даних і погано справляється з кластерами несферичної форми [11].

Після кластеризації, для кожного користувача формується профіль на основі пісень, які він вже оцінив. Цей профіль представляє середній вектор характеристик пісень, які сподобалися користувачу. Далі, цей вектор використовується для визначення найближчого кластера в просторі характеристик пісень. Пісні, що належать до цього кластера, вважаються потенційно привабливими для користувача, оскільки вони мають схожі характеристики до тих, що вже сподобалися.

На завершальному етапі, з цього вибраного кластера пісень вибираються кілька пісень, які рекомендуються користувачу.

Схематично цей процес зображено на рисунку 2.2.

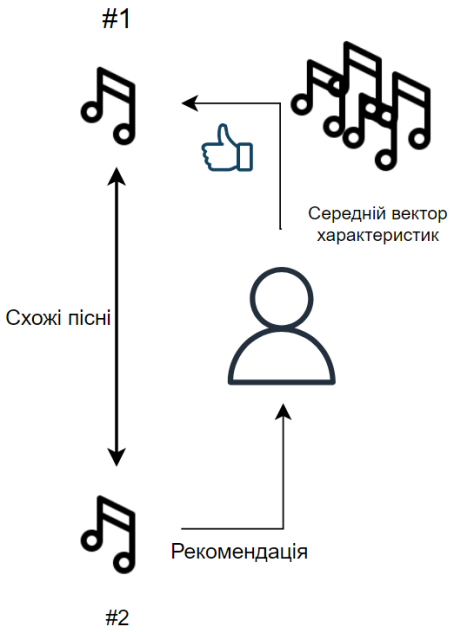

Рисунок 2.2 – Фільтрація на основі вмісту

2.2 Колаборативна фільтрація

Колаборативна фільтрація за елементами, або фільтрація від елемента до елемента – це форма колаборативної фільтрації для рекомендаційних систем, заснована на схожості між елементами, що розрахована на основі оцінок цих елементів користувачами.

Колаборативну фільтрацію на основі елементів було винайдено і використано Amazon.com у 1998 році. Вперше вона була опублікована на науковій конференції у 2001 році [12].

Більш ранні системи спільної фільтрації, засновані на оцінці схожості між користувачами (відомі як фільтрація «користувач-користувач»), мали кілька проблем:

– системи працювали погано, коли у них було багато об’єктів, але порівняно мало оцінок;

– обчислення схожості між усіма парами користувачів вимагало значних обчислювальних складнощів;

– профілі користувачів швидко змінювалися і всю модель системи доводилося перераховувати заново.

Моделі фільтрації від елемента до елемента вирішують ці проблеми в системах, де користувачів більше, ніж об’єктів. Моделі «елемент-елемент» використовують розподіл оцінок для кожного елемента, а не для кожного користувача. Це призводить до більш стабільного розподілу оцінок у моделі, тому модель не потрібно перебудовувати так часто. Коли користувачі слухають, а потім оцінюють пісню, схожі пісні вибираються з існуючої моделі системи і додаються до рекомендацій користувача.

Спочатку система виконує етап побудови моделі, знаходячи схожість між усіма парами елементів. Ця функція подібності може мати різні форми, наприклад, кореляцію між оцінками або косинус векторів оцінок (2.2). Як і в системах «користувач-користувач», функції подібності можуть використовувати нормалізовані оцінки:

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

Далі система виконує етап рекомендацій. Вона використовує найбільш схожі об'єкти з уже оціненими користувачем об'єктами, щоб сформувати список рекомендацій. Зазвичай цей розрахунок є зваженою сумою або лінійною регресією. Ця форма рекомендацій є аналогом «людям, яким сподобалась пісня X, як і вам, також подобається пісня Y, яку ви ще не оцінили, тож ви отримаєте рекомендацію».

Схематично цей процес зображено на рисунку 2.3.

Рисунок 2.3 – Колаборативна фільтрація

2.2 Використання та архітектура гібридної системи для розробки рекомендаційної системи для аудіоконтенту

Архітектура моделі навчання без учителя, яка використовує як методи фільтрації на основі вмісту, так і колаборативну фільтрацію, може бути зображена як інтегрована система, що складається з кількох ключових компонентів:

а) на початку маємо датасет, що містить як метадані об’єктів (характеристики пісень);

б) етап попередньої обробки даних включає очищення даних, нормалізацію, а також зменшення розмірності за допомогою методу PCA (аналіз головних компонент);

в) модуль фільтрації на основі вмісту:

1) використання методу кластеризації k-means;

2) пошук схожих пісень використовуючи евклідову метрику;

г) модуль колаборативної фільтрації:

1) створення матриці, яка відображає взаємодії між користувачами та об’єктами;

2) пошук схожих пісень використовуючи косинус подібності;

г) комбінування рекомендацій, отриманих з обох модулів, для формування кінцевого списку рекомендованих об’єктів;

д) представлення кінцевих рекомендацій користувачеві через користувацький інтерфейс.

Ця архітектура дозволяє використовувати переваги обох підходів: фільтрації на основі вмісту для врахування унікальних характеристик об’єктів та колаборативної фільтрації для використання колективної схожості користувачів.

Такий підхід може забезпечити більш точні та персоналізовані рекомендації.

Важливо зазначити, що ефективність цієї архітектури значною мірою залежить від якості та обсягу даних, а також від точності алгоритмів, використаних у кожному модулі.

2.3 Алгоритм застосування фільтрації на основі вмісту

Розглядається n пісень, які мають числові характеристики, а саме: популярність, танцювальність, енергетичність, інструментальність, гучність та багато інших.

Нехай X – датасет. Тоді для етапу обробки даних потрібно сформувати вектор центру інтересів користувача (v_{user}) для пісень, які він вподобав:

$$X = [x_i],$$

$$v_{user} = \frac{1}{n}\left(\sum_{j=1}^{n} x_j \right),$$

де x_i – пісня з датасету;

x_j – пісня, яку вподобав користувач.

Наступним етапом йде нормалізація даних. Для цього використовується об’єкт StandardScaler з бібліотеки scikit-learn:

$$X_{scaled} = \frac{x_i - u}{s},$$

де u – середнє значення X;

s – стандартне відхилення X.

Цей етап також потрібно повторити для того, щоб нормалізувати вектор центру інтересів користувача.

Далі йде етап обчислення відстаней між вектором центру інтересів користувача та всіма піснями у нормалізованому датасеті.

Для цього використовується евклідова метрика:

$$D = euclidean\_distances\left(X_{scaled},\ v_{user}\right) = \sqrt{\sum_{i=1}^{n}\left(X_i - v_i\right)^2}\ .$$

Останнім етапом йде вибір n пісень для рекомендації на основі найменших відстаней:

$$x_{recommended} = \arg\min_{\{x_1, x_2, \ldots, x_n\}} \subseteq D\ .$$

2.4 Алгоритм застосування колаборативної фільтрації

Алгоритм колаборативної фільтрації для побудови рекомендаційної системи базується на пошуку схожих користувачів і передбаченні вподобань користувача на основі вподобань схожих користувачів.

На першому етапі потрібно підготовити дані. Нехай R – матриця вподобань користувачів до пісень, де кожен рядок відповідає користувачеві, а кожен стовпець відповідає пісні. Елемент матриці r_{ui} вказує, чи користувач u вподобав пісню i, де:

$$r_{ui} = \begin{cases} 1, & \text{якщо користувач } u \text{ вподобав пісню } i; \\ 0, & \text{якщо користувач } u \text{ не вподобав пісню } i. \end{cases}$$

Другий етап допомагає визначити схожості користувачів. Для визначення схожості між користувачами використовується косинусна подібність:

$$similarity(u, v) = 1 - \cos(\theta_{uv}) = 1 - \frac{R_u \cdot R_v}{\|R_u\| \cdot \|R\|},$$

де R_u та R_v – вектори вподобань користувачів u та v;

$\cos(\theta_{uv})$ – косинус кута між векторами вподобань.

Це означає, що схожість між користувачами базується на тому, наскільки схожі їхні вподобання щодо пісень.

Третій етап відповідає за пошук найбільш схожих користувачів. Після обчислення схожості ми можемо знайти користувачів, схожих на даного користувача. Для цього просто сортуємо індекси користувачів на основі значення схожості.

Останній етап ми формуємо рекомендації. Для того, щоб сформувати рекомендації, потрібно знайти пісні, які були вподобані схожими користувачами, але ще не були прослухані поточним користувачем. Це означає, що ми шукаємо пісні, для яких $r_{ui} = 0$ для даного користувача u, але $r_{vi} = 1$ для схожого користувача v:

$$recommend(u) = \{i \mid r_{ui} = 0 \text{ і } \exists v : r_{vi} = 1\}.$$

Цей алгоритм дозволяє на основі схожості користувачів рекомендувати нові пісні користувачеві, що ще не прослухав їх, але ймовірно, вони йому сподобаються, оскільки їх уподобали схожі користувачі.

Висновки за розділом 2

У цьому розділі було продемонстровано загальний підхід до розробки рекомендаційної системи для аудіоконтенту за допомогою машинного навчання без учителя. Процес розробки можна розділити на чотири основні етапи.

1. Підготовка даних. Дані проходять попередню обробку, включаючи очищення, нормалізацію та зменшення розмірності за допомогою PCA.

2. Фільтрація на основі вмісту. Для цього використовується метод кластеризації k-means для групування схожих об’єктів, а також обчислення схожості

між ними за допомогою евклідової метрики. На основі інтересів користувача система шукає пісні з подібними характеристиками.

3. Колаборативна фільтрація. Система будує матрицю взаємодій користувачів з об’єктами та обчислює схожість між користувачами, використовуючи косинус подібності. Це дозволяє рекомендувати користувачу пісні, які вподобали схожі на нього користувачі.

4. Інтеграція результатів. Рекомендації з обох модулів (фільтрація на основі вмісту та колаборативна фільтрація) комбінуються для створення остаточного списку рекомендованих пісень. Цей підхід дозволяє врахувати як особливості самих об’єктів, так і вподобання інших користувачів.

Розглянутий підхід забезпечує кращу точність і персоналізацію, оскільки враховує як індивідуальні вподобання користувача, так і колективні вподобання схожих користувачів.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1 Python як універсальний інструмент для машинного навчання

Python відіграє ключову роль у сфері машинного навчання. Його універсальність та потужна екосистема бібліотек роблять його незамінним інструментом для дослідників та розробників у галузі штучного інтелекту та аналізу даних.

В освіті та навчанні Python став де-факто стандартом для викладання машинного навчання в університетах та онлайн-курсах. Мова програмування Python має відносно простий синтаксис, що робить її більш доступною для новачків і зручною для досвідчених розробників. Ця особливість дозволяє швидко імплементувати ідеї та реалізовувати складні алгоритми машинного навчання без надмірної складності кодування.

Гнучкість та інтеграція Python є однією з його ключових переваг у сфері машинного навчання. Python легко взаємодіє з іншими мовами програмування та технологіями, що дозволяє створювати складні та ефективні системи. Наприклад, можна використовувати Python для розробки алгоритмів машинного навчання та аналізу даних, а потім інтегрувати ці компоненти з високопродуктивними модулями, написаними на C++ або Java. Така гнучкість особливо корисна в великих проектах, де потрібно оптимізувати різні аспекти системи. Python також має інструменти для взаємодії з різними базами даних, API та вебсервісами, що робить його ідеальним для створення комплексних рішень машинного навчання, які можуть легко інтегруватися в існуючі інфраструктури.

У промисловому застосуванні Python став стандартом для багатьох компаній, які впроваджують системи машинного навчання у виробництво. Від стартапів до великих корпорацій Python використовується для створення широкого спектру продуктів та сервісів, заснованих на машинному навчанні. Це включає рекомендаційні системи для електронної комерції та стрімінгових платформ, системи комп’ютерного зору для автономних транспортних засобів, алгоритми

обробки природної мови для чат-ботів та віртуальних асистентів, а також складні аналітичні інструменти для фінансового сектору.

Підтримка спільноти Python є однією з найбільших та найактивніших у світі програмування. Це особливо помітно в галузі машинного навчання, де постійно з'являються нові бібліотеки, інструменти та ресурси. Спільнота Python регулярно організовує конференції, вебінари та зустрічі, де обговорюються останні досягнення та тенденції в машинному навчанні. Крім того, існує велика кількість онлайн-форумів, блогів та репозиторіїв з відкритим кодом, де розробники діляться своїм досвідом та кодом. Ця активна екосистема забезпечує постійний приплив нових ідей та рішень, що допомагає Python залишатися на передовій технологій машинного навчання. Завдяки цій підтримці, розробники завжди мають доступ до найновіших методів та інструментів, що значно прискорює процес розробки та впровадження інновацій.

Масштабованість Python є критично важливою характеристикою для проектів машинного навчання, особливо в контексті рекомендаційних систем. Python дозволяє легко переходити від простих прототипів до складних, високонавантажених систем. Це досягається завдяки різноманітності доступних інструментів та фреймворків, які підтримують паралельні обчислення та розподілену обробку даних. Крім того, Python може бути легко інтегрований з технологіями контейнеризації, такими як Docker, що спрощує розгортання та масштабування додатків. Ця здатність до масштабування особливо важлива для рекомендаційних систем, які часто працюють з величезними наборами даних про користувачів та їхню поведінку, і повинні швидко адаптуватися до зростаючих обсягів інформації та кількості користувачів.

У майбутньому очікується, що роль Python у машинному навчанні буде лише зміцнюватися, оскільки нові бібліотеки та інструменти продовжують з'являтися, а існуючі – вдосконалюватися. Це робить Python не просто мовою програмування, а ключовим елементом екосистеми машинного навчання, що формує майбутнє технологій штучного інтелекту та аналізу даних.

3.2 Додаткові бібліотеки для Python

Pandas – це бібліотека для обробки та аналізу структурованих даних в Python. Вона значно розширює можливості Python у роботі з табличними даними, що є критично важливим для багатьох задач машинного навчання.

Основні переваги Pandas включають:

а) ефективну обробка великих наборів даних;

б) інструменти для очищення даних;

в) зручні методи для агрегації та групування даних;

г) інтеграцію з різними форматами даних;

ґ) часові ряди;

д) злиття та об’єднання даних.

У контексті машинного навчання Pandas часто використовується для попередньої обробки даних, підготовки ознак та аналізу результатів. Наприклад, при розробці рекомендаційної системи Pandas може бути використана для обробки користувацьких рейтингів, агрегації даних про взаємодії користувачів з елементами та підготовки даних для навчання моделей.

Scikit-learn (sklearn) є однією з найпопулярніших бібліотек машинного навчання для Python. Вона надає широкий спектр інструментів для різних задач машинного навчання, значно розширюючи можливості Python у цій галузі.

Основні переваги Scikit-learn включають:

а) широкий набір алгоритмів;

б) уніфікований інтерфейс;

в) інструменти для оцінки моделей;

г) препроцесинг даних;

ґ) вибір ознак;

д) генерація синтетичних даних.

Scikit-learn може бути використана для реалізації різних підходів, таких як колаборативна фільтрація або фільтрація на основі вмісту з використанням алгоритмів класифікації та регресії.

SciPy (Scientific Python) є фундаментальною бібліотекою для наукових обчислень в Python. Вона розширює можливості Python у сфері математичних операцій та алгоритмів, що часто використовуються в машинному навчанні.

Основні переваги SciPy включають:

а) алгоритми оптимізації;

б) операції лінійної алгебри;

в) широкий набір статистичних тестів та розподілів;

г) обробку сигналів та зображень;

ґ) інтерполяцію та апроксимацію.

NumPy (Numerical Python) є фундаментальною бібліотекою для наукових обчислень в Python, яка значно розширює можливості мови в роботі з багатовимірними масивами та матрицями. Ця бібліотека є основою для багатьох інших інструментів машинного навчання в Python.

Основні переваги NumPy включають:

а) ефективну роботу з багатовимірними масивами;

б) векторизовані операції;

в) математичні функції;

г) генерацію випадкових чисел.

NumPy може бути використана для ефективного зберігання та обробки матриць взаємодій користувачів з елементами, виконання матричних операцій при реалізації алгоритмів колаборативної фільтрації, а також для швидких обчислень при оцінці схожості елементів або користувачів.

Разом ці бібліотеки створюють потужну екосистему для розробки, навчання та розгортання моделей машинного навчання в Python. Вони дозволяють ефективно обробляти дані, виконувати складні математичні операції, реалізовувати та оцінювати моделі машинного навчання.

Ця комбінація робить Python одним з найпопулярніших інструментів для реалізації проектів машинного навчання, включаючи складні рекомендаційні системи.

3.3 Опис програми

Розроблена програма представляє собою сучасну, багатошарову систему рекомендації аудіоконтенту, яка поєднує в собі передові технології веб-розробки та машинного навчання. Розглянемо кожен аспект цієї системи детальніше.

Програма побудована на принципах багатошарової архітектури, що забезпечує чітке розділення відповідальності між різними компонентами системи. Це робить програму більш модульною, легшою для підтримки та масштабування.

Серверна частина (бекенд) системи розроблена з використанням двох потужних технологій: ASP.NET Core та Python. ASP.NET Core – це сучасний, високопродуктивний фреймворк для створення веб-додатків, який забезпечує основну структуру бекенду. Python використовується для реалізації складних алгоритмів машинного навчання, які лежать в основі рекомендаційної системи.

Клієнтська частина реалізована за допомогою Razor Pages – легкої та ефективної технології для створення динамічних веб-сторінок в рамках екосистеми ASP.NET Core. Це дозволяє створювати інтерактивний та зручний інтерфейс користувача, тісно інтегрований з бекендом.

Програма має наступну функціональність.

Аутентифікація користувачів реалізована за допомогою OAuth 2.0. Тобто будь-який користувач у якого є Google аккаунт зможе зручно та безпечно увійти в систему (рис. 3.1).

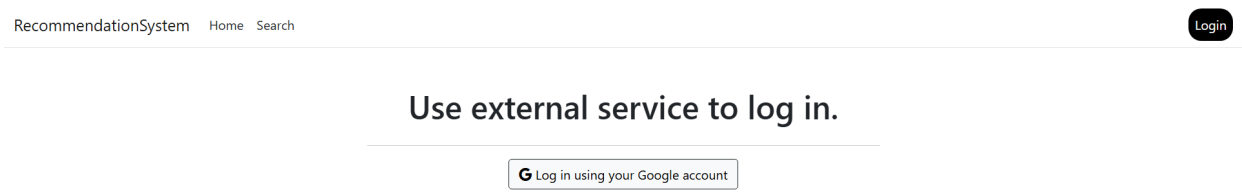

Рисунок 3.1 – Вхід за допомогою Google

Зберігання та пошук пісень здійснюється за допомогою бази даних MongoDB. Вибір MongoDB для цієї задачі обумовлений її високою продуктивністю при роботі з великими обсягами даних та гнучкістю у зберіганні складних структур даних, що ідеально підходить для швидкого пошуку музичного контенту (рис. 3.2 та рис. 3.3).

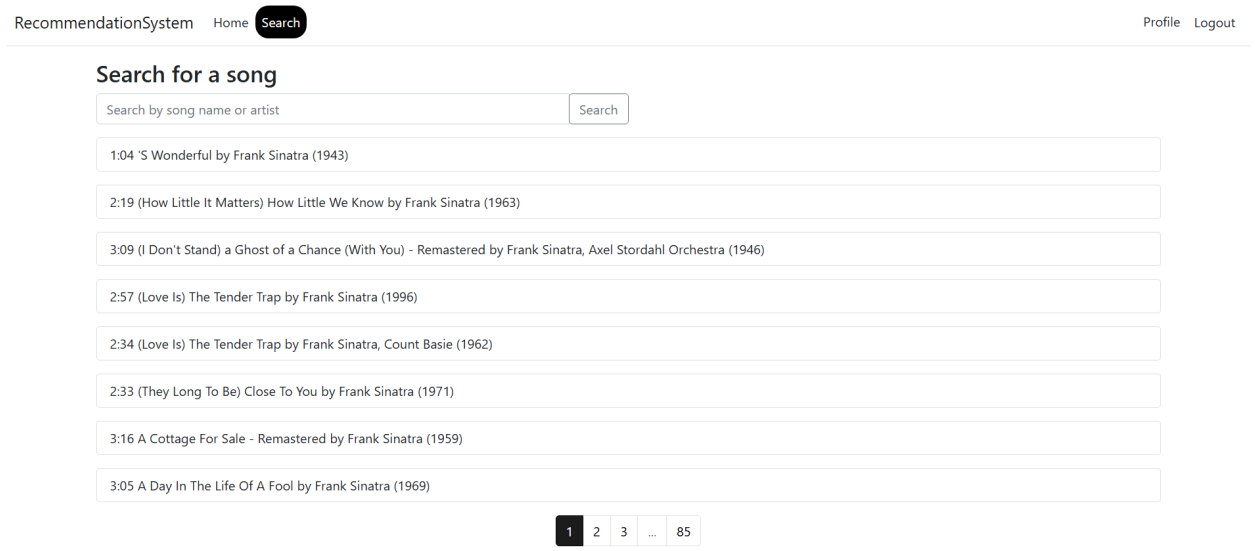

Рисунок 3.2 – Пошук пісень

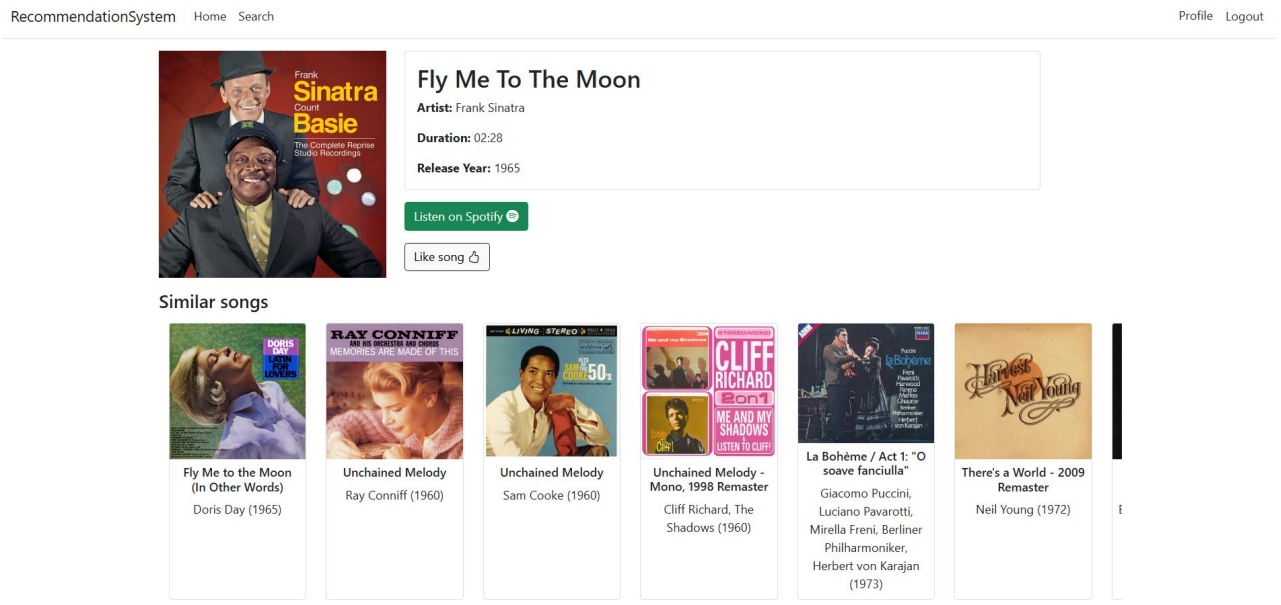

Рисунок 3.3 – Сторінка пісні

Система рекомендацій працює за двома основними принципами:

а) фільтрація на основі вмісту;

б) колаборативна фільтрація.

Ця комбінація методів дозволяє створювати персоналізовані та різноманітні рекомендації, враховуючи як індивідуальні уподобання користувача, так і загальні тренди серед схожих слухачів (рис. 3.4 та рис. 3.5).

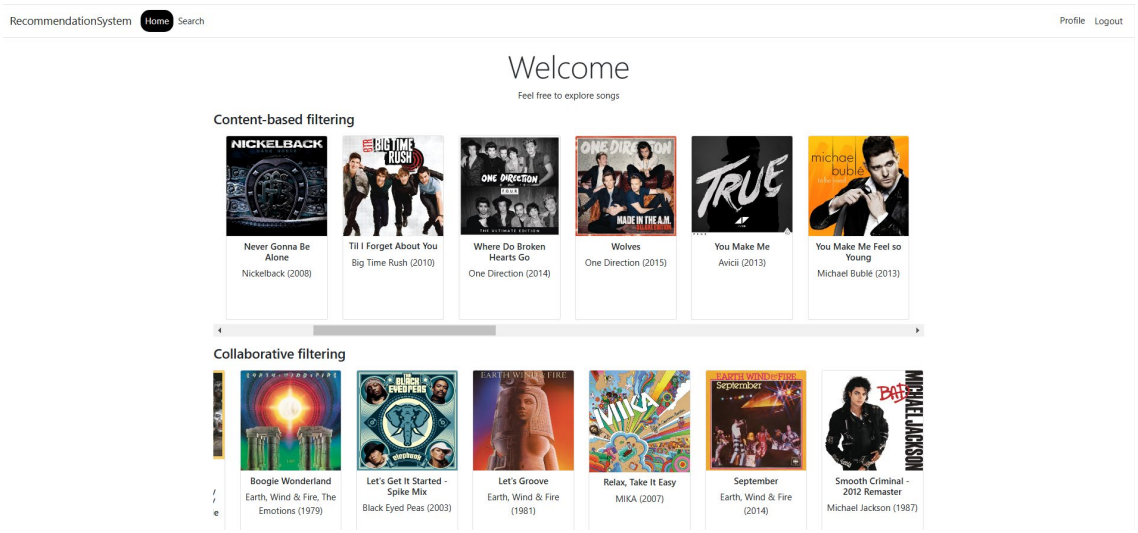

Рисунок 3.4 – Головна сторінка

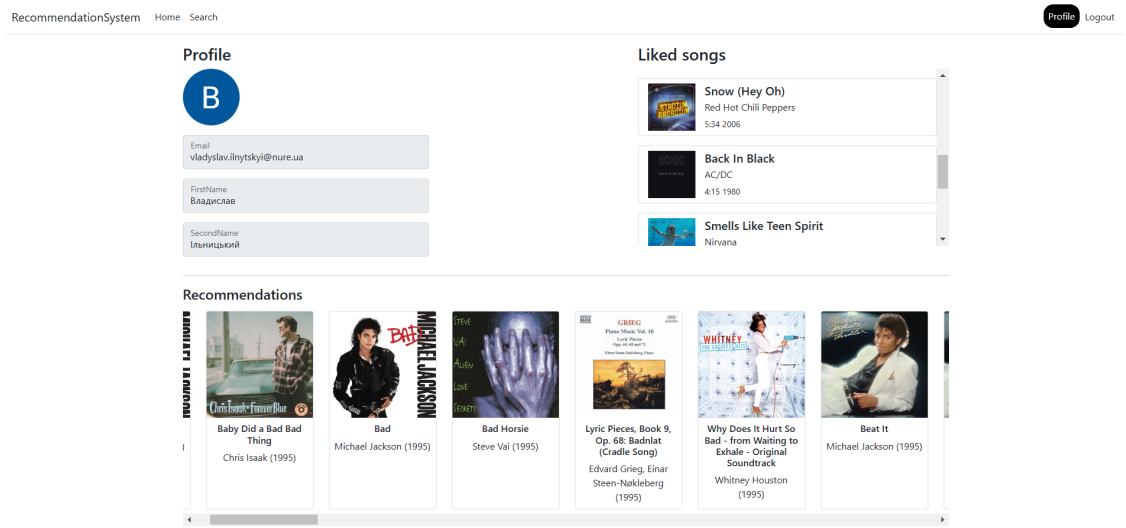

Рисунок 3.5 – Сторінка користувача

Така архітектура забезпечує високу гнучкість, масштабованість та ефективність системи, дозволяючи їй обробляти великі обсяги даних та надавати точні рекомендації в режимі реального часу.

Висновки за розділом 3

Python зарекомендував себе як потужний та гнучкий інструмент для проектів машинного навчання, включаючи складні рекомендаційні системи. Його простий синтаксис, багата екосистема бібліотек та активна спільнота розробників роблять його ідеальним вибором для розробки повномасштабних систем. Спеціалізовані бібліотеки, такі як Pandas, Scikit-learn, SciPy, NumPy, значно розширюють можливості Python у сфері обробки даних, реалізації алгоритмів машинного навчання та розгортання моделей.

Розроблена програма демонструє ефективне поєднання різних технологій для створення складної рекомендаційної системи. Багатошарова архітектура, інтеграція ASP.NET Core, Python та Razor Pages, реалізація складних алгоритмів рекомендацій та використання MongoDB для ефективного зберігання даних підкреслюють гнучкість та потужність обраного підходу.

Ця система є яскравим прикладом того, як сучасні технології та методи машинного навчання можуть бути застосовані для створення персоналізованих, масштабованих рекомендаційних систем, здатних обробляти великі обсяги даних та надавати точні рекомендації користувачам.

4 РЕЗУЛЬТАТИ ОБЧИСЛЮВАЛЬНОГО ЕКСПЕРИМЕНТУ ТА ЇХ АНАЛІЗ

4.1 Аналіз датасету

У цьому розділі ми зосередимося на детальному аналізі даних, які використовуються для розробки та оцінки нашої рекомендаційної системи. Аналіз даних є критично важливим етапом у створенні ефективної рекомендаційної системи, оскільки він дозволяє нам глибше зрозуміти структуру та характеристики наявної інформації, виявити приховані закономірності та тренди, а також підготувати дані для подальшої обробки алгоритмами машинного навчання.

Ми розглянемо різні аспекти наших даних, включаючи їх джерела, обсяг, якість та розподіл. Також дослідимо характеристики музичних треків, такі як жанри, популярність, та інші метадані, які можуть впливати на уподобання користувачів.

Представлена кореляційна матриця (рис. 4.1) відображає взаємозв’язки між різними характеристиками пісень. Аналіз цієї матриці дозволяє виявити кілька цікавих закономірностей.

Найсильніша позитивна кореляція спостерігається між енергією та гучністю пісень (0,74), що логічно, оскільки енергійні треки зазвичай звучать гучніше. Також помітна позитивна кореляція між танцювальністю та валентністю (0,47), що вказує на те, що більш танцювальні пісні часто мають більш позитивне емоційне забарвлення.

Цікаво відзначити сильну негативну кореляцію між акустичністю та енергією (–0,66), а також між акустичністю та гучністю (–0,55). Це свідчить про те, що акустичні треки зазвичай менш енергійні та тихіші.

Інструментальність має помірну негативну кореляцію з танцювальністю (–0,32) та гучністю (–0,55), що може вказувати на те, що інструментальні композиції часто менш танцювальні та тихіші.

Темп має досить слабкі кореляції з іншими характеристиками, найсиль-

ніша з яких – з енергією (0,16), що свідчить про відносну незалежність темпу від інших музичних параметрів.

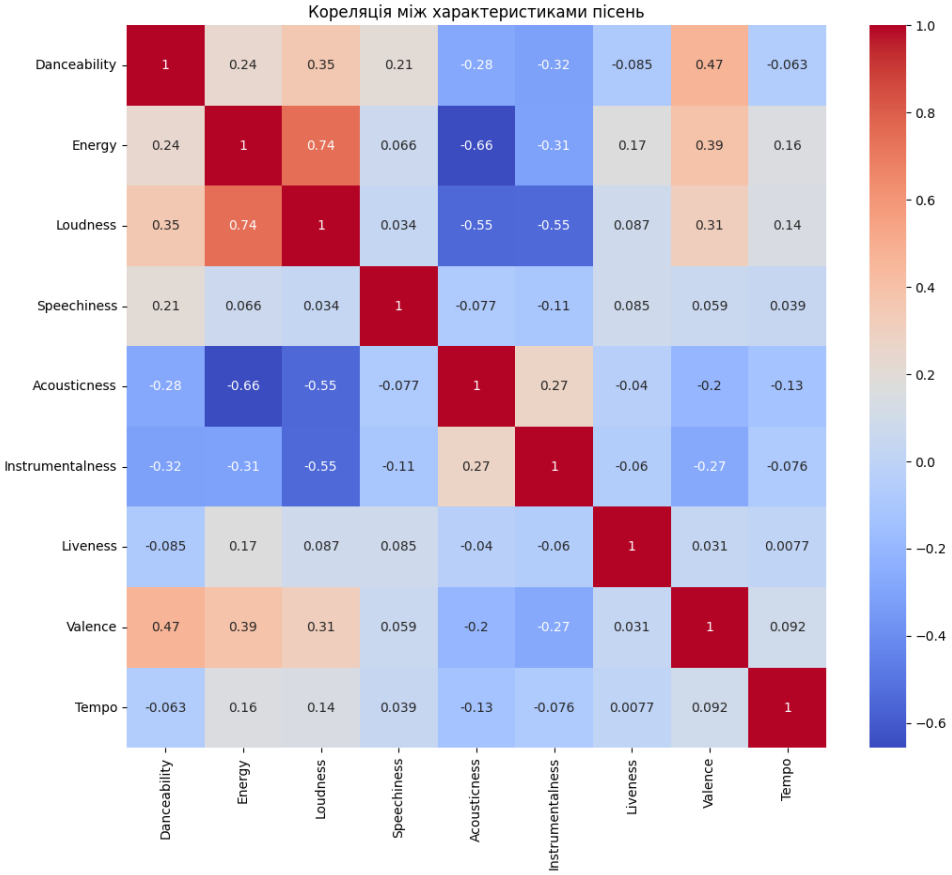

Рисунок 4.1 – Кореляційна матриця

Мовленнєвість (Speechiness) також демонструє слабкі кореляції з більшістю інших характеристик, найсильніша з яких – з танцювальністю (0,21).

Живість (Liveness) показує найслабші кореляції з іншими параметрами, що може вказувати на її незалежність від інших характеристик або на складність її точного визначення.

Загалом, ця кореляційна матриця надає цінну інформацію про взаємозв'язки між різними музичними характеристиками, що може бути корисним для розуміння структури музичних даних та розробки більш ефективних алгоритмів рекомендацій.

Наступний графік (рис. 4.2) представляє розподіл тривалості пісень у музичній колекції.

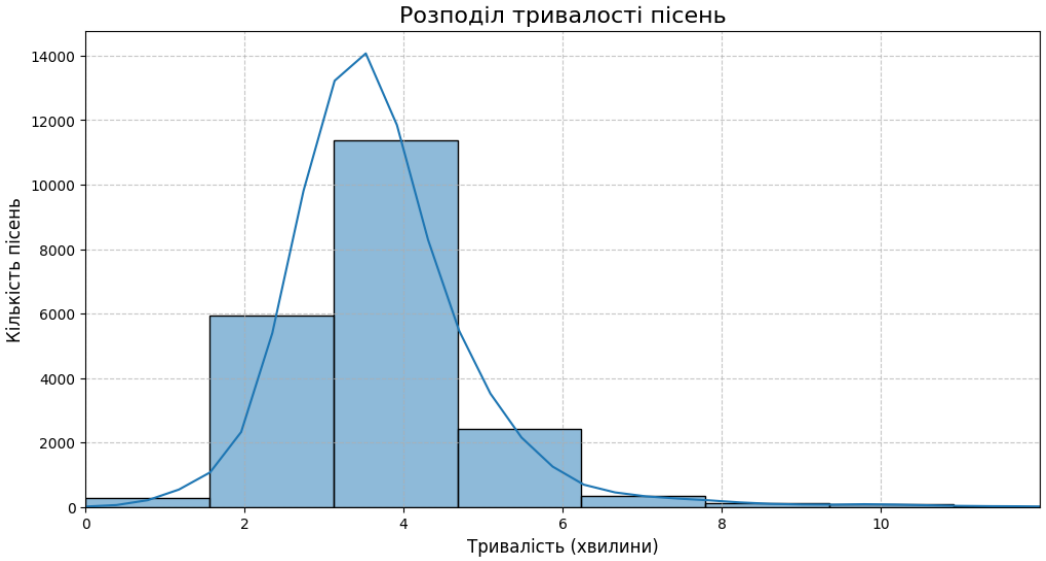

Рисунок 4.2 – Розподіл тривалості пісень

Аналізуючи цей графік, можна зробити кілька важливих спостережень:

– форма розподілу нагадує нормальний розподіл з невеликою правосторонньою асиметрією, що є типовим для тривалості пісень;

– більшість пісень у колекції мають тривалість від 2 до 6 хвилин, що відповідає стандартній довжині популярних музичних композицій;

– пік розподілу припадає на діапазон 3-4 хвилини, що вказує на те, що це найбільш поширена тривалість пісень у даній колекції;

– спостерігається різке зменшення кількості пісень тривалістю понад 6 хвилин, що свідчить про відносну рідкість довгих композицій.

Рисунок 4.3 – Залежність кількості прослуховувань від гучності

Рисунок 4.4 – Залежність кількості прослуховувань від акустичності

Рисунок 4.5 – Залежність кількості прослуховувань від енергійності

Рисунок 4.6 – Залежність кількості прослуховувань від танцювальності

Аналізуючи представлені вище графіки (рис 4.3 – 4.6), можна зробити наступні висновки:

– спостерігається значний пік прослуховувань для пісень з середньою акустичністю (близько 0,001), що може вказувати на популярність композицій, які балансують між акустичним та електронним звучанням;

– графік гучності показує кілька піків популярності на різних рівнях гучності, що свідчить про різноманітність уподобань слухачів, при чому найвищий пік спостерігається при значенні гучності близько -5 дБ;

– найбільша кількість прослуховувань припадає на пісні з низькою енергійністю (близько 0,07), після чого спостерігається різкий спад, що може вказувати на популярність спокійної, мелодійної музики;

– графік танцювальності демонструє загальну тенденцію до зростання кількості прослуховувань зі збільшенням танцювальності, з найвищим піком для найбільш танцювальних треків (близько 0,968).

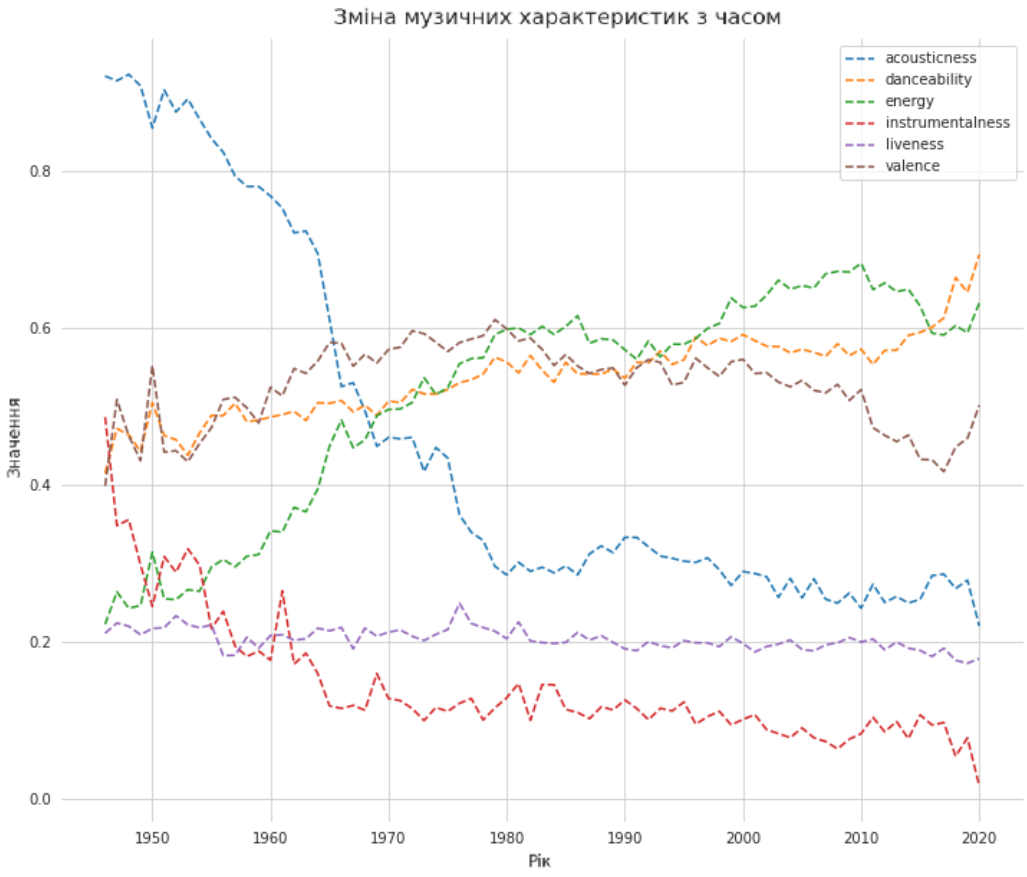

Рисунок 4.7 – Зміна музичних характеристик з часом

Графік (рис. 4.7) демонструє зміну різних музичних характеристик з 1950 по 2020 рік.

Акустичність показує значне зниження з часом, що відображає тенденцію до більш електронного звучання в музиці.

Танцювальність та енергійність, навпаки, демонструють поступове зростання, особливо помітне з 1980-х років, що вказує на збільшення популярності ритмічної та енергійної музики.

Інструментальність має тенденцію до зниження, що може свідчити про зростання ролі вокалу в сучасній музиці.

Живість (liveness) залишається відносно стабільною протягом усього періоду з невеликими коливаннями.

Валентність (емоційна позитивність) показує незначне зниження з часом, але залишається на середньому рівні.

Ці тренди відображають еволюцію музичних стилів та технологій виробництва музики, а також зміни в музичних уподобаннях слухачів протягом останніх 70 років.

4.2 Обчислювальний експеримент

4.2.1 Фільтрація на основі вмісту

Метод ліктя є важливим інструментом для визначення оптимальної кількості кластерів у задачах кластеризації. У контексті нашої рекомендаційної системи, цей метод може надати цінну інформацію для аналізу та оптимізації фільтрації на основі вмісту.

На рисунку 4.8 представлено залежність інерції від кількості кластерів k. Спостерігається різке зниження інерції при збільшенні k від 1 до 4, після чого крива починає вирівнюватися. Це вказує на те, що оптимальна кількість кластерів може бути в діапазоні 3-5, оскільки подальше збільшення k не призводить

до значного зменшення інерції.

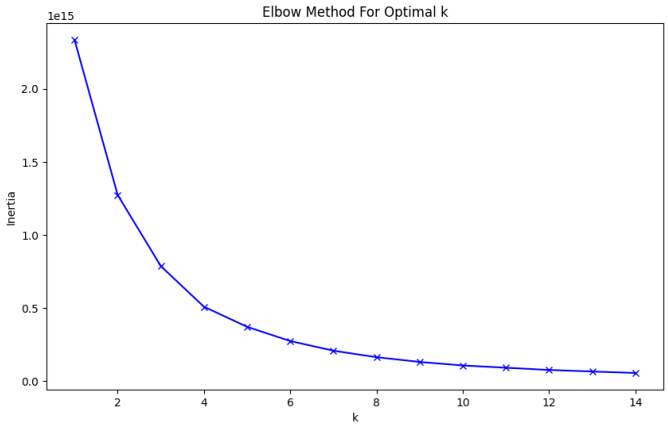

Рисунок 4.8 – Метод ліктя

Для уточнення оптимальної кількості кластерів було проведено силует-аналіз для 3, 4 та 5 кластерів:

Таблиця 4.1 – Залежність значення коефіцієнта від кількості кластерів

Кількість кластерів	Значення силует-коефіцієнта
3	0,55274
4	0,51947
5	0,50232

Найвище значення силует-коефіцієнта отримано для 3 кластерів, що вказує на найкращу якість кластеризації при цьому значенні. Однак, різниця між значеннями невелика, що свідчить про стабільність кластеризації при різних k.

На рисунку 4.9 представлено розподіл пісень у двовимірному просторі після застосування методу головних компонент (PCA). Чітко видно три основні кластери, позначені різними кольорами.

Ця візуалізація підтверджує результати силует-аналізу, демонструючи чітке розділення на 3 групи. Кожен кластер, представляє пісні зі схожими характеристиками або жанрами.

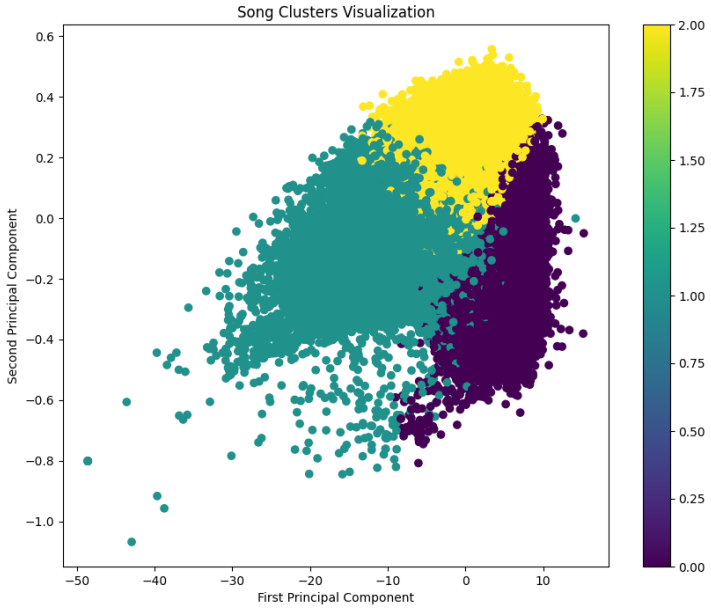

Рисунок 4.9 – Візуалізація кластерів

На рисунку 4.10 зображено рекомендації для композиції «Mozart / Arr Grieg: Piano Sonata No. 16 in C Major, K. 545: I. Allegro (Arr. Grieg for 2 Pianos)».

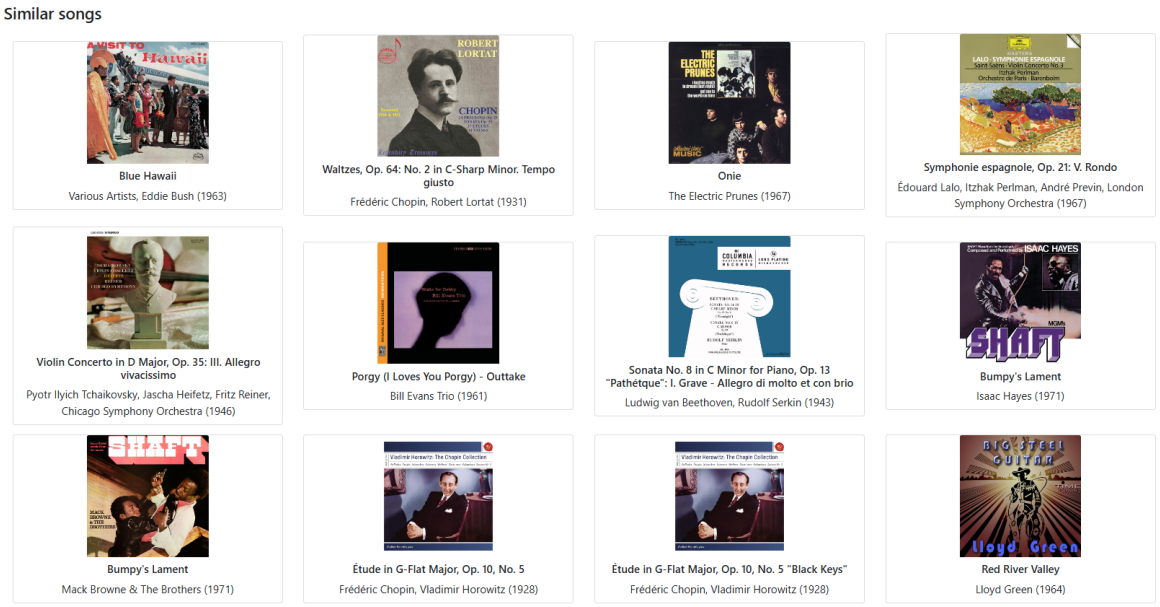

Рисунок 4.10 – Рекомендації на основі вмісту

На рисунку 4.11 зображено рекомендації для композиції «Sweet Victory David Glen Eisley, Bob Kulick»

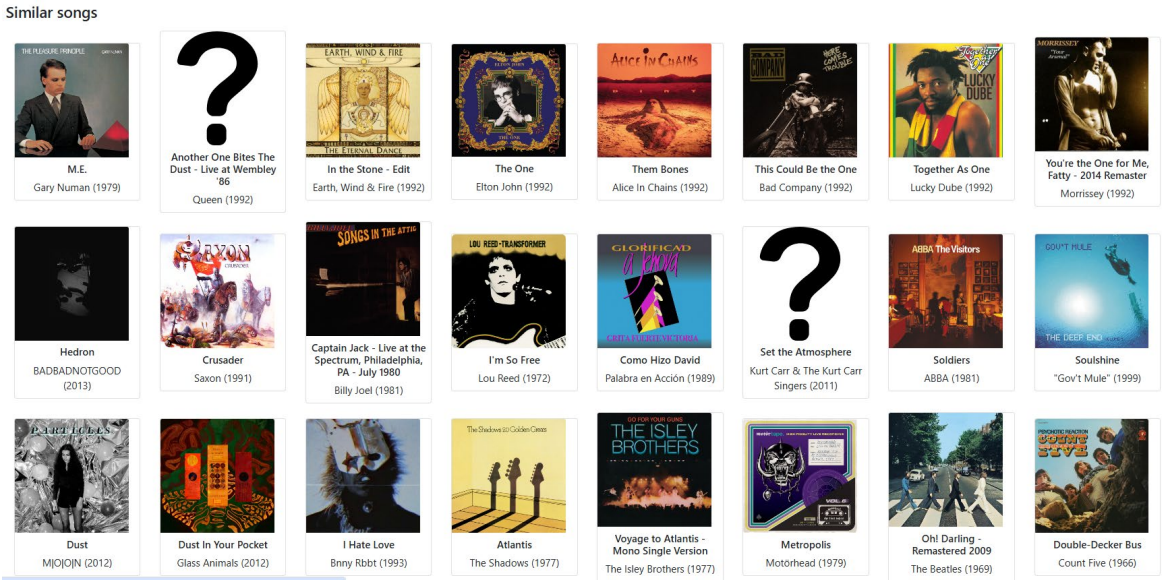

Рисунок 4.12 – Рекомендації на основі вмісту

4.2.2 Колаборативна фільтрація

Під час експерименту було створено 5 різних користувачів, які мали декілька схожих вподобань, а інша половина вподобань була різною.

Матриця схожості користувачів, представлена на рисунку 4.13, відображає ступінь подібності між 5 користувачами на основі їхніх вподобань.

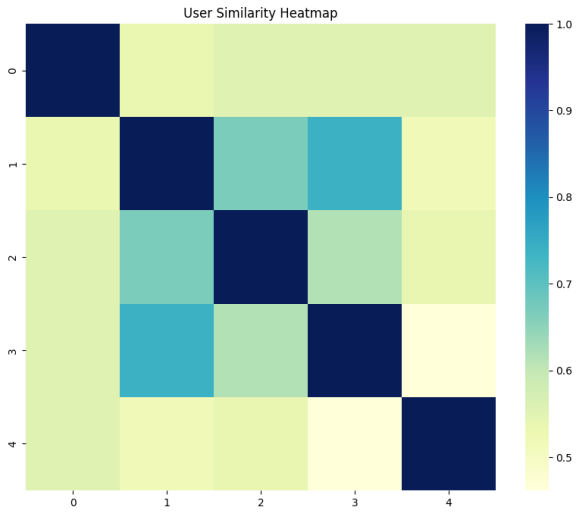

Рисунок 4.13 – Матриця схожості користувачів

З рисунку 4.13 можна побачити, що перший користувач має більш схожі вподобання з третім, четвертим і п’ятим користувачем.

Ця візуалізація дозволяє ефективно ідентифікувати групи користувачів зі схожими смаками, що є ключовим для побудови точних рекомендацій в системі колаборативної фільтрації.

Вподобання користувачів зображено на рисунку 4.14. Зеленими зірочками показано ті пісні, що будуть рекомендуватись для першого користувача.

На рисунку 4.15 зображено роботу колаборативної фільтрації для першого користувача.

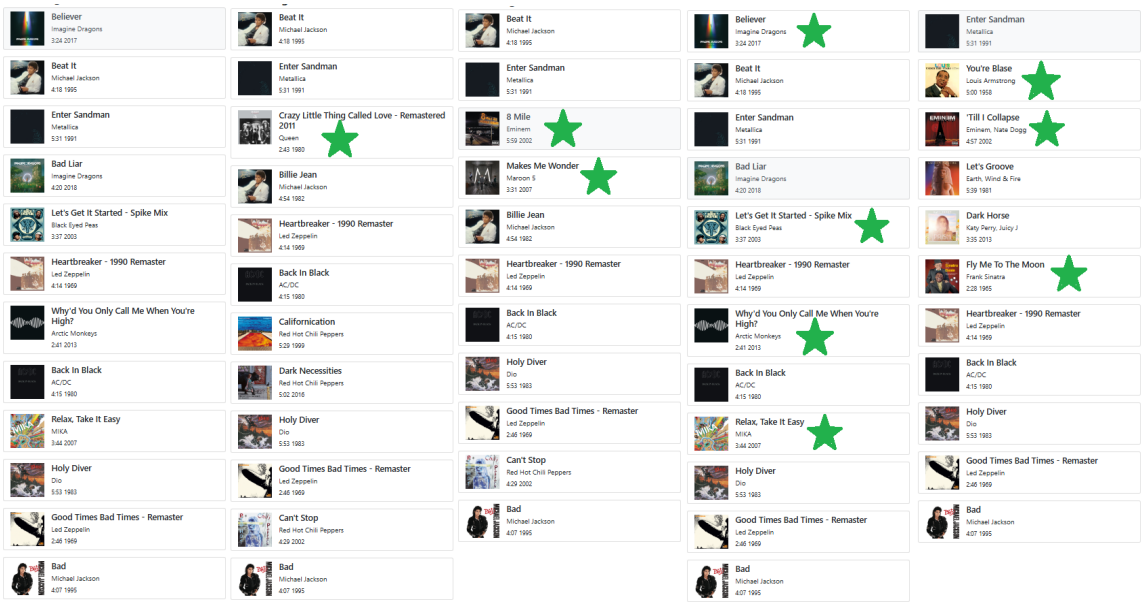

Рисунок 4.14 – Вподобання користувачів

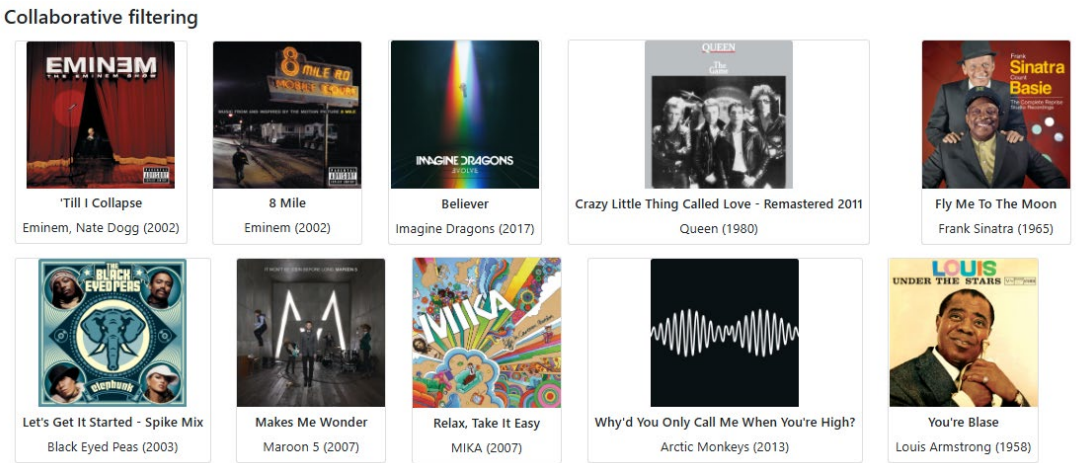

Рисунок 4.15 – Рекомендації для першого користувача

Висновки за розділом 4

Загалом, представлені результати демонструють ефективність розробленої рекомендаційної системи.

Фільтрація на основі вмісту за допомогою кластеризації контенту дозволяє групувати схожі елементи, а колаборативна фільтрація за допомогою аналізу схожості користувачів забезпечує персоналізовані рекомендації, які основуються на схожості вподобань користувачів.

Комбінація цих підходів створює потужний інструмент для надання релевантних рекомендацій користувачам.

ВИСНОВКИ

У рамках кваліфікаційної роботи було успішно розроблено та реалізовано рекомендаційну систему для аудіоконтенту, яка ефективно поєднує методи колаборативної фільтрації та фільтрації на основі вмісту. Дослідження охопило широкий спектр аспектів, від теоретичного обґрунтування до практичної реалізації та експериментальної оцінки.

Результати роботи демонструють, що розроблена рекомендаційна система здатна ефективно визначати вподобання користувачів та пропонувати релевантний аудіоконтент. Використання методів машинного навчання без учителя, зокрема кластеризації для фільтрації на основі вмісту та матричних обчислень для колаборативної фільтрації, дозволило створити гнучку та адаптивну систему.

Практична значимість роботи полягає в можливості застосування розробленої системи в різних сферах, де потрібні персоналізовані рекомендації аудіоконтенту, таких як музичні стрімінгові сервіси, подкаст-платформи чи освітні ресурси.

Подальші напрямки досліджень можуть включати вдосконалення алгоритмів рекомендацій, інтеграцію методів глибокого навчання для аналізу аудіоконтенту, а також розширення системи для роботи з мультимодальними даними.

Загалом, ця робота забезпечує міцну основу для подальшого розвитку та вдосконалення рекомендаційних систем у сфері аудіоконтенту, демонструючи потенціал інтеграції різних підходів машинного навчання для створення більш ефективних та персоналізованих рекомендаційних систем.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Ільницький В. Б. Розробка рекомендаційної системи для аудіоконтенту з застосуванням методів машинного навчання без учителя. *ІІІ Міжнародній науково-практичній конференції англійською мовою «Навчання і викладання: у світі після війни»* : зб. матеріалів форуму (м. Харків, 8 листопада 2024 р.).

2. Згуровський М. З., Панкратова Н. Д. Основи системного аналізу. Київ : Видавнича група BHV, 2007. 544 с.

3. Катренко А. В. Системний аналіз. Львів : «Новий світ – 2000», 2011. 396 с.

4. Recommender system. URL: https://en.wikipedia.org/wiki/Recommender_system (дата звернення: 04.09.2024).

5. Recommendation Systems: Content-Based Filtering. URL: https://medium.com/@zbeyza/recommendation-systems-content-based-filtering-e19e3b0a309e (дата звернення: 05.09.2024).

6. What Is Content-Based Filtering? Benefits and Examples in 2024. URL: https://www.upwork.com/resources/what-is-content-based-filtering (дата звернення: 05.09.2024).

7. Collaborative filtering. URL: https://developers.google.com/machine-learning/ recommendation/collaborative/basics (дата звернення: 06.09.2024).

8. What is collaborative filtering? URL: https://www.ibm.com/topics/ collaborative-filtering (дата звернення: 07.09.2024).

9. Kar P., Roy M., Datta S. Recommender Systems: Algorithms and Their Applications: eBook. Springer, 2024. 174 p.

10. Falk K. Practical Recommender Systems : eBook. Manning Publications Co., 2019. 432 p.

11. Басюк Т. М., Литвин В. В., Захарія Л. М., Кунанець Н. Е. Машинне навчання. Львів : Видавництво «Новий Світ - 2000», 2021. 315 с.

12. Item-item collaborative filtering. URL: https://en.wikipedia.org/wiki/Item-item_collaborative_filtering (дата звернення: 12.09.2024).