МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Чорноморський національний університет імені Петра Могили

Факультет комп'ютерних наук

Кафедра інтелектуальних інформаційних систем

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувач кафедри інтелектуальних інформаційних систем

_____ Євген СІДЕНКО

«____»_____ 2026 р.

КВАЛІФІКАЦІЙНА РОБОТА

НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ БАКАЛАВРА

ІНТЕЛЕКТУАЛЬНА СИСТЕМА РЕКОМЕНДАЦІЙ МУЗИЧНИХ КОМПОЗИЦІЙ НА ОСНОВІ АНАЛІЗУ ГАРМОНІЧНОЇ СТРУКТУРИ

Спеціальність 122 Комп'ютерні науки

Освітня програма «Комп'ютерні науки»

Здобувач _____Вікторія ВРАДІЙ

«___»_____ 2026 р.

Керівник канд. фіз.-мат. наук, доцент _____Інесса КУЛАКОВСЬКА

«___»_____ 2026 р.

Миколаїв – 2026

Чорноморський національний університет імені Петра Могили
(повне найменування закладу вищої освіти)

Факультет	Комп’ютерних наук
Кафедра	Інтелектуальних інформаційних систем
Рівень вищої освіти	Перший (бакалаврський)
Освітній ступень	Бакалавр
Спеціальність	122 Комп’ютерні науки
Освітня програма	Комп’ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри інтелектуальних інформаційних систем

_____ Євген СІДЕНКО

«____» _____ 2025 р.

ЗАВДАННЯ
на кваліфікаційну роботу здобувача

Врадій Віткорія Сергіївна
((прізвище, ім’я, по батькові здобувача)

1. Тема кваліфікаційної роботи: «Інтелектуальна система рекомендацій музичних композицій на основі аналізу гармонічної структури».

Керівник роботи: Кулаковська Інесса Василівна, канд. фіз.-мат. наук, доцент каф. ІІС.

Затверджена наказом ЧНУ ім. Петра Могили від «25» грудня 2025 р. № 353.

2. Строк представлення кваліфікаційної роботи «___» _____ 2025 р.

3. Очікуваний результат роботи та початкові дані, якщо такі потрібні: розроблена інтелектуальна система рекомендацій музичних композицій, яка здійснює аналіз гармонічної структури аудіосигналу, виконує семантичний аналіз тексту пісень та формує персоналізовані музичні рекомендації з використанням алгоритмів машинного навчання.

4. Перелік питань, що підлягають розробці: огляд та аналіз предметної сфери музичних рекомендаційних систем; постановка задачі; дослідження методів цифрової обробки аудіосигналів; аналіз та вибір технологій розробки системи; формування та підготовка датасету для навчання моделей; розробка модуля попередньої обробки аудіосигналів та екстракції ознак; реалізація ансамблю нейронних мереж ResNet та MLP; інтеграція інструментів NLP та моделей Speech-to-Text; реалізація алгоритму мультисигнального ранжування рекомендацій; проєктування та наповнення бази даних; програмна реалізація клієнтської та серверної частини системи; тестування функціоналу системи та аналіз отриманих результатів; визначення напрямів подальшого розвитку системи .

Керівник роботи _____ Інесса КУЛАКОВСЬКА
(Особистий підпис) *(Власне ім'я ПРІЗВИЩЕ)*

Здобувач _____ Вікторія ВРАДІЙ
(Особистий підпис) *(Власне ім'я ПРІЗВИЩЕ)*

Дата видачі завдання «23» грудня 2025 р.

КАЛЕНДАРНИЙ ПЛАН
кваліфікаційної роботи

Тема: Інтелектуальна система рекомендацій музичних композицій на основі аналізу гармонічної структури

№	Найменування роботи	Початок	Закінчення	Примітки
1	Отримання завдання на виконання КР	21.12.2025	24.12.2025	
2	Аналіз предметної області та постановка задачі	25.12.2025	30.01.2026	
3	Огляд літературних джерел за темою кваліфікаційної роботи, зокрема огляду сучасних рекомендаційних систем, методів аналізу звукових сигналів та обробки тексту пісень	31.01.2026	01.03.2026	
4	Огляд існуючих моделей глибокого навчання та методів NLP для аналізу аудіо й семантики текстів	02.03.2026	01.04.2026	
5	Програмна реалізація інтелектуальної системи рекомендацій та аналіз отриманих результатів	02.04.2026	24.05.2026	
6	Перший попередній захист КР на засіданні комісії кафедри	25.05.2026	25.05.2026	
7	Корегування роботи за результатами попереднього захисту	26.05.2026	04.06.2026	
8	Другий попередній захист КР на засіданні комісії кафедри	05.06.2026	05.06.2026	
9	Доробка та остаточне оформлення КР	06.06.2026	14.06.2026	
10	Подання КР, її електронної копії та інших документів (відгуку, рецензії) до захисту	15.06.2026	19.06.2026	

Керівник роботи _____ Інесса КУЛАКОВСЬКА
(Особистий підпис) *(Власне ім'я ПРІЗВИЩЕ)*

Здобувач _____ Вікторія ВРАДІЙ
(Особистий підпис) *(Власне ім'я ПРІЗВИЩЕ)*

Дата складання календарного плану
«29» січня 2026 р.

АНОТАЦІЯ

до кваліфікаційної роботи
здобувачки групи 401 ЧНУ ім. Петра Могили

Врадій Вікторії Сергіївни

на тему: **“ІНТЕЛЕКТУАЛЬНА СИСТЕМА РЕКОМЕНДАЦІЙ МУЗИЧНИХ КОМПОЗИЦІЙ НА ОСНОВІ АНАЛІЗУ ГАРМОНІЧНОЇ СТРУКТУРИ”**

Актуальність даної роботи полягає у необхідності розробки музичної рекомендаційної системи, яка працює без поведінкових даних користувачів, вирішує проблему «холодного старту» та забезпечує точність рекомендацій на основі об'єктивних акустичних і семантичних ознак.

Об'єктом роботи є процес автоматичного формування персоналізованих музичних рекомендацій.

Предметом роботи є методи контентного аналізу звукових сигналів та обробки природної мови в задачах музичної рекомендації.

Метою роботи є розробка інтелектуальної системи рекомендацій музичних композицій на основі контентно-гармонічного аналізу, що поєднує акустичні характеристики аудіосигналу та семантичний аналіз тексту пісень.

В результаті виконання роботи було розроблено та навчено ансамбль моделей глибокого навчання для класифікації акустичних ознак, інтегровано моделі Whisper та DistilBERT для аналізу семантики текстів пісень, а також реалізовано алгоритм мультисигнального ранжування.

Дана робота складається з вступу, чотирьох розділів, висновків та додатків. У розділах проведено огляд аналогів, досліджено математичні моделі та описано програмну реалізацію з результатами тестування. Загальний обсяг роботи — 85 сторінок. Кваліфікаційна робота містить 5 додатків, 25 рисунків, 6 таблиць і 33 джерела посилання.

Ключові слова: музичні рекомендаційні системи, контентно-гармонічний аналіз, мел-спектрограма, глибоке навчання, обробка природної мови, мультисигнальне ранжування.

ABSTRACT

to the qualification work by the student of the group 401 of Petro Mohyla Black Sea National University

Vradii Viktoriia

“INTELLIGENT MUSIC RECOMMENDATION SYSTEM BASED ON HARMONIC STRUCTURE ANALYSIS”

A relevance of this work lies in the need to develop a music recommendation system that operates without user behavioral data, resolves the "cold start" problem, and provides high-accuracy recommendations based on objective acoustic and semantic characteristics of songs.

An object of the work is the process of automatic generation of personalized music recommendations.

A subject of the work is the methods of audio signal content analysis and natural language processing in music recommendation tasks.

A purpose of the work is to develop an intelligent music recommendation system based on content-harmonic analysis that combines acoustic characteristics of the audio signal and semantic analysis of song lyrics.

As a result of the work, an ensemble of deep learning models was developed and trained to classify acoustic features, Whisper and DistilBERT models were integrated for lyrics semantics analysis, and a multi-signal ranking algorithm was implemented.

This work consists of an introduction, four sections, conclusions, and applications. The sections review existing analogs, study mathematical models and methods, and describe the software implementation with testing results. The overall scope of the work is 85 pages. Thesis contains 5 applications, 25 figures, 6 tables and 33 references in it.

Key words: music recommendation systems, content-harmonic analysis, mel-spectrogram, deep learning, natural language processing, multi-signal ranking.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

AI – Artificial Intelligence

API – Application Programming Interface

BPM – Beats Per Minute

CF – Collaborative Filtering

CNN – Convolutional Neural Network

MFCC – Mel-Frequency Cepstral Coefficients

MIR – Music Information Retrieval

MLP – Multilayer Perceptron

NLP – Natural Language Processing

ORM – Object-Relational Mapping

ResNet – Residual Network

RMS – Root Mean Square

STFT – Short-Time Fourier Transform

ВСТУП

Стрімінгові музичні платформи стали домінуючою формою споживання музики, а щодня на них завантажуються сотні тисяч нових композицій. В умовах такого інформаційного перенавантаження якість системи рекомендацій перетворилась на ключовий фактор конкурентоспроможності будь-якого музичного сервісу та головний інструмент персоналізації музичного досвіду слухача.

Домінуючим підходом у галузі залишається колаборативна фільтрація — метод, що аналізує поведінкові дані користувачів та виявляє схожість між ними. Попри широке поширення, цей підхід має суттєві системні обмеження. Він страждає від упередженості до популярності: система просуває треки з найбільшою кількістю прослуховувань, залишаючи поза увагою маловідомих виконавців. Крім того, колаборативна фільтрація не здатна рекомендувати нові треки, для яких ще не накопичено статистику взаємодій, — так звана проблема холодного старту. Нарешті, непрозорість таких алгоритмів унеможливлює пояснення рекомендацій користувачеві.

Актуальність роботи зумовлена необхідністю розробки рекомендаційної системи, яка долає зазначені обмеження: функціонує без поведінкових даних, забезпечує рекомендації для нових треків незалежно від їхньої популярності та дає змогу пояснити кожну рекомендацію на основі об'єктивних музичних характеристик.

Метою роботи є розробка інтелектуальної системи рекомендацій музичних композицій на основі контентно-гармонічного аналізу, що поєднує аналіз акустичних характеристик аудіосигналу та семантичний аналіз тексту пісень.

Для досягнення поставленої мети необхідно вирішити такі завдання:

– проаналізувати існуючі методи побудови музичних рекомендаційних систем та виявити їхні ключові обмеження;

– дослідити методи цифрової обробки аудіосигналів як основи для побудови векторних представлень музичних композицій;

– розробити та навчити модель глибокого навчання для класифікації акустичних ознак треків;

– реалізувати модуль семантичного аналізу тексту пісень для визначення їхнього емоційного забарвлення;

– розробити алгоритм формування рекомендацій на основі багатовимірної схожості акустичних і семантичних характеристик;

– реалізувати програмну систему з веб-інтерфейсом та сформувати базу даних музичних композицій.

Об'єктом роботи є процес автоматичного формування персоналізованих музичних рекомендацій.

Предметом роботи є методи контентного аналізу аудіосигналів та природної мови в задачах музичних рекомендацій.

Практичне значення роботи полягає у створенні програмної системи, яка здатна рекомендувати музичні композиції виключно на основі їхнього звучання та текстового змісту, без потреби у накопиченій статистиці прослуховувань. Це відкриває рівні можливості для просування як відомих, так і маловідомих виконавців та забезпечує прозорість і обґрунтованість прийнятих рекомендаційних рішень.

1 АНАЛІЗ ПРОБЛЕМАТИКИ МУЗИЧНИХ РЕКОМЕНДАЦІЙНИХ СИСТЕМ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Опис предметної сфери

Сучасна індустрія цифрової музики переживає етап стрімкого зростання завдяки розвитку стрімінгових платформ. Щодня створюються і завантажуються на сервери мільйони нових музичних композицій, що створює проблему інформаційного перенавантаження для кінцевого користувача. У цих умовах ключовим фактором конкурентоспроможності будь-якого музичного сервісу стає наявність ефективної системи рекомендацій, здатної автоматично формувати персоналізовані плейлисти та пропонувати слухачеві релевантний контент.

Традиційно предметна сфера музичних рекомендаційних систем базується на методах аналізу великих даних. Історично першими та найбільш поширеними стали алгоритми колаборативної фільтрації. Їхня суть полягає в аналізі історії прослуховувань: якщо користувачі мають схожі музичні вподобання, композиції, що сподобались одним користувачам, можуть бути рекомендовані іншим.

Проте було виявлено суттєві недоліки такого підходу. У дослідженні [1] описано проблему популярнісного зміщення (popularity bias), пов’язану з феноменом «довгого хвоста» (The Long Tail) у музичних рекомендаційних системах. Автори показують, що сучасні алгоритми рекомендацій частіше пропонують популярний контент, тоді як менш популярні музичні композиції недостатньо представлені в рекомендаціях (див. рис. 1.1). Це може призводити до того, що користувачі з нішевими музичними вподобаннями отримують менш релевантні рекомендації.

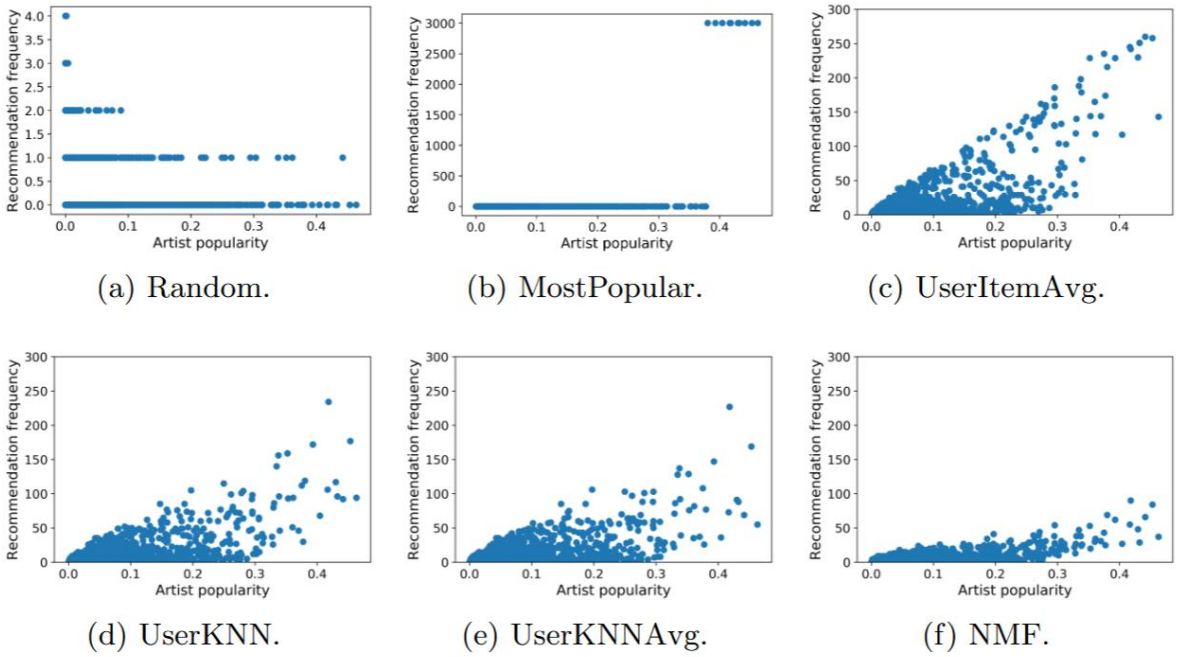

Рисунок 1.1 – Залежність між популярністю виконавця та частотою рекомендацій на прикладі 6 алгоритмів [1]

Іншою важливою проблемою музичних рекомендаційних систем є явище «холодного старту» (Cold Start). За відсутності історії взаємодій для нових об'єктів схеми колаборативної фільтрації не можуть використовувати колаборативні сигнали для визначення вподобань користувачів [2]. Тому системи, що використовують традиційні рекомендаційні методи мають труднощі з рекомендацією нових музичних композицій, для яких ще не накопичено достатньої інформації про взаємодію користувачів [3]. Для вирішення цієї проблеми дослідники запропонували використовувати контентний аналіз музики, зокрема застосування згорткових нейронних мереж (CNN) для аналізу аудіосигналів безпосередньо у рекомендаційних системах [3]. Такий підхід став одним із важливих напрямів розвитку галузі автоматичного вилучення музичної інформації (MIR).

Подальший розвиток методів машинного навчання та обробки природної мови сприяв переходу до мультимодальних рекомендаційних систем. Сучасні мультимодальні підходи поєднують аналіз гармонії, акустичних характеристик

музики, текстової інформації, метаданих та інших джерел даних, формуючи більш інформативні представлення контенту поряд із поведінковими даними користувачів [4,5]. Такі підходи демонструють покращення якості рекомендацій та ефективності роботи у сценаріях холодного старту, зокрема для нових або малопопулярних об'єктів [5].

1.2 Класифікація методів побудови рекомендаційних систем

Перед аналізом конкретних комерційних рішень необхідно розглянути базові підходи, на яких базується сучасна наука про рекомендаційні системи. Згідно з класифікацією, наведеною у [6] (див. рис. 1.2), до базових категорій рекомендаційних систем належать системи колаборативної фільтрації, контент-базовані системи та гібридні підходи.

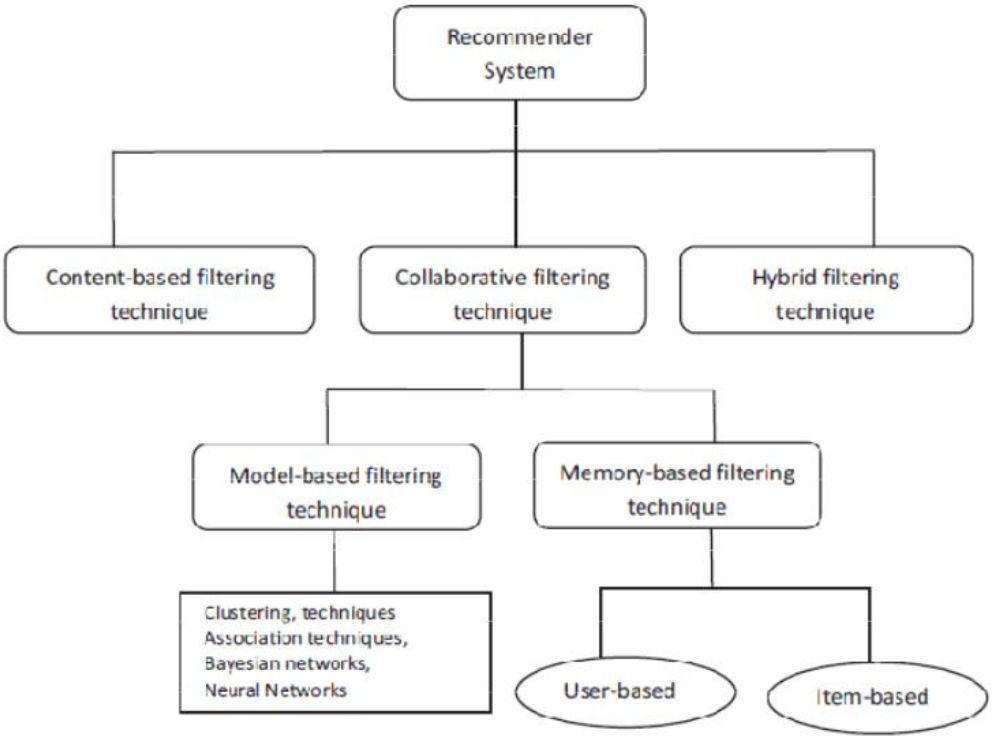

Рисунок 1.2 – Класифікація підходів до створення рекомендаційних систем [6]

Колаборативна фільтрація є одним із базових підходів у системах рекомендацій, який ґрунтується на аналізі історії взаємодій користувачів з

об’єктами, зокрема оцінок, переглядів або іншого поведінкового фідбеку [6]. Сучасні оглядові роботи також визначають колаборативну фільтрацію як метод, що використовує лише user-item взаємодії для прогнозування інтересів користувача без використання інформації про контент об’єктів [7].

Основна гіпотеза методу полягає в тому, що користувачі, які демонстрували схожі вподобання в минулому, з великою ймовірністю збережуть подібність і в майбутньому, що лежить в основі neighborhood-based моделей рекомендацій, де використовується принцип знаходження схожих користувачів або схожих об'єктів. [8].

У межах memory-based колаборативної фільтрації виділяють два основні підходи: user-based, де знаходяться користувачі з подібними профілями поведінки, та item-based, де обчислюється схожість між об’єктами на основі спільних взаємодій користувачів [6, 8]. Сучасні дослідження підтверджують, що item-based підхід часто є більш стабільним у масштабних системах через меншу змінність характеристик об’єктів у порівнянні з користувацькими профілями [8].

Водночас метод має суттєві обмеження, пов’язані з розрідженістю матриці взаємодій та проблемою холодного старту, коли відсутність достатньої кількості історичних даних не дозволяє формувати якісні рекомендації [6, 7]. Додатковим недоліком є те, що підхід не враховує семантичні характеристики самих об’єктів, оскільки базується виключно на поведінкових даних користувачів, що особливо обмежує його ефективність у випадках нових або слабо оцінених об’єктів [7].

Контент-базовані системи, на відміну від колаборативних, фокусуються на аналізі внутрішніх властивостей самих об’єктів. У межах такого підходу алгоритм формує інформаційний профіль користувача на основі історії його взаємодій і здійснює порівняння математично визначених атрибутів нових об’єктів із накопиченими даними про вподобання для безпосередньої генерації рекомендацій [6]. Оцінювання подібності дозволяє пропонувати позиції, що мають найбільшу відповідність раніше обраним об'єктам, без залучення інформації про поведінку інших суб'єктів системи.

Перевагою цього підходу є здатність функціонувати за відсутності великої кількості глобальних зв'язків, що частково нівелює проблему розрідженості матриць взаємодій. Проте ефективність методу безпосередньо залежить від того, наскільки точно описано характеристики об'єктів та правильно сформовано профіль користувача [6].

Гібридні системи рекомендацій поєднують переваги колаборативного та контент-базованого підходів і дозволяють компенсувати їхні окремі недоліки [6]. Основна ідея таких методів полягає в інтеграції різних стратегій рекомендацій для підвищення якості прогнозування та стабільності результатів.

У гібридних системах виділяють кілька основних підходів до поєднання методів: незалежне застосування колаборативної та контентної фільтрації з подальшим об’єднанням результатів, використання елементів одного методу в іншому, а також побудову єдиної моделі, яка одночасно враховує обидва типи інформації. Така комбінація дозволяє зменшити вплив окремих недоліків кожного з підходів, зокрема проблеми розрідженості даних та обмежень контентного представлення об’єктів.

Гібридний підхід є одним із найбільш досліджуваних напрямів у сучасних системах рекомендацій, оскільки він забезпечує більшу гнучкість у виборі алгоритмів та дозволяє адаптувати модель до різних типів даних та прикладних задач. Проте сучасні дослідження у сфері музичних рекомендацій показують, що застосування методів глибокого навчання для контентного аналізу музичних даних дозволяє ефективно працювати з мультимодальною інформацією та частково вирішувати проблему холодного старту, що є однією з ключових обмежень класичних підходів [9].

1.3 Огляд та аналіз наявних аналогів

Для обґрунтування напрямку розробки було проведено огляд існуючих на ринку аналогів інтелектуальних рекомендаційних систем, серед яких лідерами є Spotify, Apple Music та YouTube Music.

Система рекомендацій Spotify базується на гібридному підході, що поєднує методи колаборативної фільтрації, моделей представлення та аналізу контентних ознак аудіо й метаданих. Основним джерелом сигналів є поведінкові дані користувачів, включаючи історію прослуховувань, взаємодії з треками та статистику пропусків, що дозволяє будувати моделі неявного зворотного зв'язку [10]. Додатково застосовуються методи представлення музичних об'єктів у векторному просторі, де треки, плейлисти та користувачі відображаються у вигляді числових векторів для подальшого пошуку схожих елементів. [11]. Для обробки контентних характеристик використовуються як метадані (жанри, виконавці, текстові атрибути плейлистів), так і аудіоознаки, отримані шляхом аналізу спектральних характеристик сигналу, включаючи глибокі нейронні мережі для витягання ознак [12]. Для масштабування обчислень у реальному часі застосовуються алгоритми наближеного пошуку найближчих сусідів (Approximate Nearest Neighbors), що дозволяє ефективно працювати з великими просторами ознак [13].

YouTube Music використовує двоетапну архітектуру рекомендацій, що базується на двовежевих нейронних мережах (Two-Tower Neural Networks, TTNN) [14]. На першому етапі генерації кандидатів система з мільярдів відеозаписів відбирає кілька сотень релевантних треків за допомогою колаборативної фільтрації, спираючись на історію прослуховувань та уподобання схожих користувачів. На другому етапі ранжування відібрані кандидати оцінюються за детальнішим набором ознак, включаючи контекстуальні сигнали [15]. Двовежева архітектура відображає запити користувачів та контент у спільний векторний простір, де семантично близькі об'єкти розміщуються ближче одне до одного, що дозволяє ефективно шукати найрелевантніших кандидатів. Головним недоліком такого підходу є виражена схильність до упередженості популярності (popularity bias): система максимізує релевантність на основі статистичної частоти взаємодій, що призводить до надмірного просування популярних треків та недостатнього охоплення маловідомих виконавців [1, 15].

Apple Music реалізує гібридну стратегію, яку самі розробники платформи визначають як «алготоріальну» (від англ. algotorial — поєднання algorithmic та editorial), що базується на синергії ручного редакторського курування та алгоритмів машинного навчання [16]. Платформа залучає понад тисячу музичних редакторів по всьому світу, які вручну формують тематичні плейлисти; ці редакційні рішення, у свою чергу, стають вхідними сигналами для автоматизованих рекомендаційних моделей [17]. З алгоритмічного боку система поєднує колаборативну фільтрацію, контентну фільтрацію на основі метаданих (жанр, темп, настрій, інструментарій) та контекстуальні сигнали (час доби, тип пристрою, нещодавня активність). Через центральну роль редакторів у формуванні рекомендаційного простору система є менш гнучкою до автоматичного масштабування та персоналізації у порівнянні з повністю алгоритмічними підходами [17].

У сучасних наукових публікаціях акцентується увага на визначенні акустичних характеристик треків безпосередньо з аудіосигналу [20]. Водночас більшість досліджень зосереджена переважно на акустичних ознаках і не враховує семантичний аналіз тексту пісні як окремий інформаційний сигнал [21].

Паралельним напрямом досліджень є аналіз емоційного забарвлення музичних творів на основі тексту лірики із застосуванням методів NLP. Дослідження показують, що NLP-методи здатні виявляти тематичні кластери та емоційні патерни в текстах пісень, розкриваючи зв'язок між мовними структурами, емоційним тоном та художнім вираженням [21]. Однак об'єднання акустичного та текстового аналізу в єдиній рекомендаційній системі залишається невирішеним завданням у більшості існуючих підходів [4, 5].

Окремою актуальною проблемою є прозорість рекомендаційних систем. Сучасні дослідження фіксують зростаючий інтерес до пояснюваності алгоритмів рекомендацій: пояснення підвищують довіру користувачів і дозволяють їм розуміти, чому система пропонує той чи інший контент [22]. Вимоги до прозорості алгоритмів починають закріплюватися навіть на законодавчому рівні — зокрема, у

рамках Акту ЄС про цифрові послуги [22]. Проте на практиці більшість комерційних платформ досі не надають користувачу жодного пояснення логіки рекомендацій, і користувач не має можливості зрозуміти, на якій підставі йому запропоновано той чи інший трек. Така непрозорість знижує довіру до системи та позбавляє користувача контролю над власним музичним досвідом.

Для систематизації результатів огляду та обґрунтування унікальної архітектури розроблюваної системи було визначено перелік порівняльних критеріїв, що базуються на актуальних проблемах галузі:

- залежність від популярності — дослідження показують, що колаборативна фільтрація схильна до ігнорування маловідомих композицій на користь популярним [1, 13]; ідеальна система повинна рекомендувати музику на основі її об'єктивного звучання;
- проблема холодного старту — здатність системи рекомендувати абсолютно новий трек без накопиченої статистики взаємодій;
- аналіз гармонії — наявність математичного аналізу акордів та тональності прямо з аудіосигналу для забезпечення плавних переходів [20];
- семантичний аналіз лірики — здатність системи транскрибувати текст пісні та розуміти її емоційне забарвлення [21];
- прозорість прийняття рішень — можливість чітко пояснити математичну причину рекомендації [22].

Таблиця 1.1 – Порівняльний аналіз існуючих музичних рекомендаційних систем та розроблюваного продукту

Критерій	**Spotify**	**YouTube Music**	**Apple Music**	**Розроблювана система**
Основний підхід	Гібридний (CF + NLP + CNN) [10–12]	Гібридний (TTNN) [14, 15]	«Алготоріальний» (CF + CBF + курація) [16, 17]	Мультисигнальний контентно-гармонічний (8 зважених сигналів)

Продовження таблиці 1.1

Критерій	**Spotify**	**YouTube Music**	**Apple Music**	**Розроблювана система**
Стійкість до холодного старту	Середня, CF потребує історії прослуховувань	Низька, нові треки без історії не потрапляють у кандидати	Середня, частково вирішується куратією	Повна, аналізується лише контент треку без потреби в історії
Упередженість до популярності	Присутня, CF просуває популярні треки	Виражена, архітектура максимізує статистичну релевантність	Помірна, компенсується редакційною куратією	Відсутня, оцінка базується виключно на акустичних та семантичних ознаках
Аналіз гармонії	Непрямо (спектральні ознаки) [12]	Ні	Ні (закрита архітектура)	Так, ядро системи: Chroma CQT, коло квінт, Tonnetz
Аналіз мел-спектрограм	Так, CNN [10, 12]	Так, спектральний аналіз [14]	Закрита архітектура [16]	Так, ResNet + MLP
Семантичний аналіз тексту	Так, NLP + Word2Vec [12]	Так, аудіо-текстова модель [19]	Ні, лише метадані [16, 17]	Так, Whisper, Google Translate, DistilBERT Zero-Shot
Прозорість методу	Низька	Низька	Низька	Висока, формула зваженої суми 8 сигналів з інтерпретованими вагами
Прозорість прийняття рішень	Відсутня, користувач не бачить причин рекомендації	Відсутня, закрита двовежева модель	Відсутня, закрита «алготоріальна» модель	Повна, користувач бачить жанр, емоційний настрій, тональність, темп та відсоток подібності кожного треку
Масштабування	Так, Annoy Trees [10]	Так, ANN [14]	Так [16]	NumPy In-Memory Index, оптимально для каталогу ≤ 100K треків

Кінець таблиці 1.1

Критерій	Spotify	YouTube Music	Apple Music	Розроблювана система
Підтримка нових виконавців	Обмежена [13]	Обмежена [15]	Часткова [17]	Повна, автоматичний аналіз кожного аудіо

Як видно у табл. 1.1, комерційні системи ефективні для масового споживача, але породжують проблему холодного старту для нових артистів. YouTube Music демонструє найвищий рівень популяризаційного упередження, Apple Music обмежена ручною куратією, а Spotify, хоч і аналізує звук, все одно надає перевагу колаборативній складовій.

Натомість розроблювана система повністю відмовляється від метрик популярності та історії прослуховувань інших користувачів. Векторні представлення формуються виключно з акустичних характеристик звуку та емоційного забарвлення лірики. Це дозволяє знаходити ідеальні рекомендації на основі чистої математичної подібності навіть для композицій із нульовою статистикою, що вирізняє розробку на тлі існуючих аналогів та підтверджує її практичну цінність.

1.4 Мультимодальна природа музики: акустика та семантика

Однією з головних складностей розробки інтелектуальних систем у музичній сфері є те, що музика є мультимодальним об'єктом: вона складається з акустичного сигналу та семантичного змісту тексту пісні. Сприйняття пісні слухачем формується на перетині цих двох вимірів.

Акустичний вимір містить інформацію про ритм, гармонію, мелодію та тембр. Добре відомою акустичною ознакою для гармонічного аналізу є хромаграма — 12-вимірне представлення гармонії аудіосигналу, що відображає розподіл енергії по класах висот [23]. Chroma-характеристики та гармонічні зв'язки

у просторі Tonnetz є ключовими для аналізу акордових прогресій і тональності безпосередньо з аудіосигналу [23].

Окремо варто виділити семантичний вимір пісень. На противагу суто інструментальним композиціям, наявність вокалу додає трекам лінгвістичного контексту. Сучасні дослідники з MIR зазначають, що хоча методи NLP активно використовуються для тематичного моделювання, визначення структури пісні та класифікації настрою, їхня роль у розпізнаванні музичних емоцій залишається недооціненою, попри те, що класифікатори емоцій на основі лірики демонструють вищу точність порівняно з аудіо [25]. Дві композиції можуть мати ідентичний акустичний темп і схоже інструментальне забарвлення, але різний емоційний текст — і це визначатиме їхню доречність у плейлисті.

Саме тому жодна рекомендаційна система не може ефективно функціонувати, спираючись лише на один параметр або одну модальність. Музичний досвід є багатовимірним за своєю природою: темп описує ритмічну динаміку, але не визначає емоційного забарвлення; жанр окреслює стилістичну приналежність, але не відображає гармонічної близькості двох треків; текст розкриває смисловий і емоційний зміст, але не враховує акустичну подібність інструментального звучання. Кожен із цих параметрів охоплює лише певний зріз музичного об'єкта, і їхнє ізольоване використання неминуче призводить до поверхневих або нерелевантних рекомендацій. Лише комплексне поєднання акустичних, гармонічних та семантичних сигналів дозволяє системі наблизитися до того, як людина насправді сприймає і порівнює музику — одночасно через звук, ритм, настрій і зміст.

Результати сучасних досліджень підтверджують, що більшість рекомендаційних систем покладалися або лише на одну модальність, або максимум на дві, тоді як комплексне об'єднання акустичного контенту, лірики та тегів залишається малодослідженим [26]. Запропоновані мультимодальні підходи на основі злиття досягають суттєвого покращення порівняно з використанням лише

акустичних чи текстових ознак, що обґрунтовує необхідність розробки систем, які об'єднують обидва виміри у єдиний математичний профіль композиції [26].

1.5 Постановка задачі

Актуальність теми. Порівняльний аналіз (див. підрозділ 1.3) виявив спільну вразливість чотирьох досліджених комерційних платформ: жодна з них не реалізовує повний аналіз звучання композиції без оцінки поведінки користувачів. Spotify та YouTube Music залежать від накопиченої статистики прослуховувань, а Apple Music обмежена ручним тегуванням метаданих. Як наслідок, нові та нішеві композиції систематично випадають із рекомендаційних потоків. Водночас дослідження мультимодальної природи музики (див. підрозділ 1.4) підтвердило, що поєднання акустичного та семантичного аналізу дозволяє оцінювати подібність треків незалежно від їхньої ринкової популярності. Це формує запит на створення системи, вільної від поведінкових даних.

Метою роботи є розробка інтелектуальної системи рекомендацій музичних композицій на основі контентно-гармонічного аналізу, що поєднує аналіз акустичних характеристик аудіосигналу та семантичний аналіз тексту пісень.

Об'єктом дослідження є процес автоматичного формування персоналізованих музичних рекомендацій.

Предметом дослідження є методи контентного аналізу звукових сигналів та обробки природної мови в задачах музичної рекомендації. Для досягнення поставленої мети визначено такі завдання:

– проаналізувати предметну область, виявити обмеження існуючих рекомендаційних систем та обґрунтувати доцільність контент-орієнтованого підходу;

– дослідити та обрати математичні методи перетворення аудіосигналу у числові представлення, придатні для обробки нейронними мережами;

– розробити архітектуру моделей класифікації, що поєднує аналіз візуалізованого звуку (спектрограм) з аналізом числових акустичних ознак, а також інтегрувати модуль семантичного аналізу тексту пісень;

– розробити алгоритм формування рекомендацій на основі обчислення подібності між векторними представленнями композицій;

– здійснити програмну реалізацію системи у вигляді клієнт-серверного застосунку та провести тестування на реальних аудіоданих.

Висновки до розділу 1

Проведений аналіз предметної сфери дозволив встановити, що домінуючі на ринку алгоритми колаборативної фільтрації мають два критичних недоліки: системне популяризаційне упередження, яке ігнорує нішевий контент на користь мейнстриму, та нездатність рекомендувати нові композиції через відсутність поведінкової статистики. Порівняльний аналіз трьох комерційних аналогів (Spotify, YouTube Music, Apple Music) підтвердив, що жодна з існуючих платформ не реалізує повноцінного мультимодального підходу: перші дві платформи критично залежать від накопиченої статистики взаємодій користувачів, тоді як остання обмежується редакційною роботою з метаданими без глибокого аналізу самого аудіосигналу.

Дослідження мультимодальної природи музики показало, що об'єктивна оцінка подібності композицій можлива лише за умови паралельного аналізу двох вимірів: акустичного (гармонічна структура, ритм, тембр) та семантичного (емоційне забарвлення лірики). Ігнорування будь-якого з них призводить до рекомендацій, які сприймаються користувачем як помилка алгоритму. На основі цих висновків сформульовано мету, об'єкт та предмет дослідження, а також визначено п'ять завдань, спрямованих на створення контент-орієнтованої системи, повністю незалежної від метрик популярності.

2 МАТЕМАТИЧНІ МОДЕЛІ ТА МЕТОДИ АНАЛІЗУ МУЗИЧНИХ КОМПОЗИЦІЙ

2.1 Методи цифрової обробки аудіосигналів

Оскільки комп’ютер не може напряму аналізувати аудіосигнал у вигляді безперервної звукової хвилі, першим етапом є застосування методів цифрової обробки сигналів (Digital Signal Processing, DSP). Метою цього етапу є перетворення одновимірного аудіосигналу (амплітуди у часі) у багатовимірні ознаки, придатні для подальшого аналізу моделями машинного навчання.

У розробленій системі використовуються наступні методи перетворень.

Короткочасне перетворення Фур'є (STFT) та мел-спектрограми. Аудіосигнал може бути представлений як сума синусоїдальних компонент різних частот. Для аналізу частотного складу сигналу використовується дискретне перетворення Фур'є (DFT), яке розкладає фрагмент сигналу на окремі частотні компоненти [27]. Оскільки спектр усього сигналу не відображає зміни частот у часі, у системі застосовується STFT — послідовне застосування DFT до коротких вікон з перекриттям, що дозволяє отримати частотно-часове представлення.

Стандартне частотне представлення не враховує особливості людського слуху. Для цього використовується перетворення частот у шкалу мел:

$$m = 2595 \cdot log_{10}\left(1 + \frac{f}{700}\right), \quad (2.1)$$

де f – частота в герцах.

Після застосування STFT з перекриттям вікон та використання мел-фільтрів формується мел-спектрограма. Вона відображає енергетичний розподіл частот у часі та використовується як вхідне представлення для згорткових нейронних мереж [28].

Chroma-ознаки призначені для аналізу структури сигналу. Вони базуються на розподілі звуків за 12 класами висоти тону відповідно до музичних півтонів (від

С до В). Для кожного моменту часу формується 12-вимірний вектор, що відображає енергетичний розподіл між нотами. Це дозволяє визначати тональність та структуру гармонії незалежно від октави або інструмента.

Мел-частотні кепстральні коефіцієнти. MFCC використовуються для опису тембру звуку. Процес їх обчислення включає: попереднє підсилення високих частот, розбиття сигналу на фрейми із застосуванням вікна Ганна, обчислення спектра через перетворення Фур'є, застосування мел-фільтрів, логарифмування енергії та застосування дискретного косинусного перетворення (Discrete Cosine Transform,). Результатом є компактне представлення спектральної огинаючої. У розроблюваній системі використано перші 13 коефіцієнтів, які містять основну інформацію про тембральні характеристики сигналу [29].

Просторова модель Tonnetz. Tonnetz — це метод проєкції акордів на шестивимірний тороїдальний простір. Кожна вісь цього простору відповідає одному з базових гармонічних інтервалів: великій терції, малій терції та квінті. Видобуток Tonnetz-ознак дозволяє виявити складні гармонічні зв'язки між акордами та аналізувати характер переходів у часовій послідовності.

Для кожного з описаних часових рядів (MFCC, Chroma, Tonnetz) додатково обчислюються шість статистичних показників: середнє значення, стандартне відхилення, мінімум, максимум, асиметрія та ексцес. Отриманий числовий вектор (понад 150 значень) нормалізується за допомогою Z-score нормалізації (віднімання середнього та ділення на стандартне відхилення тренувальної вибірки) для приведення всіх ознак до єдиного масштабу.

2.2 Математичні моделі глибокого навчання (ResNet та MLP)

Центральним інтелектуальним елементом системи є ансамбль моделей машинного навчання. В процесі розробки системи було прийнято рішення відмовитися від базових CNN на користь сучасніших та глибших архітектур.

Для досягнення високої точності рекомендацій у роботі застосовано комплексний підхід до аналізу аудіо, що поєднує гармонічний та спектральний

методи. Аналіз структури гармонії, що базується на Chroma-ознаках та тональних характеристиках, забезпечує ідентифікацію семантичного ядра композиції, відповідаючи за визначення її гармонічного портрету. Водночас, для врахування складних акустичних особливостей, тембрального забарвлення та динамічних нюансів, які визначають жанрову специфіку, залучено моделі глибокого навчання (ResNet). Така комбінація методів є взаємодоповнюваною: якщо гармонічний аналіз визначає тональну основу твору, то спектральний аналіз розкриває його звукове забарвлення та жанрові деталі. Поєднання цих підходів у єдиній моделі дозволяє системі здійснювати рекомендації з урахуванням як гармонії, так і акустики, яка в свою чергу також впливає на гармонію, що в сукупності забезпечує глибше розуміння музичного твору, ніж при використанні лише одного метода.

Глибока залишкова мережа. Для аналізу двовимірних мел-спектрограм застосовується архітектура ResNet. У класичних глибоких нейромережах під час навчання виникає проблема згасання градієнта: при зворотному поширенні помилки через велику кількість шарів градієнт зменшується до значень, близьких до нуля, і модель припиняє навчатися [30]. ResNet вирішує цю проблему за допомогою залишкових блоків (residual blocks) із обхідними зв'язками (skip connections). Кожен такий блок навчається не прямому перетворенню вхідного сигналу у вихідний, а залишковій функції, тобто різниці між бажаним виходом і вхідним значенням. Це спрощує оптимізацію та дозволяє будувати значно глибші архітектури без зменшення точності [30] (див. рис. 2.5).

Використання спектрограм як вхідних зображень для ResNet є усталеною практикою в задачах класифікації музичних жанрів [31]. Це дозволило побудувати глибоку архітектуру, здатну виокремлювати складні абстрактні патерни (наприклад, характерний ритм техно або джазові синкопи) з високою точністю.

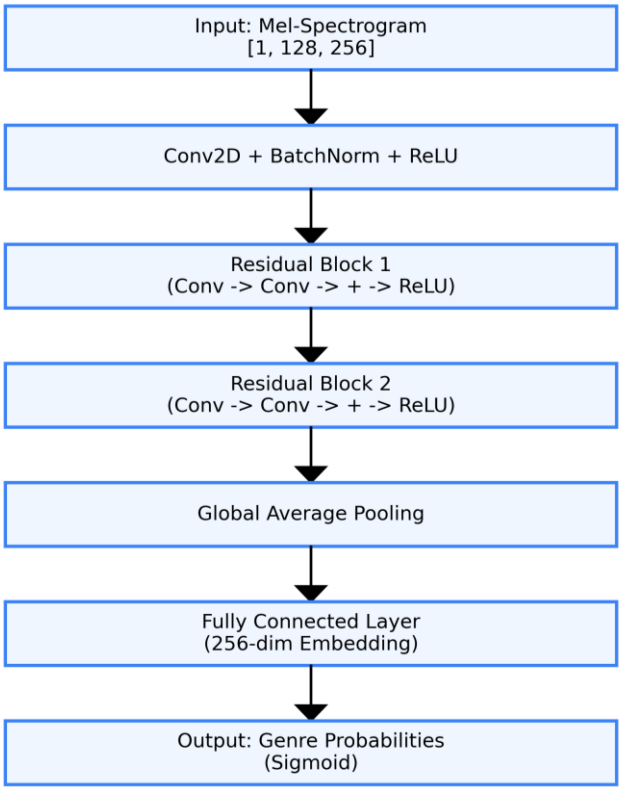

Рисунок 2.5 – Структурна ResNet

Багатошаровий перцептрон. Паралельно з ResNet працює класична нейронна мережа прямого поширення (Feedforward Neural Network, FNN), реалізована у вигляді MLP. На її вхід подається одновимірний вектор числових характеристик (MFCC, Chroma, статистичні метрики). Для реалізації потенціалу багатошарових архітектур необхідним є застосування нелінійних функцій активації до кожного прихованого нейрона після лінійного перетворення; популярним вибором є функція активації ReLU [32]. MLP обробляє числові ознаки через послідовність прихованих шарів з функціями активації ReLU (див. рис. 2.6). Завдання MLP — знаходити приховані нелінійні кореляції між гармонічними показниками, що дозволяє точно класифікувати тональність та жанрові відхилення пісні.

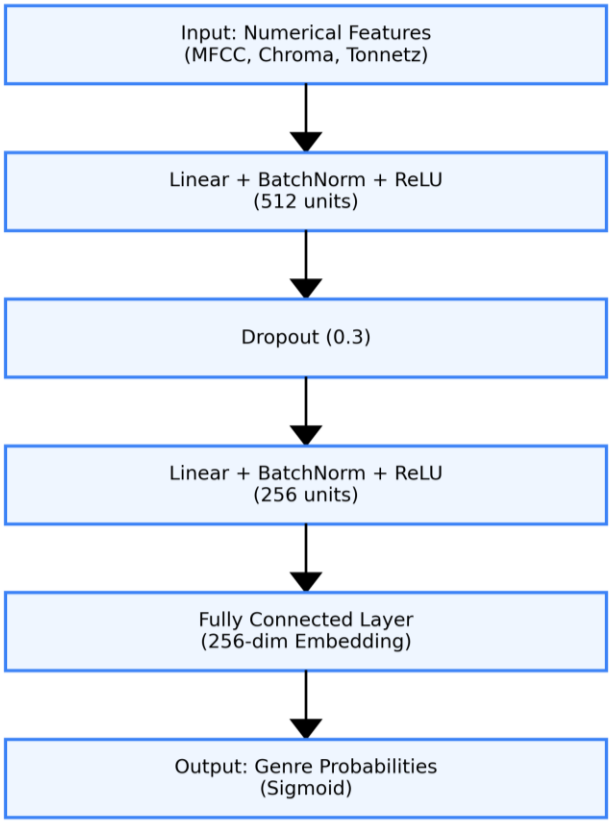

Рисунок 2.6 – Структурна MLP

Обидві моделі генерують числові векторні представлення (ембединги) та видають ймовірності класів (жанрів). Їхні результати об'єднуються методом ансамблевого навчання [33].

2.3 Методи обробки природної мови

Аналіз лірики у розроблюваній системі складається з двох ключових етапів: отримання текстової складової пісні та визначення її емоційного забарвлення.

Для забезпечення максимальної точності та швидкодії, пріоритетним методом отримання тексту є його пряме завантаження через інтегровані зовнішні програмні інтерфейси (LRCLIB API) музичних баз даних. Проте, як надійний резервний механізм для випадків, коли готовий текст відсутній у зовнішній базі, система застосовує алгоритмічний метод автоматичної транскрипції вокалу прямо з аудіосигналу.

Перша задача належить до класу Speech-to-Text (перетворення мовлення у текст). Її складність у контексті музики полягає в тому, що вокал накладається на інструментальний супровід, що значно ускладнює розпізнавання порівняно з аналізом чистого мовлення. Сучасні моделі на базі архітектури Transformer вирішують цю проблему завдяки механізму уваги, який дозволяє моделі фокусуватися на частотних діапазонах, характерних для людського голосу, ігноруючи інструментальні складові.

Друга задача — визначення емоційного забарвлення тексту — вирішується методом класифікації з нульовим навчанням. (Zero-Shot Classification, ZSC). На відміну від класичного підходу, де модель потребує великого розміченого датасету для кожної категорії емоцій, Zero-Shot підхід використовує попередньо навчену мовну модель, яка вже має загальне розуміння семантики тексту. Модель отримує текст та перелік цільових категорій і обчислює ймовірність належності тексту до кожної з них. Це робить систему гнучкою, оскільки набір емоційних категорій можна змінювати без перенавчання моделі.

Отриманий емоційний профіль використовується для уточнення жанрових передбачень. Якщо семантика тексту узгоджується з жанром, визначеним на основі звучання (наприклад, агресивний текст та жанр Metal), відповідна ймовірність підсилюється. Це забезпечує узгодженість між акустичним та семантичним аналізом.

2.4 Метод мультисигнального формування рекомендацій

Для вирішення задачі пошуку релевантних музичних композицій у розробленій системі застосовується **мультисигнальний метод ранжування з механізмами забезпечення різноманітності**. На відміну від традиційних підходів, які часто спираються на єдиний вектор ознак (наприклад, виключно акустична подібність або лише семантична близькість тексту), запропонований метод розглядає музичний твір як сукупність незалежних акустичних та контекстуальних характеристик.

Такий підхід, як поєднання аудіо-сигналів та семантики тексту дозволяє точніше імітувати людське емоційне сприйняття музики, значно перевершуючи системи, що спираються на один тип даних.

Основою методу є обчислення комплексної оцінки подібності S між поточним треком q (запитом) та треком-кандидатом c з бази даних. Ця оцінка формується як зважена сума незалежних функцій подібності для кожного сигналу:

$$s_{bpm}(q,c) = \max\left(0{,}1 - \frac{|BPM_q - BPM_c|}{20}\right), \quad (2.2)$$

де s_{bpm} – коефіцієнт близькості темпу;

BPM_q – значення темпу запиту;

BPM_c – значення темпу кандидата;

20 – порогове значення різниці, при якому сумісність стає нульовою.

Фінальна оцінка релевантності нормалізується діленням на максимально можливу суму ваг, що дозволяє отримати відсоткове або дробове значення подібності від 0 до 1.

Для кожного типу музичних даних застосовується специфічна функція оцінки подібності $s_i(q, c)$: жанрова та семантична сумісність, гармонічна подібність, сумісність тональностей, темпова сумісність, модальна сумісність.

Оскільки **жанрова та семантична сумісність** є категоріальними ознаками, їхня подібність визначається простим збігом: або жанри однакові, або ні. Ці два параметри виконують роль основних фільтрів, які уточнюють результати аналізу гармонії та допомагають відсіяти стилістично несумісні треки.. Якщо класифікований жанр запиту співпадає з кандидатом, $s=1$, інакше $s=0$.

Оскільки головною спеціалізацією системи є створення ідеально зведених музичних потоків (Harmonic Mixing), **гармонічна подібність** має найбільшу вагу. Для оцінки акустичної близькості за гармонією використовується косинусна

подібність між 12-вимірними векторами хромограм (розподіл енергії по 12 нотах хроматичної гами) обох треків:

$$s_{harmony}(q, c) = \frac{v_q \cdot v_c}{|v_q||v_c|}, \quad (2.3)$$

де v_q – вектор ознак запиту (першого об'єкта);

v_c – вектор ознак кандидата (другого об'єкта).

Це дозволяє знаходити пісні, які використовують схожі акордові прогресії, навіть якщо вони знаходяться у різних тональностях, забезпечуючи високу музичну сумісність.

Використання косинусної подібності над Chroma-векторами є найбільш надійним математичним інструментом для автоматизованого Harmonic Mixing, оскільки цей метод не залежить від октави та тембру [16]

Другим за важливістю сигналом є музична **спорідненість тональностей**. Відстань між тональностями визначається як найкоротший шлях на кварто-квінтовому колі (від 0 до 6 кроків). Кварто-квінтове коло — це музична схема, що розташовує всі 12 тональностей по колу таким чином, що сусідні тональності звучать найбільш природно і гармонійно разом. Відстань між двома тональностями на цьому колі показує, наскільки вони акустично сумісні: тональності-сусіди (1 крок) звучать дуже схоже, тоді як протилежні (6 кроків) є найбільш контрастними. Це гарантує, що при переході між піснями не виникатиме дисонансу. Функція подібності має лінійне спадання залежно від відстані *d*:

$$s_{key}(q, c) = \max\left(0{,}1 - \frac{d(q_{key}, c_{key})}{6}\right), \quad (2.4)$$

де s_{key}– міра близькості тональностей музичних творів;

q_{key}, c_{key} – тональності запиту та кандидата відповідно;

$d(q_{key}, c_{key})$ – відстань між тональностями на квінтовому колі;

6 – максимальна можлива відстань у квінтовому колі, при якій сумісність стає рівною нулю.

Подібність за темпом оцінюється на основі абсолютної різниці ударів на хвилину. Використовується функція лінійного спадання з пороговим значенням (наприклад, 20 BPM), після якого подібність вважається нульовою:

В **модальній сумісності** враховується лад композиції. Якщо обидва треки мажорні або обидва мінорні, *s=1*. У разі розбіжності ладів застосовується штрафний коефіцієнт (наприклад, *s=0.3*), що дозволяє уникати різких емоційних перепадів.

Класичні рекомендаційні системи часто страждають на проблему детермінованості: для одного й того ж запиту вони завжди повертають ідентичний список кандидатів. Така поведінка призводить до утворення так званої «бульбашки фільтрів» (filter bubble) — ситуації, коли система постійно пропонує однотипний контент, поступово звужуючи музичний кругозір користувача. Щоб уникнути цього та зробити прослуховування більш непередбачуваним, у запропонованому методі впроваджено механізм диверсифікації

Стохастичне збурення оцінок реалізується шляхом додавання до обчисленої комплексної оцінки гауссівського шуму із середнім значенням 0 та невеликим стандартним відхиленням:

$$S'_{\text{total}} = S_{\text{total}} + \mathcal{N}(0, \sigma^2), \tag{2.5}$$

де S'_{total} – фінальна (збурена) оцінка ранжування;

S_{total} – початкова розрахункова оцінка релевантності;

$\mathcal{N}$ – випадкова величина, що підпорядковується нормальному розподілу з нульовим математичним сподіванням та дисперсією σ^2

σ – параметр, що визначає рівень випадковості (ступінь «розмитості» ранжування).

Це призводить до того, що треки з приблизно однаковою релевантністю будуть змінювати свій порядок при кожному новому запиті, постійно відкриваючи користувачеві новий контент.

Обмеження домінування авторів. Для формування підсумкового плейлиста використовується жадібний алгоритм вибірки з широкого набору топових кандидатів (наприклад, топ-50). Під час вибірки застосовується обмеження (Artist Cap): до підсумкового списку може потрапити не більше *K* треків від одного й того самого виконавця. Це гарантує стилістичну та авторську різноманітність рекомендацій.

2.5 Обґрунтування вибору інформаційних технологій

Для реалізації інтелектуальної системи було обрано комплексний стек технологій. Головним критерієм вибору стала здатність інструментів забезпечувати високу швидкість обробки великих масивів аудіоданих, гнучкість у проєктуванні нейромережевих архітектур та загальна масштабованість сервісу.

Базовою мовою програмування обрано Python (version 3.10.0). На сьогодні це загальновизнаний стандарт галузі у сфері машинного навчання та Data Science, що володіє розвиненою екосистемою бібліотек. Для попередньої обробки наборів даних та швидких матричних обчислень задіяно фундаментальні бібліотеки NumPy та Pandas, а для візуалізації результатів (зокрема побудови графіків точності роботи моделей) — бібліотеки Matplotlib.

Аналіз акустичного виміру музики виконано за допомогою бібліотеки *librosa*, яка дозволяє з високою точністю екстрагувати гармонічні ознаки та генерувати мел-спектрограми із не оброблених аудіофайлів. Безпосередня розробка та тренування нейронних мереж здійснюється на базі фреймворку PyTorch. Вибір PyTorch зумовлений його можливістю динамічно керувати обчислювальними графами та ефективно використовувати апаратне прискорення відеокарт для зменшення часу тримання передбачень та навчання моделей [18].

Оскільки розроблювана система оперує не лише звуком, а й текстом пісень, критично важливим компонентом є інструменти обробки природної мови (NLP). Для розпізнавання тексту з аудіо (у разі його відсутності в базах даних) інтегровано модель Whisper від OpenAI. Водночас для семантичного аналізу лірики та виділення емоційного забарвлення застосовуються сучасні трансформерні мовні моделі через бібліотеку Transformers від Hugging Face, де використовується модель DistilBERT для ZSC. Це дозволяє трансформувати текстову складову пісень у набір конкретних емоційних ознак із збереженням глибокого контексту.

Використання трансформерів дозволяє аналізувати зміст лірики та розпізнавати неявні емоції краще за класичні рекурентні мережі

Бекенд частину системи реалізовано за допомогою FastAPI — сучасного асинхронного веб-фреймворку. Він гарантує мінімальні затримки при обробці користувацьких запитів, відмінно підтримує асинхронну роботу фонових задач (що критично для завантаження та тривалої AI-обробки аудіо). Взаємодія між програмними модулями та клієнтом відбувається через REST-архітектуру.

Для зберігання даних було обрано реляційну систему керування базами даних PostgreSQL версії 15 (Alpine), яка розгортається у Docker-контейнері за допомогою Docker Compose. Вибір PostgreSQL обумовлений підтримкою підтримкою ACID-транзакцій (атомарність, узгодженість, ізольованість, довговічність), що гарантує цілісність даних при паралельному записі, та наявністю вбудованих механізмів індексування для прискорення пошукових запитів рекомендаційного движка. Взаємодія з базою даних на рівні програмного коду здійснюється через сумісну ORM-бібліотеку SQLAlchemy.

Фронтенд частина розроблена на базі бібліотеки React. Таке архітектурне рішення дає змогу створити реактивний, швидкий та інтуїтивно зрозумілий інтерфейс для зручного пошуку музики, відтворення треків та візуалізації персоналізованих рекомендацій без потреби постійного перезавантаження сторінок веб-додатку.

Висновки до розділу 2

Аналіз методів цифрової обробки сигналів дозволив визначити оптимальний набір акустичних представлень: мел-спектрограми для аналізу загальної звукової картини засобами CNN, Chroma-вектори та простір Tonnetz для аналізу гармонії, MFCC для опису тембру.

Обґрунтовано вибір архітектури ResNet: механізм залишкових зв'язків усуває проблему згасання градієнта, що дозволяє будувати глибокі моделі для аналізу спектрограм. Паралельне використання MLP для обробки числових ознак забезпечує формування збалансованого ансамблю двох моделей, де ResNet відповідає за аналіз загальної звукової картини, а MLP деталізує та уточнює тональні характеристики композиції.

Доведено доцільність інтеграції DistilBERT у режимі ZSC для семантичного аналізу: підхід не потребує попередньо розмічених даних та здатний класифікувати емоційне забарвлення тексту довільною мовою. Косинусна подібність обрана як метод формування рекомендацій, оскільки вона оцінює структурну подібність композицій незалежно від їхньої тривалості чи гучності.

3 РОЗРОБКА РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ ТА АНАЛІЗ ОТРИМАНИХ РЕЗУЛЬТАТІВ

3.1 Формування, категоризація та складання датасету для навчання моделей

Фундаментом для розробки будь-якої моделі машинного навчання є якісний та репрезентативний набір даних. На початковому етапі було проаналізовано існуючі відкриті академічні датасети, які традиційно використовуються для завдань класифікації музики. Як приклад було розглянуто набір GTZAN, який є одним із найбільш цитованих у сфері MIR. В ході вивчення структури та вмісту датасету було встановлено, що він має низку обмежень для сучасних рекомендаційних систем. Записи у ньому обмежені тривалістю у 30 секунд, а класифікація охоплює лише 10 узагальнених жанрів (*Rock, Pop, Jazz* тощо), що не дозволяє сучасній системі розрізняти тонкі стилістичні та емоційні відтінки всередині одного макрожанру.

Паралельно було досліджено інший підхід для збору даних, який полягав у масовому завантаженні вже готових жанрових плейлистів безпосередньо з музичної платформи YouTube Music. Передбачалося, що офіційні плейлисти, сформовані за жанровими мітками платформи, забезпечать швидке отримання великого обсягу розсортованих аудіоданих. Проте в ході експерименту виявилось, що жанрова розмітка цих плейлистів є неточною: в межах одного списку часто зустрічались композиції з принципово різними акустичними характеристиками. Наприклад, плейлист з міткою *Rock* міг одночасно містити класичний рок 70-х, сучасний інді-рок та пост-панк, які мають відмінні тембральні та ритмічні профілі. Така неоднорідність робила автоматично зібрані плейлисти непридатними для навчання моделей, що потребують чистих жанрових міток.

Зважаючи на виявлені обмеження як академічних датасетів, так і автоматизованих джерел, було прийнято рішення сформувати власний набір музичних даних з детальною ієрархією піджанрів.

Процес формування датасету складався з наступних етапів.

1. Вибір піджанрів та поглиблення до піджанрів. На першому етапі розробки датасет було розділено на 9 основних піджанрів (*Pop, Rock, Hip-Hop, EDM, Metal* тощо). Однак, в процесі навчання нейромереж стало очевидно, що такі широкі категорії викликають плутанину в моделі. Наприклад, класичний поп та к-поп акустично мають різні тембральні картини, хоча належать до одного макрокласу.

Тому архітектуру датасету було переглянуто та розширено до 37 специфічних піджанрів. Серед них:

- у межах сімейства Rock (7 піджанрів): *Classic Rock, Modern Metal, Punk Hardcore, Post-Punk, Grunge, Alt-Rock, Prog Rock;*
- у межах сімейства Electronic (6 піджанрів): *House/Techno, Dubstep/Bass, Drum and Bass, Retrowave, Darksynth, Ambient IDM;*
- у межах сімейства Hip-Hop (5 піджанрів): *Oldschool Hip-Hop, Trap, Drill, Cloud Rap, Phonk;*
- у межах сімейства Pop (4 піджанри): *Mainstream Pop, K-Pop, Indie Pop, Hyperpop;*
- у межах сімейства Other (4 піджанри): *Lofi/Chillhop, Reggaeton/Latin, Folk/Acoustic, Midwestern Emo;*
- у межах сімейства Classical (3 піджанри): *Orchestral, Opera, Chamber Piano;*
- у межах сімейства Jazz (3 піджанри): *Classic Jazz, Smooth Jazz, Jazz Fusion;*
- у межах сімейства Asian (3 піджанри): *J-Rock, Anime OST, City Pop;*
- у межах сімейства R&B (2 піджанри): *Classic R&B, Modern R&B.*

Така глибока деталізація забезпечила виділення моделями найменших акустичних відмінностей в гармонії та ритмі.

2. Збір аудіофайлів та розширення обсягу даних. Відбір музичних треків здійснювався з відкритих джерел за напівавтоматизованою методикою. Для кожного з 37 піджанрів було вручну сформовано текстовий перелік композицій, що

однозначно належать до відповідної категорії. Відбір проводився на основі ручного прослуховування та оцінки стилістичної відповідності кожного треку обраному піджанру. Далі сформовані переліки оброблялися скриптом пакетного завантаження *batch download script*, який автоматично знаходив кожну композицію за назвою, завантажував аудіофайл та зберігав його у відповідну директорію піджанру.

На початковому етапі для кожного піджанру було відібрано тестову вибірку треків. Однак для уникнення ефекту перенавчання складних моделей датасет потребував масштабного розширення.

Було проведено кілька ітерацій дозбору даних, під час яких дотримувалась вимога щодо високої акустичної якості треків (уникалися артефакти сильного стиснення) та їх чіткої приналежності до одного з 37 піджанрів без міжжанрових змішувань. Для досягнення цільового обсягу вибірки та підвищення стійкості нейромережевих моделей було застосовано методи офлайн-аугментації аудіоданих. Зокрема, використано алгоритм зміщення висоти тону (Pitch Shift) на +2 півтони, що дозволяє системі розпізнавати жанрові ознаки незалежно від тональності виконання. Також застосовувалася зміна темпу (Time Stretch) з коефіцієнтом 1.1 (прискорення на 10%), що забезпечило стабільну класифікацію композицій з різною ритмічною динамікою. У результаті проведених заходів для кожного з 37 піджанрів було сформовано базу з **450 композиціями на кожен піджанр**, а загальний обсяг датасету для навчання моделей склав **16 650 повноцінних музичних композицій.**

3. Зберігання та маркування. Усі завантажені треки були фізично розсортовані у відповідні директорії за назвами піджанрів. Така структура забезпечила автоматичне присвоєння міток класів під час програмного парсингу даних скриптами.

У результаті було створено структурований набір даних загальним обсягом 16 650 композицій. Запропонований підхід до класифікації з використанням 37

піджанрів забезпечив необхідну деталізацію даних для подальшого навчання та тестування алгоритмів у рамках даної робот.

3.2 Архітектура програмного забезпечення та логіка серверної частини

Центральним ядром бекенду є скрипт *server.py* (див. додаток А), який ініціалізує середовище та керує життєвим циклом застосунку. Для оптимізації роботи сервера всі моделі та конфігурації завантажуються у пам'ять одноразово під час запуску застосунку, що робить неможливим їх повторну ініціалізацію при кожному новому запиті. Його суть полягає в тому, що всі нейромережі та допоміжні конфігурації завантажуються у пам'ять сервера лише один раз, під час його запуску. Завдяки цьому кожен новий користувацький запит на аналіз пісні опрацьовується миттєво, без необхідності повторної ініціалізації моделей.

Процес аналізу нової композиції розпочинається після того, як сервер отримує від клієнтської частини відповідний запит на обробку треку. Цей робочий цикл складається з таких послідовних кроків:

– отримання системою аудіоданих композиції, яка додається для аналізу, перевірка цілісності даних та їхнє завантаження в середовище сервера для подальшої обробки;

– передача аудіоданих модулям для отримання музичних характеристик;

– підготовка даних, їх нормалізація та проведення через ансамбль моделей;

– відправка текстових даних до мовних моделей для семантичного аналізу;

– агрегація отриманих результатів (ймовірностей жанрів, емоційного профілю, мови) та серіалізація їх у формат JSON для відправки клієнту;

– автоматичне видалення тимчасового аудіофайлу для очищення дискового простору.

3.3 Програмна реалізація модуля попередньої обробки та отримання музичних характеристик

Перш ніж звук потрапляє у нейромережі, він проходить стадію математичних перетворень у модулі *extractor.py* (див. додаток Б).

Система не аналізує весь трек повністю, що важливо для оптимізації обчислювальних ресурсів. Замість цього застосовано автоматичний відбір найінформативнішого фрагмента аудіо. Аудіо завантажується за допомогою *librosa.load* із фіксованою частотою дискретизації 22050 Гц та зведенням стерео-каналів у моно. Далі алгоритм розраховує середньоквадратичну енергію (RMS), що дозволяє системі оцінити рівень гучності та насиченості звуку в кожний момент часу. Застосовуючи математичну згортку *np.convolve*, система знаходить фрагмент тривалістю рівно 60 секунд, яке має найвищу концентрацію енергії. Зазвичай це забезпечує потрапляння в аналізатор найбільш інформативної частини пісні, а саме приспіву або кульмінації, відкидаючи тихі вступи та закінчення

З отриманого фрагмента програмно видобуваються два типи даних: одновимірні та двовимірні дані.

Двовимірні зображення у вигляді мел-спектрограм формуються шляхом переведення сигналу у частотно-часову область за допомогою STFT. Частотна вісь логарифмується до мел-шкали (*n_mels=128*), оскільки вона найбільше відповідає людському слуху. Отримана матриця переводиться у логарифмічний масштаб (децибели) та жорстко нормалізується до діапазону [0; 1]. Спектрограма форматується до фіксованої ширини (256 пікселів по осі часу) (див. рис. 3.1).

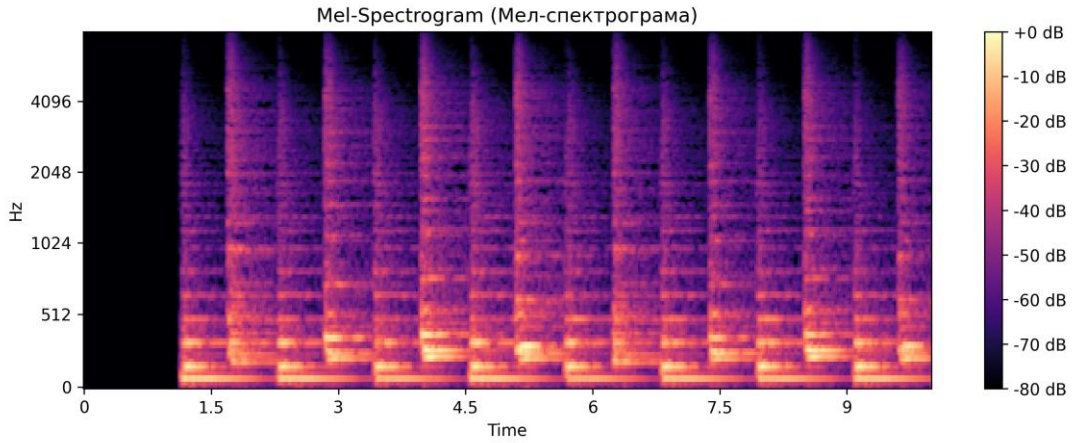

Рисунок 3.1 – Візуалізація мел-спектрограми аудіосигналу

Одновимірний вектор акустичних ознак формується за допомогою функцій *librosa* і містить понад 150 значень. Програма обчислює:

– мел-частотні кепстральні коефіцієнти (MFCC) — для опису тембру інструментів (див. рис. 3.2);

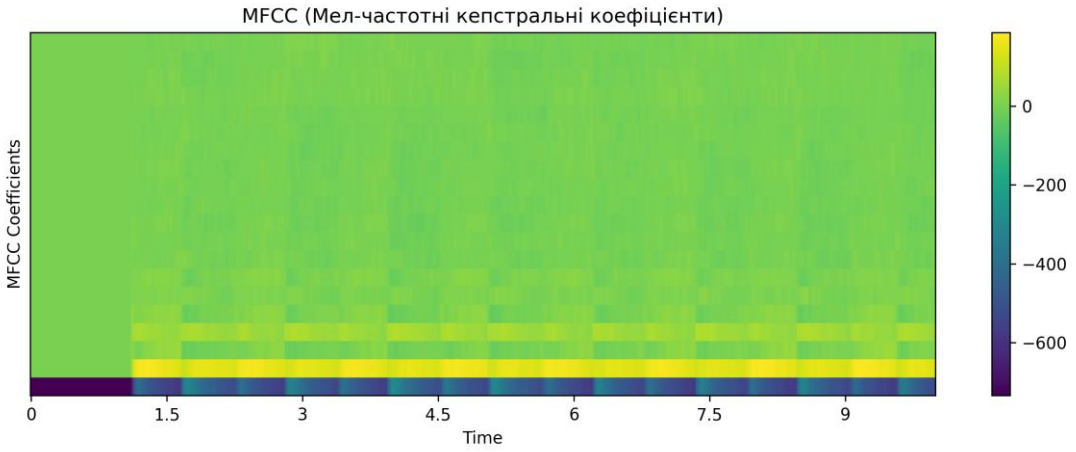

Рисунок 3.2 – Візуалізація MFCC

– chroma-вектори — розподіл енергії по 12 нотах хроматичної гами (від C до B), що є прямим математичним поданням акордів (див. рис. 3.3);

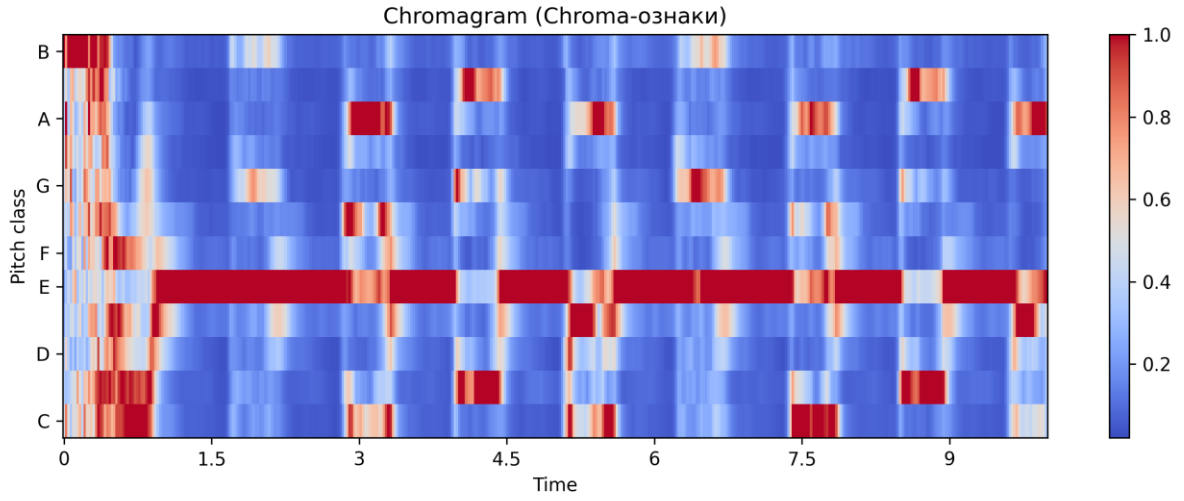

Рисунок 3.3 – Розподіл гармонічної енергії композиції

– ознаки Tonnetz — проекція гармонічних зв'язків між нотами у шестивимірний математичний простір (див. рис. 3.4);

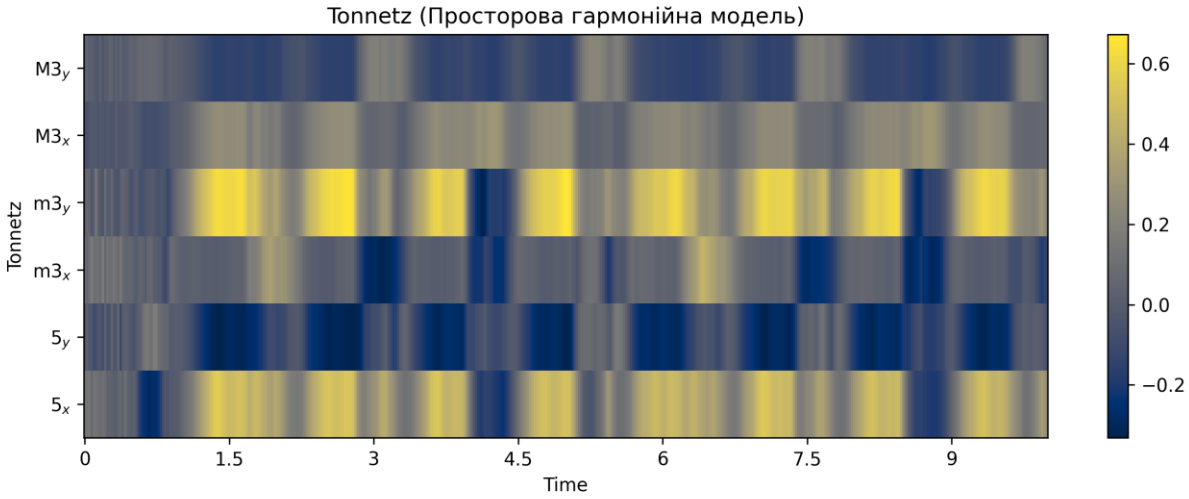

Рисунок 3.4 – Проекція тональних зв'язків у 6-вимірному просторі Tonnetz

– для кожного часового ряду програма додатково розраховує 6 статистичних показників: середнє значення, стандартне відхилення, мінімум, максимум, асиметрію та ексцес.

Отриманий числовий вектор нормалізується шляхом віднімання середнього значення та ділення на стандартне відхилення тренувальної вибірки, що приводить

усі ознаки до єдиного масштабу. Екстремальні значення вектора додатково обмежуються допустимим діапазоном для виключення аномальних викидів, що можуть спотворити результати класифікації.

3.4 Програмна реалізація ансамблю нейронних мереж (ResNet та MLP)

Процес розпізнавання жанрів та гармонічної складності базується на одночасній роботі двох архітектур, написаних на фреймворку PyTorch. Навчання ансамблю нейромереж здійснювалося на зібраному раніше датасеті з 16 650 композицій. Дані було випадковим чином розділено на тренувальну (80%), валідаційну (10%) та тестову (10%) вибірки. У результаті тренування ансамбль моделей продемонстрував високу здатність до узагальнення, досягнувши точності класифікації на рівні 70,9% на незалежній тестовій вибірці.

ResNet приймає на вхід тензор мел-спектрограми розмірністю [1, 1, 128, 256]. MLP найкраще справляється з пошуком кореляцій між статистичними гармонічними параметрами, розпізнаючи тональність та складність акордів. Графік зміни функції втрат та точності класифікації моделі ResNet по епохах наведено на рис. 3.5.

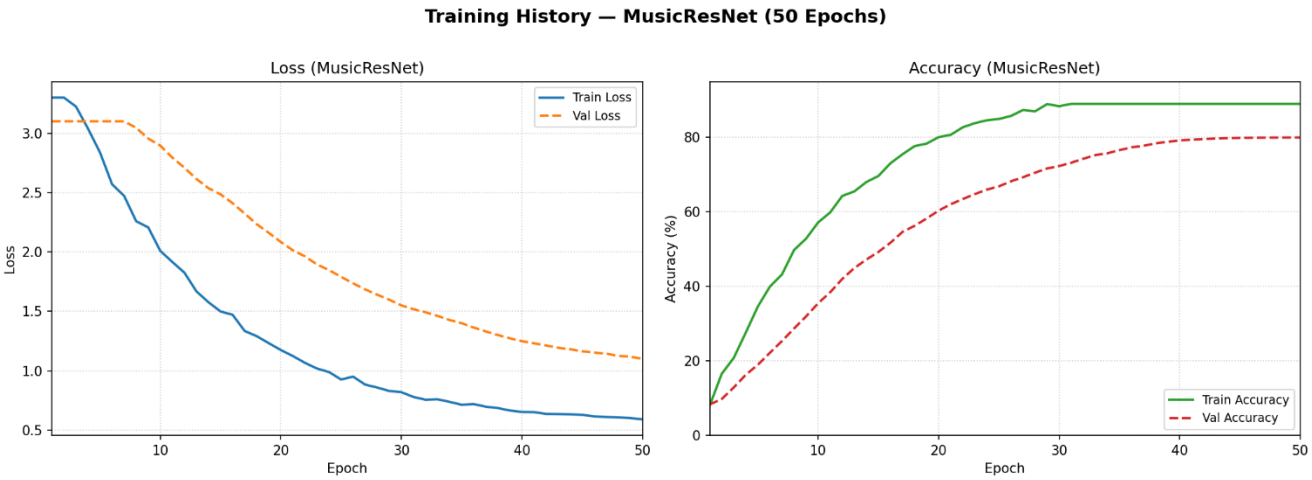

Рисунок 3.5 – Динаміка функції втрат та точності класифікації моделі ResNet у процесі навчання

Паралельно нормалізований числовий масив з 150+ ознак подається на вхід MLP. Графіки навчання MLP із застосуванням раннього зупинення наведено на рис. 3.6.

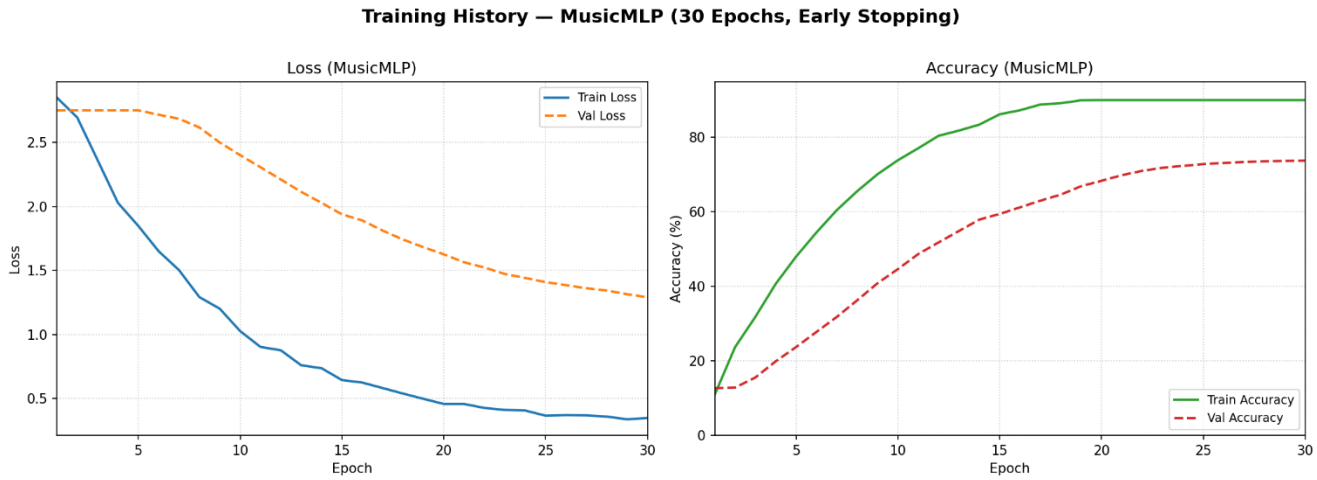

Рисунок 3.6 – Динаміка функції втрат та точності класифікації моделі MLP у процесі навчання

Кожна з мереж на передостанньому шарі генерує 256-вимірний вектор, а на останньому шарі пропускає результати через функцію активації сигмоїда, повертаючи вектор ймовірностей для кожного з визначених жанрів (від 0,0 до 1,0). Кінцевий результат обчислюється шляхом зваженого додавання: 70% вагомості надається висновкам ResNet і 30% надається висновкам MLP.

Вибір співвідношення ваг 70 на 30 між ResNet та MLP обґрунтовано експериментально з урахуванням природи передбачень кожної моделі. ResNet аналізує мел-спектрограму цілісно і, як правило, формує розподілений список ймовірностей (наприклад, Metal 26%, Rock 30%, Alt-Rock 18%), де жоден жанр не отримує абсолютної впевненості, але загальна картина відображає реальну акустичну близькість композиції до кількох стилів одночасно. MLP натомість, працюючи з числовими гармонічними характеристиками, здатна точно розрізнити тонкі гармонічні відмінності між піджанрами і схильна до категоричних

передбачень (наприклад, Metal 100%), що відіграє вирішальну роль у формуванні фінального результату.

Розглянемо конкретний приклад. Нехай ResNet повертає Metal 26%, Rock 30% — правильний жанр присутній у списку, але не є домінуючим. Якщо MLP у цей момент також передбачає Metal з високою ймовірністю, зважене додавання підсилює Metal до лідируючої позиції у фінальному результаті. Проте якщо ваги моделей були б рівними (50/50), а MLP помилилась і повернула хибний результат зі стовідсотковою впевненістю, вона повністю пригнітила б розподіл ResNet. Обмеження ваги MLP до 30% вирішує цю проблему: модель зберігає свою роль, але її категоричність у разі помилки не є достатньою для спотворення загального результату. Таким чином, співвідношення 70/30 забезпечує оптимальний баланс між точністю спектрального аналізу та гармонічним аналізом. Результати навчання ансамблю із зваженим агрегуванням наведено на рис. 3.7.

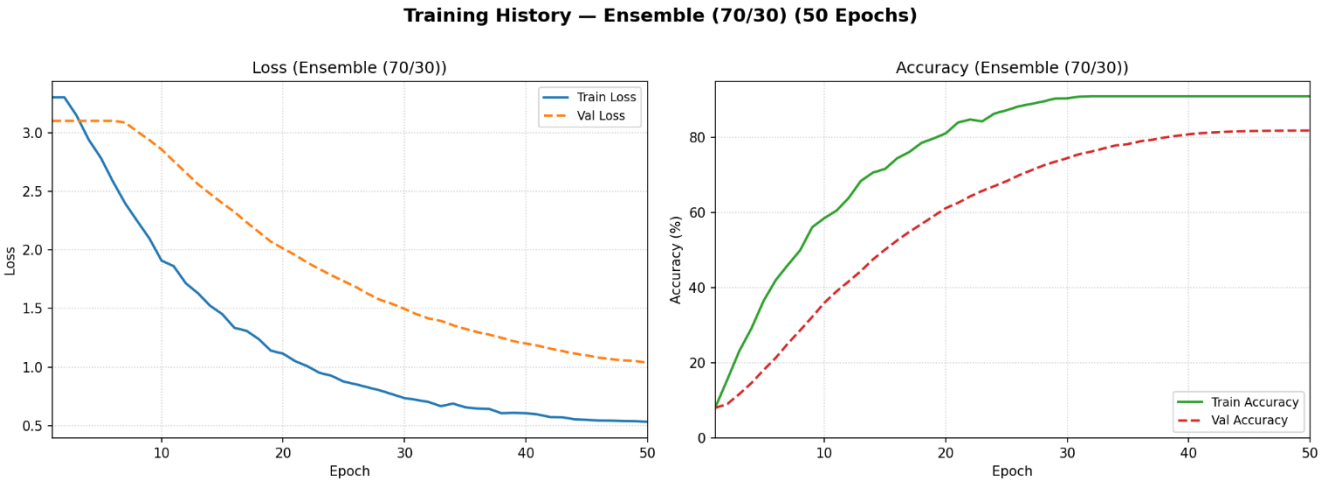

Рисунок 3.7 – Проекція тональних зв'язків у 6-вимірному просторі Tonnetz

Приклад роботи обох моделей з визначення ймовірності жанрів для обраної композиції наведено на рис. 3.5. На зображенні продемонстровано, як система розкладає пісню на складові відсотки, визначаючи домінантний піджанр та наявність суміжних стилістичних домішок.

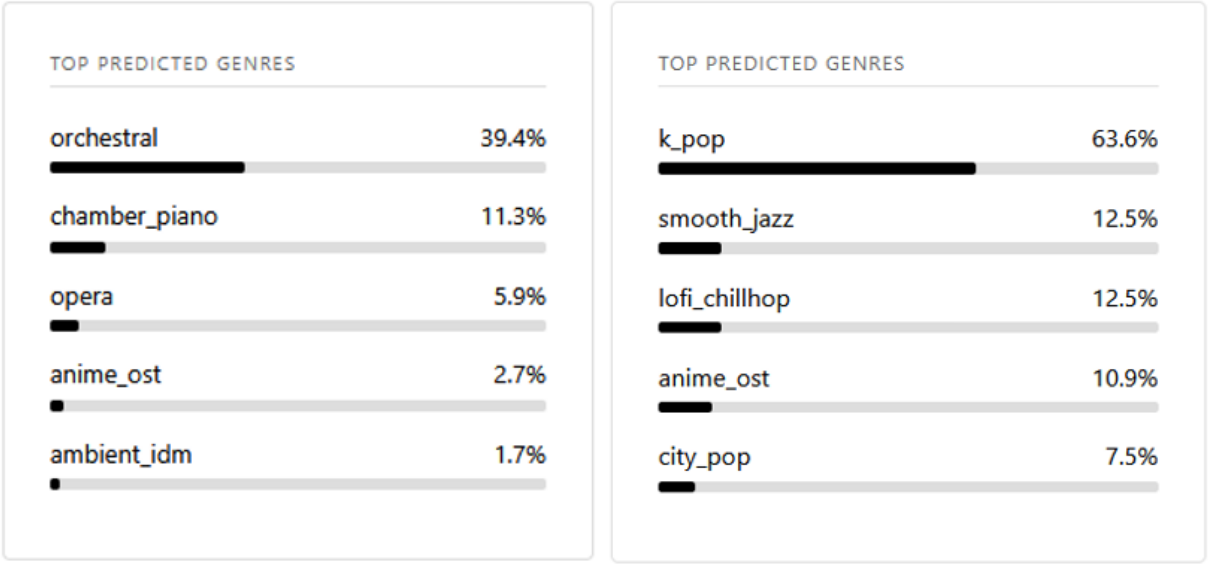

Рисунок 3.8 – Приклад розрахунку ймовірностей належності музичної композиції до різних піджанрів

Таким чином, розроблений ансамбль поєднує переваги двох принципово різних підходів до аналізу музики. Їх зважена взаємодія дозволяє досягти точності класифікації 70,9% на незалежній тестовій вибірці, що є конкурентним результатом для задачі класифікації 37 піджанрів.

3.5 Інтеграція інструментів NLP-аналізу семантики та Speech-to-Text моделей

Процес отримання лірики в розробленій системі реалізовано як послідовність кількох етапів із резервними механізмами. Для мінімізації обчислювальних витрат та пришвидшення роботи, першим пріоритетним кроком є прямий пошук тексту пісні через відкритий програмний інтерфейс музичної бази LRCLIB API. Пошук здійснюється на основі очищених метаданих (імені виконавця та назви треку), отриманих з YouTube Music API. Якщо оригінальний текст успішно знайдено, система автоматично перекладає його англійською мовою за допомогою бібліотеки *deep_translator* (на базі сервісу Google Translate). Такий переклад є необхідним кроком для приведення тексту до єдиного формату перед тим, як передати його на вхід англомовній моделі аналізу емоцій.

У разі, якщо пісня є маловідомою і текст відсутній у базі LRCLIB, система автоматично активує резервний механізм — алгоритмічну транскрипцію аудіо. Її реалізовано через інтеграцію з рішенням Speech-to-Text від OpenAI (модель Whisper). Whisper має вбудовану здатність автоматично визначати мову виконання та одночасно здійснювати машинний переклад отриманого тексту на англійську, що ідеально вписується в існуючий конвеєр обробки. Приклад результату транскрипції, роботи API та визначення мови в інтерфейсі системи наведено на рис. 3.6.

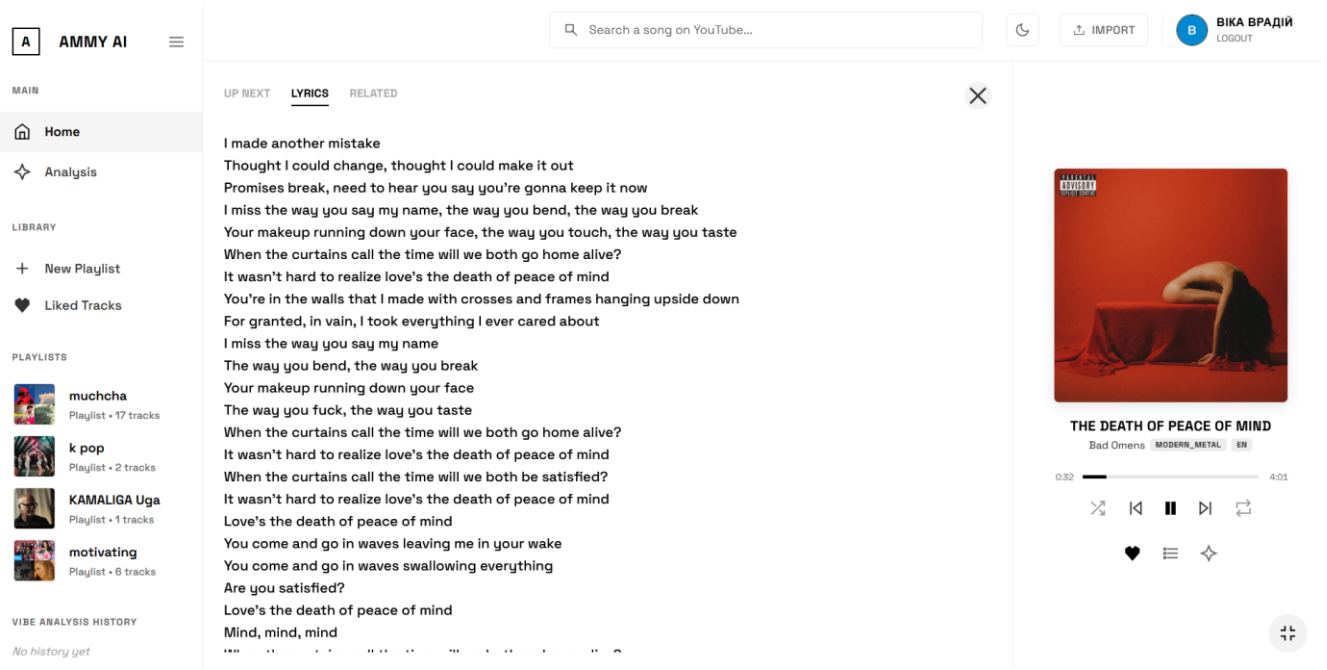

Рисунок 3.9 – Приклад відображення розпізнаного тексту та мови виконання композиції

У випадках, коли алгоритми API не виявляють вокалу в аудіосигналі, трек автоматично позначається спеціальною міткою «Instrumental» (див. рис. 3.7). За наявності такої мітки подальший семантичний аналіз тексту не виконується, оскільки єдиним джерелом інформації для моделі залишається акустична складова пісні.

Рисунок 3.10 – Позначення інструментальної композиції в інтерфейсі системи

Отриманий англомовний текст передається до класу *VibeExtractor*, який ініціалізує попередньо навчену мовну модель. Класу передається фіксований набір з 14 емоційних категорій (Vibes): *Energetic, Sad, Motivating, Melancholic, Playful, Aggressive, Romantic, Chill, Dark, Uplifting, Angry, Dreamy, Epic, Nostalgic*.

Класифікація відбувається за принципом співставлення сенсів: алгоритм аналізує семантику пісенного тексту і визначає ймовірність математичного збігу змісту лірики з кожною із вказаних категорій без потреби у попередньому навчанні на музичних даних. Модель повертає ймовірності для кожної категорії, з яких обираються два найвищих значення — основний та додатковий настрій (primary vibe та secondary vibe).

Результат семантичного аналізу передається до модуля класифікатора, який виконує фінальне коригування передбачень. Програмна логіка цього модуля базується на трьох незалежних механізмах впливу на кінцевий результат розпізнавання. По-перше, застосовуються емоційні множники: для кожної категорії настрою прописані вагові коефіцієнти. Наприклад, якщо текст визначається як *Aggressive*, система штучно підвищує ймовірності для важких жанрів (таких як *Metal* або *Punk*) та зменшує для спокійних (наприклад, *Lofi*). По-друге, визначена мова виконання автоматично підвищує ймовірність відповідних регіональних жанрів. (наприклад, розпізнана японська мова підвищує ймовірність *J-Pop*). По-третє, система застосовує комбіновані мовно-емоційні правила, де одночасно враховується мова виконання та емоційний настрій тексту. Наприклад, поєднання корейської мови та емоції *Energetic* суттєво посилює ймовірність

класифікації треку як *K-Pop*. Завдяки такій багаторівневій корекції досягається максимальна точність і узгодженість між музичним звучанням та змістовим наповненням пісні.

3.6 Формування та наповнення бази даних рекомендацій

Ключовою складовою інтелектуальної системи рекомендацій є спеціалізована база даних, що зберігає результати комплексного аналізу музичних композицій. На відміну від традиційних рекомендаційних систем, які спираються на історію прослуховувань або колаборативну фільтрацію, розроблена система оперує об'єктивними акустичними характеристиками кожного треку — тональністю, ладом, темпом, домінуючими нотами та емоційним забарвленням тексту. Для забезпечення цього підходу було спроєктовано реляційну базу даних, розроблено автоматизовану послідовність обробки її наповнення та реалізовано механізм валідації результатів.

Для розгортання та забезпечення працездатності бази даних використовується технологія контейнеризації Docker. Контейнер працює на базі системи керування базами даних PostgreSQL, створює базу даних *ammy_music* та реалізує механізм збереження файлів у виділеному томі даних, що гарантує постійне збереження інформації між перезапусками контейнера. Програмну взаємодію з базою даних реалізовано через відповідний інтерфейс доступу до даних. Схему бази даних наведено на рис. 3.8.

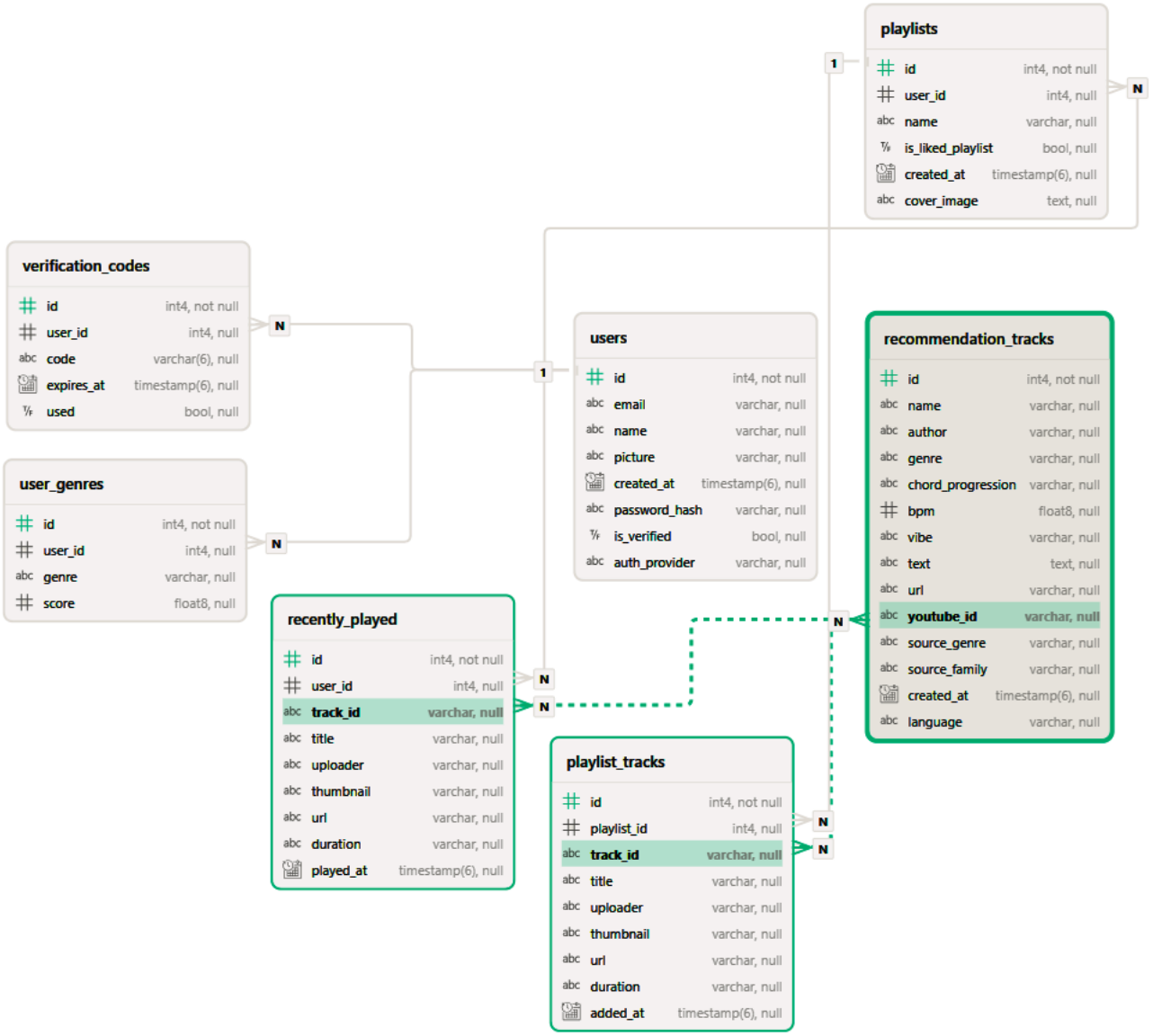

Рисунок 3.11 – ER-діаграма бази даних рекомендацій

Центральною таблицею для рекомендаційної підсистеми є *recommendation_tracks*, що зберігає результати повного AI-аналізу кожної музичної композиції. ORM-модель цієї таблиці реалізовано засобами SQLAlchemy і містить поля, описані у табл. 3.1.

Таблиця 3.1 – Структура таблиці recommendation_tracks

Поле	Тип	Індекс	Призначення
id	Integer	PK	Первинний ключ, автоінкремент

Кінець таблиці 3.1

Поле	Тип	Індекс	Призначення
Name	String		Назва музичної композиції
author	String		Ім'я виконавця
genre	String	Так	Жанр, класифікований ResNet + MLP
chord_progression	String		Гармонічна структура у форматі «тональність_лад.нота1.нота2.нота3»
bpm	Float		Темп композиції (ударів на хвилину)
vibe	String	Так	Емоційний настрій, визначений NLP-аналізом тексту
text	Text		Повний текст пісні (nullable)
language	String		Мова тексту
url	String		Посилання на відео YouTube
youtube_id	String	Unique	Унікальний ідентифікатор YouTube для дедуплікації
source_genre	String		Оригінальна жанрова мітка з датасету
source_family	String		Рід музики з датасету (Rock Family, Pop Family тощо)
created_at	DateTime		Дата та час індексації запису

Особливу увагу слід звернути на поле chord_progression, яке є одним з ключових для алгоритму рекомендацій. Значення містить тональність і лад композиції до першої крапки, а три ноти після крапок є домінуючими, відсортованими за середньою хроматичною енергією, наприклад: «A#_major.D.A.F#». Частина до крапки визначає тональність та лад, ноти після крапок відображають акордову основу композиції. Такий компактний запис дозволяє системі зчитувати тональну інформацію без повторного аналізу аудіофайлу.

Поля *genre* та *vibe* мають індекси для прискорення фільтрації при формуванні рекомендацій, а поле *youtube_id* має обмеження унікальності, що запобігає появі дублікатів у базі даних.

Інші таблиці бази даних забезпечують функціонал користувацького інтерфейсу: таблиця users зберігає профілі автентифікованих користувачів; *user_genres* зберігає жанрові вподобання, обчислені на основі історії прослуховувань; *recently_played* є журналом останніх відтворених композицій; *playlists ma playlist_tracks* містять створені користувачем плейлисти та їх вміст (див. рис. 3.9).

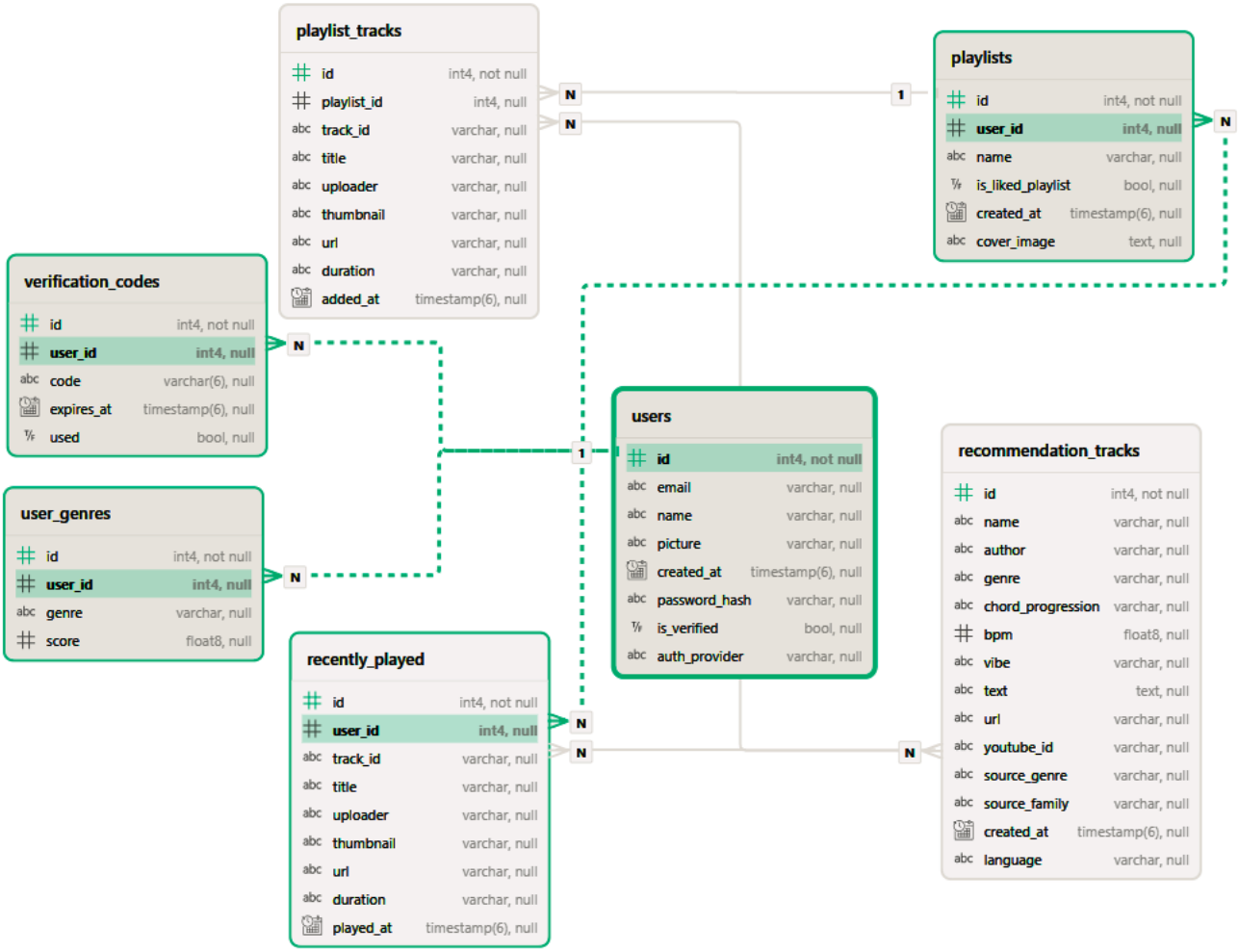

Рисунок 3.12 – ER-діаграма бази даних користувача

Вхідними даними для первинного наповнення бази даних слугують дев'ять текстових файлів датасету, які були сформовані на етапі категоризації

(див. підрозділ 3.1). Кожен файл відповідає одному сімейству жанру та містить від двох до шести піджанрів із 150 композиціями у кожному. Загальний обсяг датасету склав 5550 треків, розподілених по 37 піджанрах.

Формат текстового файлу має таку ієрархічну структуру:

```
==============================
POP FAMILY
mainstream_pop
1. The Weeknd - Blinding Lights
2. Dua Lipa - Don't Start Now
...
150. Charlie Puth - We Don't Talk Anymore
k_pop
1. BTS - Dynamite
...
```

Парсинг цієї структури виконується функцією *parse_dataset_file()* (див. додаток В), яка послідовно обробляє кожен рядок файлу та визначає його тип: рядок-розділювач (починається з «===»), назва роду музики (містить слово «FAMILY»), ідентифікатор піджанру (валідується відносно списку жанрів із *config.yaml*) або запис треку (відповідає регулярному виразу формату «Номер. Автор – Назва»). Результатом парсингу є список словників, де кожен елемент містить поля *name, author, source_genre* та *source_family*.

Для наповнення бази даних було розроблено скрипт, який реалізує повний автоматизований конвеєр обробки музичних композицій. Кожен трек з текстового датасету проходить десятикрокову послідовність обробки, яку схематично зображено на рис. 3.10.

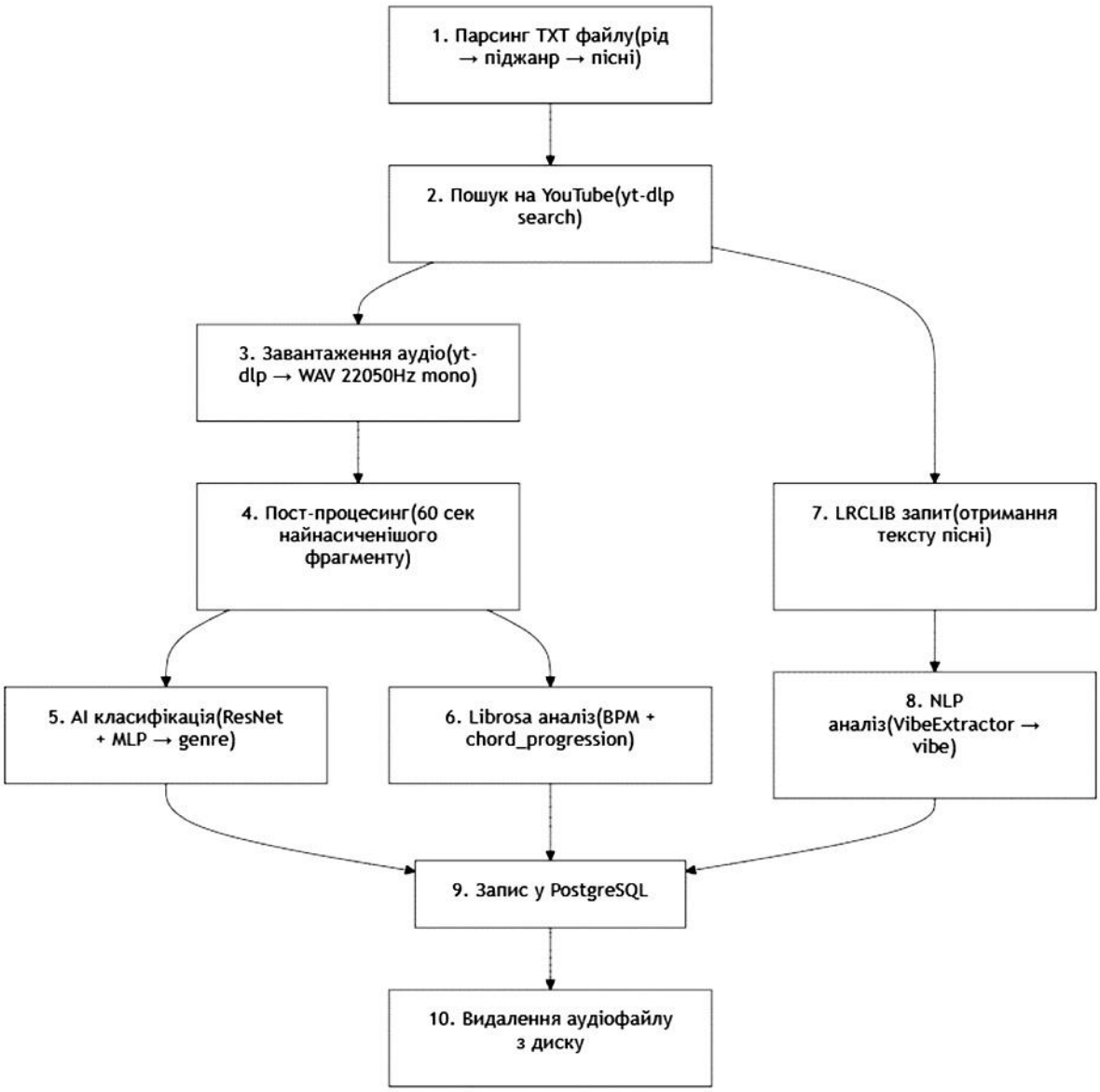

Рисунок 3.13 – Блок-схема конвеєра обробки треку при наповненні БД рекомендацій

Крок 1. Парсинг вхідних даних. Скрипт зчитує вибраний TXT-файл та формує ієрархічний список треків за структурою «рід музики, піджанр, пісні». Підтримується часткова обробка через аргументи командного рядка: можна обробити один конкретний файл, обмежити кількість треків або пропустити задану кількість початкових записів, що є корисним для відновлення після збою. Перед початком обробки кожного треку система виконує перевірку його наявності в базі

даних за комбінацією назви, автора та піджанру, а також за унікальним ідентифікатором YouTube. Якщо трек вже присутній у базі, обробка пропускається, що дозволяє безпечно перезапускати скрипт без ризику створення дублікатів.

Крок 2. Пошук на YouTube. Для кожного треку формується пошуковий запит у вигляді «Автор – Назва» та виконується пошук відповідного аудіо на платформі YouTube за допомогою утиліти *yt-dlp* з обмеженням результатів одним відео. Результатом пошуку є посилання на відео, унікальний ідентифікатор та метадані.

Крок 3. Завантаження аудіо. Знайдене аудіо завантажується та конвертується у формат WAV із частотою дискретизації 22050 Гц та зведенням у моно. У разі невдачі система виконує повторну спробу з модифікованим запитом, додаючи суфікс «official audio» для підвищення релевантності результатів пошуку.

Крок 4. Вирізання найнасиченішого фрагменту. Оскільки повна тривалість композиції може сягати кількох хвилин, а аналіз потребує стандартизованого вхідного сигналу, система автоматично знаходить та вирізає 60-секундний фрагмент із найвищою музичною насиченістю. Алгоритм реалізовано у функції *extract_dense_fragment()* (див. додаток Г) і працює наступним чином:

- спочатку виконується розділення звуку на мелодичну та ритмічну частини;
- обчислюється RMS-енергія гармонічної складової з кроком в одну секунду;
- вікно шириною 60 секунд послідовно зсувається по всьому сигналу і обирається та ділянка, де сумарна енергія є найвищою
- відповідний фрагмент вирізається з оригінального аудіосигналу.

Крок 5. AI-класифікація жанру. AI-класифікація жанру. Вирізаний фрагмент подається на вхід ансамблю нейронних мереж (див. підрозділ 3.4). ResNet аналізує мел-спектрограму, MLP обробляє числові акустичні ознаки, після чого ймовірності обох моделей усереднюються. Жанр із найвищою середньою ймовірністю записується у відповідне поле таблиці рекомендацій.

Крок 6. Аналіз гармонічної структури. Функція extract_bpm_and_chords() (див. додаток Д) виконує два ключових аналізи і видобуває інформацію, на якій ґрунтується алгоритм рекомендацій.

Визначення BPM здійснюється шляхом аналізу ритмічної структури аудіосигналу. Алгоритм відстежує моменти появи нових звукових подій у часі, знаходить їх періодичність та обчислює темп у ударах на хвилину.

Визначення тональності та ладу базується на алгоритмі Крумханзл–Кесслера. Обчислюється хроматограма шляхом розподілу енергії аудіосигналу за 12 класами висоти тону. На її основі формується середній хроматичний вектор, що узагальнює тональний склад усієї композиції. Цей вектор порівнюється з еталонними профілями мажорного та мінорного ладу за допомогою коефіцієнта кореляції Пірсона — числової міри подібності між двома наборами значень від -1 до 1. Порівняння виконується для кожної з 24 можливих тональностей (12 тонік у двох ладах). Тональність і лад із найвищим значенням кореляції обираються як результат. Додатково визначаються три домінуючі ноти.

Крок 7. Отримання тексту пісні. Текст композиції завантажується через відкритий API сервісу LRCLIB. Система виконує два послідовних запити: спочатку пряме співпадіння за полями *artist_name* та *track_name*, а у разі невдачі — повнотекстовий пошук. Отриманий текст зберігається у полі *text* таблиці.

Крок 8. NLP-аналіз емоційного забарвлення. Якщо текст пісні було успішно отримано, він передається класифікатору VibeExtractor (підрозділ 3.5), який визначає ймовірність належності тексту до кожної з 15 визначених емоційних категорій. Категорія з найвищою ймовірністю записується у відповідне поле бази даних.

Крок 10. Запис у базу даних та очищення. Зібрані дані формуються в об'єкт ORM-моделі RecommendationTrack та зберігаються у базі PostgreSQL через транзакцію SQLAlchemy. Після успішного запису тимчасовий аудіофайл видаляється з файлової системи, що забезпечує економію дискового простору при масовій обробці.

У результаті виконання повного циклу наповнення було оброблено дев'ять файлів датасету, що охоплюють дев'ять родів музики та 37 піджанрів. Загальна кількість проіндексованих треків у базі даних склала 5410 записів. Розподіл треків за родами музики наведено у табл. 3.2.

Таблиця 3.2 – Розподіл треків у базі даних за родами музики

Рід музики (Family)	**Кількість піджанрів**	**Початкова кількість треків**	**Вдало перенесена в БД кількість треків**
Rock Family	7	1050	1042
Pop Family	4	600	593
Hip-Hop Family	5	750	725
Electronic Family	6	900	879
Jazz Family	3	450	421
R&B Family	2	300	290
Classical Family	3	450	448
Asian Family	3	450	429
Other	4	460	583
Разом	**36**	**5550**	**5410**

Для кожного запису у базі зберігається повний набір характеристик: AI-класифікований жанр, гармонічна структура (тональність, лад, три домінуючі ноти), темп у BPM, емоційний настрій (vibe), текст пісні та посилання на джерело. Приклад вмісту бази даних із кількома записами проілюстровано на рис. 3.11.

	name	aut…	genre	chord_pro…		vi…	text	url	you…	source…	s.	crea…	language
1	4665	Blitzkrieg Bop	Ramones	punk_hardcore	D#_major.A.E.D	123	aggressive	It's good to be back…	https://ww…	268C3N2dDYk	punk_hardcore	ROCK FA…	2026-05-17…	en
2	4666	I Wanna Be Sed…	Ramones	punk_hardcore	F#_major.F#.B.C#	161.5	chill	20, 20, 24 hours to …	https://ww…	bm51ihfi1p4	punk_hardcore	ROCK FA…	2026-05-17…	en
3	4667	Sheena Is A Pu…	Ramones	punk_hardcore	C_major.C.G.C#	89.1	chill	Well the kids are al…	https://ww…	yCW7Aw8ugOI	punk_hardcore	ROCK FA…	2026-05-17…	en
4	4668	The KKK Took M…	Ramones	punk_hardcore	D_major.A#.D.A	152	nostalgic	She went away for th…	https://ww…	RYX5Sq0105Y	punk_hardcore	ROCK FA…	2026-05-17…	en
5	4669	Pet Sematary	Ramones	post_punk	G_major.C.D.A#	73.8	motivating	Under the arc of a w…	https://ww…	HJWFsZ_YUc4	punk_hardcore	ROCK FA…	2026-05-17…	en

Рисунок 3. 14 – Скріншот таблиці *recommendation_tracks*

Наведені записи підтверджують коректність роботи конвеєра наповнення, усі ключові поля заповнені відповідно до визначеної структури бази даних.

3.7 Програмна реалізація алгоритму мультисигнального ранжування

Після того як кожна композиція бази даних отримала свій мультимодальний профіль (жанр, акордова прогресія, темп, настрій лірики), наступним кроком є реалізація механізму пошуку найбільш подібних треків.

Архітектура in-memory індексу HarmonicIndex. Послідовне сканування бази даних засобами ORM для кожного запиту виявилось повільним: при обсязі каталогу понад 5 000 треків час відповіді перевищував 3 секунди. Для вирішення цієї проблеми реалізовано клас HarmonicIndex, який при старті серверу одноразово завантажує всі записи таблиці *recommendation_tracks* у оперативну пам'ять та формує NumPy-структури для кожного сигналу. Це дозволяє виконувати векторизовані обчислення подібності за час, що не перевищує 15 мілісекунд навіть для повного каталогу.

Індекс зберігає для кожного треку такі структури даних:

- матрицю хроматичних векторів розмірністю N на 12, де N — кількість треків у каталозі. Завдяки цьому косинусна подібність між поточним треком та всією базою обчислюється однією матричною операцією завдяки оптимізаціям NumPy.
- одновимірні масиви кореневих тональностей, ладу (мажор/мінор) та BPM;
- цілочислові ідентифікатори жанру, настрою, виконавця та мови. Рядкові значення попередньо конвертуються в числові ID, що дозволяє виконувати порівняння за допомогою векторних операцій;
- статичну матрицю відстаней розміром 12 на 12, що будується один раз при запуску системи. Під час ранжування система зчитує готове значення з матриці за номерами двох тональностей, не виконуючи обчислень повторно.

Побудова хроматичного вектора з акордової прогресії здійснюється методом *_parse_chord*. Рядок формату «A_minor.A.C.E» розбивається на компоненти: перший токен визначає кореневу ноту з вагою 1,0 та лад, решта нот отримують вагу 0,5. Отриманий 12-вимірний вектор нормалізується діленням на суму його компонент.

Система вагових коефіцієнтів. Алгоритм обчислює зважену суму восьми незалежних сигналів подібності між треком-запитом та кожним кандидатом у каталозі. До фінального скору додається невеликий гауссівський шум, що запобігає ефекту фільтрувальної бульбашки, коли користувач щоразу отримує ідентичний набір рекомендацій.

У табл. 3.3 наведено перелік сигналів, відсортованих за спаданням ваги, разом із описом їх обчислення.

Таблиця 3.3 – Сигнали подібності та їхні вагові коефіцієнти

№	Сигнал	Вага	Спосіб обчислення	Діапазон
1	Жанрова відповідність	1	Точний збіг = 1,0; та ж сім'я = 0,5; споріднена сім'я = 0,2–0,35; інша = 0	[0; 1]
2	Гармонічна відповідність	0,75	Косинусна подібність хроматичних векторів	[0; 1]
3	Тональна сумісність	0,63	Лінійний спад залежно від відстані за колом квінт: 0 кроків = 1,0; 6 кроків = 0	[0; 1]
4	Емоційна відповідність	0,38	Точний збіг vibe: збіг = 1,0; інший = 0	(0; 1)
5	Збіг виконавця	0,25	Точний збіг автора: збіг = 1,0; інший = 0	(0; 1)
6	Ладова сумісність	0,25	Однаковий лад = 1,0; різний = 0,3	(0,3; 1)

Кінець таблиці 3.3

№	Сигнал	Вага	Спосіб обчислення	Діапазон
7	Близькість темпу	0,13	Лінійний спад: максимум при однаковому BPM, нуль при різниці понад 40 BPM	[0; 1]
8	Мовна відповідність	0,1	Точний збіг мови лірики: збіг = 1,0; інша = 0	(0; 1)

Жанрова відповідність отримала найвищу вагу, оскільки жанр є первинним контекстним фільтром. Водночас система не обмежується лише точним збігом: усі 37 піджанрів об'єднано у 9 родин (*Rock, Hiphop, Pop, Classical, Jazz, Rnb, Electronic, Asian, Other*), а між спорідненими родинами визначено коефіцієнти взаємної спорідненості. Наприклад, пара *Pop* і *Rnb* отримує коефіцієнт 0,35 через значний стилістичний перетин, тоді як пара *Rock* і *Electronic* отримує лише 0,2.

Алгоритм диверсифікованого відбору. Після обчислення значень подібності для всіх треків каталогу система не обирає просто *N* найкращих кандидатів. Натомість функція *_diversified_select* застосовує двопрохідний послідовний алгоритм відбору з обмеженнями різноманітності.

На першому етапі формується розширений пул із 80 кандидатів із найвищими значеннями подібності. На другому етапі з цього пулу послідовно обираються треки із дотриманням двох обмежень:

– від одного виконавця у фінальному списку може бути не більше 5 треків) що запобігає домінуванню одного артиста;

– частка треків того самого жанру, що й трек-запит, обмежена на рівні 70%, що залишає щонайменше 30% місць для композицій із споріднених жанрів;

– якщо після першого проходу залишаються незаповнені позиції, алгоритм виконує другий прохід, послаблюючи обмеження на жанрову різноманітність, та заповнює місця з раніше відкладених кандидатів. Система також ігнорує короткостроковий кеш при кожному новому тригері пісні, що гарантує

унікальність списку для кожного відтворення. Фінальне значення подібності нормалізується до діапазону від 0 до 1, що дозволяє інтерфейсу відображати відсоток подібності кожного треку у зрозумілій для користувача шкалі.

Асинхронна інтеграція у життєвий цикл плеєра. Щоб уникнути блокування інтерфейсу під час розрахунку рекомендацій, система використовує асинхронний сервіс RecommendationService. Процес інтегровано у життєвий цикл плеєра у три етапи.

Паралельний запуск. Коли користувач вмикає пісню, паралельно із запитом на потокове аудіо фронтенд надсилає асинхронний POST-запит на запуск генерації рекомендацій.

Фонове виконання. На бекенді створюється фонова задача, яка делегує інтенсивні векторні обчислення в окремий пул потоків. Це гарантує, що основний цикл обробки запитів сервера не блокується.

Опитування та кешування. Під час програвання пісні клієнт періодично надсилає запити на сервер щодо стану рекомендацій. Після завершення обчислень результати зберігаються у внутрішньому кеші та повертаються на фронтенд для відображення.

Висновки до розділу 3

У ході програмної реалізації системи сформовано унікальний датасет із 16 650 музичних композицій, розподілених по 37 піджанрах у межах 9 родин, із застосуванням офлайн-аугментації (Pitch Shift та Time Stretch). Розроблено модуль попередньої обробки аудіо, що реалізує інтелектуальне кадрування та формує два типи вхідних представлень: мел-спектрограми для ResNet і числові вектори акустичних ознак для MLP. Ансамбль цих архітектур із зваженим агрегуванням (70/30) досяг точності класифікації 70,9% на незалежній тестовій вибірці. Паралельно інтегровано багаторівневий NLP-конвеєр, що поєднує пошук тексту через LRCLIB API, машинний переклад та Zero-Shot класифікацію емоцій на базі DistilBERT із резервним механізмом транскрипції через OpenAI Whisper, що

дозволяє визначати емоційне забарвлення як вокальних, так і інструментальних композицій.

Спроєктовано реляційну базу даних PostgreSQL із 5410 проіндексованими записами, кожен з яких містить AI-класифікований жанр, гармонічну структуру, темп у BPM та емоційний настрій, отриманий через автоматизований десятикроковий конвеєр наповнення. На основі цих даних реалізовано алгоритм мультисигнального ранжування з in-memory індексом HarmonicIndex, що обчислює зважену суму восьми сигналів подібності за час не більше 15 мілісекунд для каталогу понад 5000 треків. Двопрохідний алгоритм диверсифікованого відбору з обмеженнями на представленість артиста (не більше 5 треків) та жанрову однорідність (не більше 70%), доповнений гауссівським шумом, гарантує різноманітність рекомендацій при кожному відтворенні. Асинхронна інтеграція механізму через asyncio та run_in_executor унеможливлює блокування головного циклу подій FastAPI, забезпечуючи плавний користувацький досвід без затримок інтерфейсу.

4 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ РЕКОМЕНДАЦІЙ ТА ТЕСТУВАННЯ

4.1 Результати навчання моделей та експериментальна оцінка точності класифікації

Наявність двох жанрових полів у кожному записі, *genre* (результат AI-класифікації) та *source_genre* (оригінальна мітка з датасету), створює можливість для валідації якості роботи ансамблю нейронних мереж безпосередньо на реальних даних, які слухатимуть майбутніми рекомендаціями. Для цього було розроблено скрипт, який порівнює передбачений жанр із оригінальною міткою для кожного з 5410 треків та обчислює точність класифікації як загальну, так і по кожному піджанру окремо

Результати валідації наведено на табл. 4.1, 4.2. Загальна точність ансамблю на всій базі даних становить 81.0 %, що підтверджує працездатність моделей при обробці реальних аудіозаписів і відповідає тестуванню при навчанні, завантажених з YouTube, на відміну від контрольованого середовища навчальної вибірки.

Таблиця 4.1 – Точність за жанрами

Жанр	Правильно	Всього	Точність
folk_acoustic	148	148	100.0%
drill	136	144	94.4%
chamber_piano	141	150	94.0%
classic_jazz	138	148	93.2%
modern_metal	140	146	95.9%
opera	141	148	95.3%
reggaeton_latin	130	141	92.2%
orchestral	139	150	92.7%
modern_rnb	132	143	92.3%
hyperpop	134	146	91.8%
house_techno	135	149	90.6%
city_pop	130	144	90.3%

Кінець таблиці 4.1

Жанр	**Правильно**	**Всього**	**Точність**
lofi_chillhop	130	145	89.7%
trap	130	148	87.8%
midwestern_emo	130	149	87.2%
oldschool_hiphop	129	148	87.2%
punk_hardcore	128	148	86.5%
retrowave	127	147	86.4%
phonk	119	140	85.0%
smooth_jazz	124	147	84.4%
darksynth	122	145	84.1%
post_punk	124	149	83.2%
cloud_rap	118	145	81.4%
dubstep_bass	118	147	80.3%
j_rock	119	149	79.9%
ambient_idm	115	147	78.2%
jazz_fusion	97	126	77.0%
mainstream_pop	110	149	73.8%
anime_ost	100	136	73.5%
grunge	106	150	70.7%
drum_and_bass	101	144	70.1%
indie_pop	96	148	64.9%
classic_rnb	89	147	60.5%
k_pop	83	150	55.3%
prog_rock	80	150	53.3%
classic_rock	74	150	49.3%
alt_rock	72	149	48.3%
OVERALL	**4385**	**5410**	**81.1%**

Детальний аналіз показаний у табл. 4.1 дозволяє виділити три групи жанрів за рівнем успішності класифікації: жанри з високою точністю (86–100%); жанри із середньою точністю (65–85%); жанри з низькою точністю (менше 65%).

Жанри з високою точністю (86–100%). Найкращі результати ансамбль нейромереж продемонстрував на жанрах із яскраво вираженою унікальною акустичною сигнатурою або специфічним набором інструментів. Наприклад, *folk_acoustic* (100%), *chamber_piano* (94%) та *opera* (95%) характеризуються чистотою звучання та відсутністю електронної дисторсії, що формує чіткі патерни на мел-спектрограмах. Жанр *drill* (94%) безпомилково розпізнається завдяки характерним ковзаючим басам та специфічному синкопованому ритму хай-хетів. *Modern_metal* (96%) розпізнається через високу щільність звуку та агресивну дисторсію гітар у нижньому регістрі.

Жанри із середньою точністю (65–85%). Електронна та урбан-музика, зокрема *phonk, trap, cloud_rap і dubstep_bass*, мають хороші показники, оскільки здебільшого створена у цифрових аудіостанціях із чітким темпом. Зниження точності пов'язане з тим, що ці піджанри часто запозичують елементи один в одного.

Жанри з низькою точністю (менше 65%). Найбільші труднощі система мала з рок-музикою, а саме *classic_rock, alt_rock, prog_rock*, та азійською поп-музикою *k_pop*. Показники на рівні 48–61% свідчать про те, що моделям важко знайти чітку акустичну межу між цими категоріями. Причина полягає у високому ступені інструментального та структурного перекриття: класичний, альтернативний та прогресивний рок використовують ідентичний набір інструментів, гітара, бас, ударні, вокал, і часто збігаються за частотними характеристиками спектрограми.

Для глибшого розуміння природи помилок ансамблю було сформовано матрицю найчастіших хибних спрацьовувань, візуалізацію якої наведено на рис. 4.1, а результати зведено у табл. 4.2.

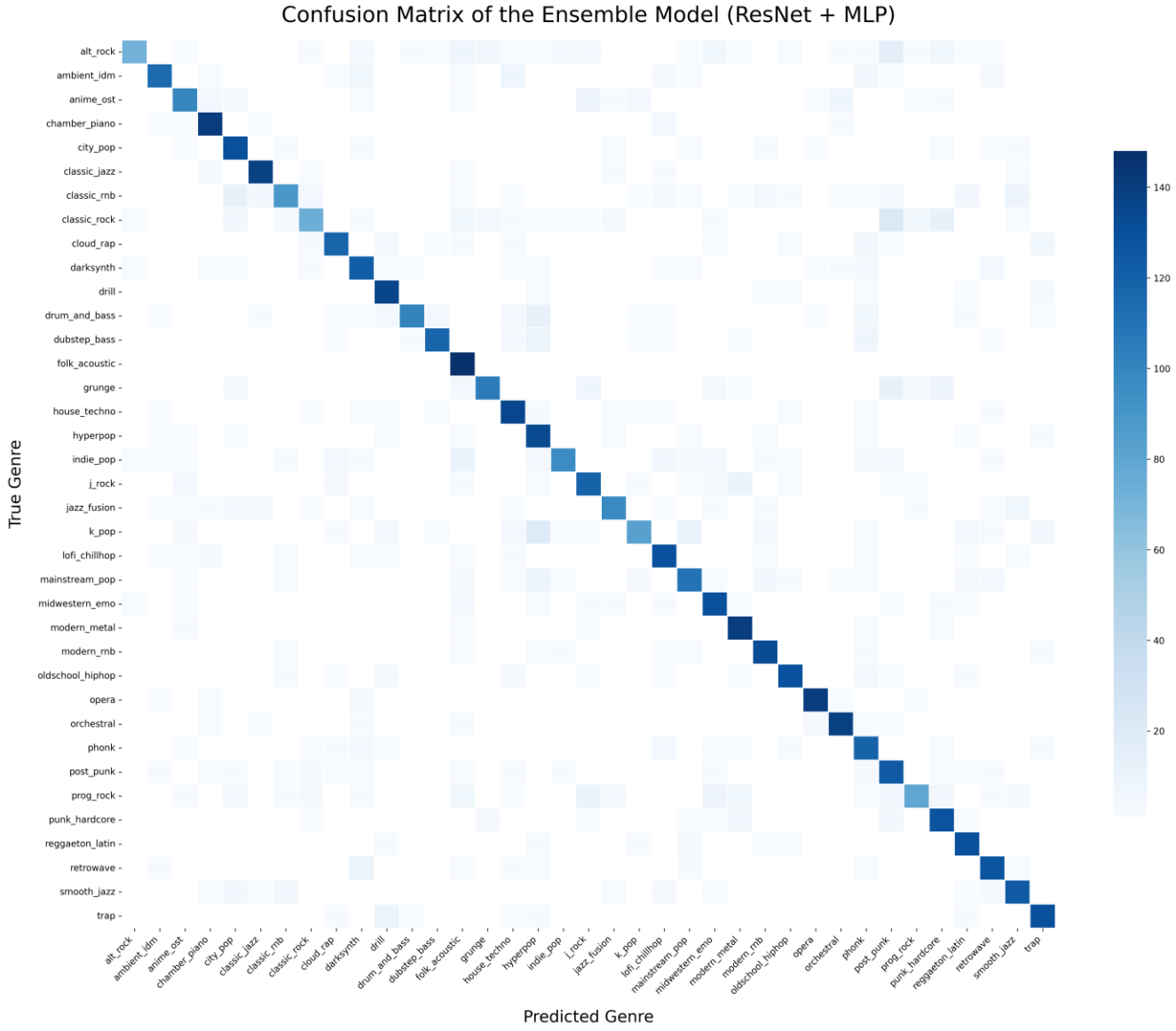

Рисунок 4.1 – Матриця хибних справцьовувань

Таблиця 4.2 – Найчастіші помилки класифікації

Справжній жанр	**Передбачений жанр**	**Кількість**
classic_rock	post_punk	22
k_pop	hyperpop	20
alt_rock	post_punk	16
classic_rock	punk_hardcore	13
grunge	post_punk	12
drum_and_bass	hyperpop	12

Кінець таблиці 4.2

Справжній жанр	**Передбачений жанр**	**Кількість**
classic_rnb	city_pop	12
prog_rock	j_rock	11
j_rock	modern_metal	11
k_pop	mainstream_pop	11

Було проведено аналіз природи хибних класифікацій. Дані з табл. 4.2 ілюструють феномен внутрішньородинної плутанини, що підтверджує здатність нейромережі правильно розуміти базовий характер музики, навіть коли вона помиляється з точним піджанром.

Можна зробити наступні висновки, що майже всі помилки класифікації *classic_rock, alt_rock* та *grunge* зводяться до того, що система відносить їх до *post_punk* або *punk_hardcore*. Це означає, що ансамбль чітко розпізнає трек як рок-композицію (тобто глобальна класифікація працює праивльно), але через розмите поняття піджанрів у рок-музиці (де межа між альтернативою та гранжем часто є суб'єктивною навіть для людини) мережа обирає суміжний клас.

Жанр *k_pop* найчастіше плутають із *hyperpop* (20 випадків) та *mainstream_pop* (11 випадків). Це пояснюється тим, що ResNet та MLP аналізують виключно акустичні властивості сигналу, тобто текст пісні на цьому етапі не враховується. Тому з суто акустичної точки зору *k_pop* дійсно є яскравою, насиченою синтезаторами поп-музикою, що робить його спектральний профіль близьким до *hyperpop*.

Схожа ситуація із *drum_and_bass*, який періодично відносять до *hyperpop*. Обидва жанри характеризуються екстремально високим темпом (150-170+ BPM) та насиченим синтетичним звучанням, що призводить до схожих векторів ознак у просторі MLP-моделі.

Показовим є випадок із *prog_rock*, який нейромережа класифікує як *j_rock*. Японський рок історично запозичив багато складних гармонічних і ритмічних

структур саме з прогресивного року, тому їхня спектральна подібність є обґрунтованою.

Досягнута точність 81,1% на 37 піджанрах підтверджує, що нейромережа засвоїла правильні семантичні зв'язки між музичними стилями: помилки відбуваються переважно між стилістично близькими піджанрами. Для рекомендаційної системи це не є недоліком, оскільки можливі неточності жанрової класифікації нівелюються на етапі мультисигнального ранжування, яке додатково враховує гармонічну структуру та емоційне забарвлення тексту.

4.2 Демонстрація роботи розробленої системи

Після завершення інтеграції бекенд-сервісів та розробки клієнтської частини, було отримано повнофункціональний веб-додаток, який інкапсулює складність нейромережевих обчислень та надає користувачу зручний інструментарій для взаємодії з музичним контентом. Клієнтська архітектура побудована за принципом SPA (Single Page Application) з використанням збірки Vite, що забезпечує миттєву реакцію інтерфейсу на дії користувача.

Перший екран, з яким стикається новий користувач це сторінка авторизації (Authentication Page) (див. рис. 4.2). Система підтримує два методи автентифікації: класичну реєстрацію через електронну пошту з підтвердженням 6-значним кодом верифікації та швидкий вхід через протокол OAuth 2.0 з використанням Google Identity Services. Форма реєстрації включає валідацію полів (ім'я, email, пароль із підтвердженням) та кнопку для перемикання видимості пароля. Після реєстрації система автоматично надсилає верифікаційний код на вказану адресу, а інтерфейс переходить до екрану введення коду з можливістю повторного надсилання.

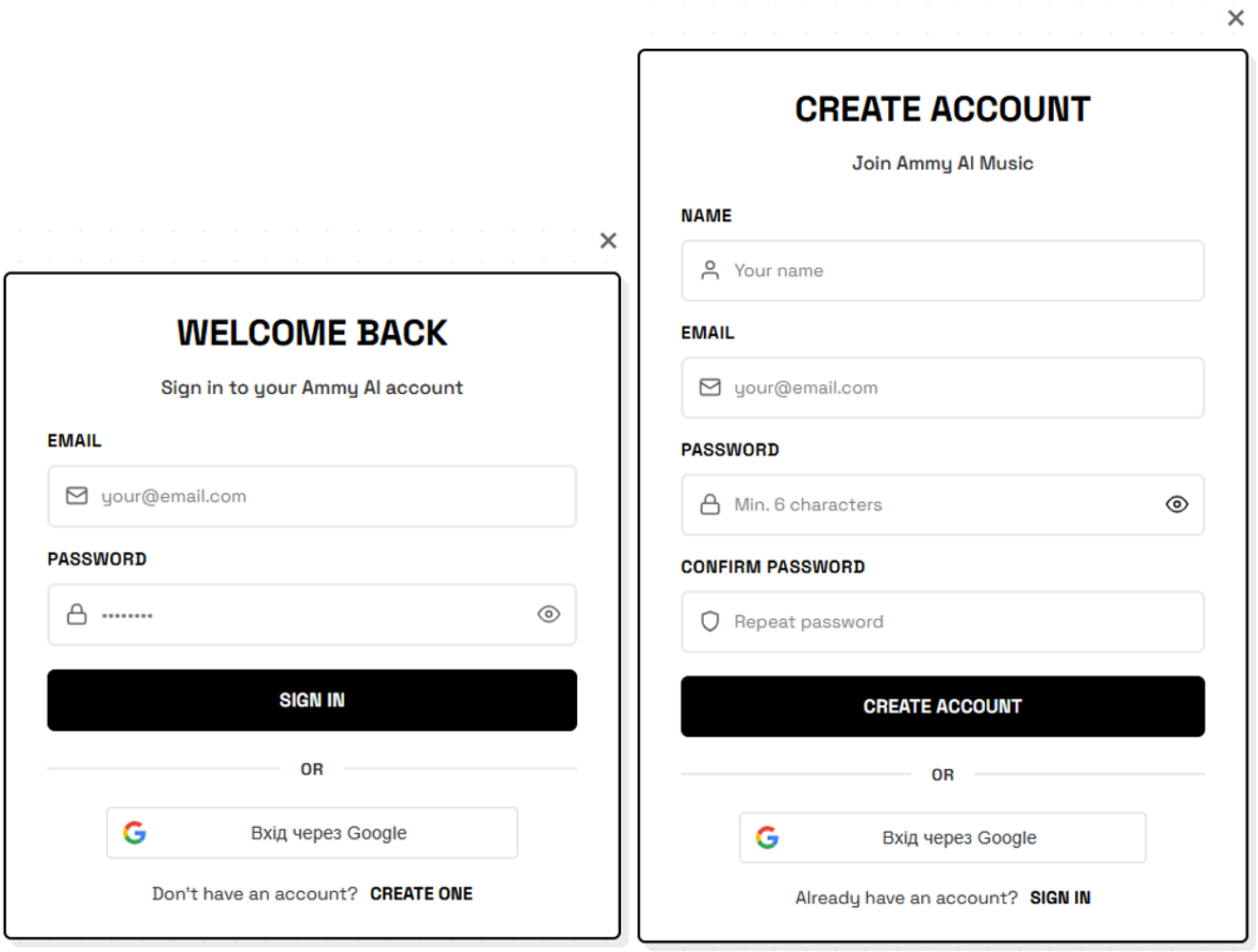

Рисунок 4.2 – Сторінка авторизації (Login / Register) із підтримкою Google OAuth

Після успішної авторизації користувач потрапляє на головний екран системи (див. рис. 4.3), який розділений на кілька функціональних зон: бічна панель навігації Sidebar для швидкого доступу до розділів Home, Analysis, бібліотеки плейлистів та історії аналізу; верхня панель Header із рядком пошуку, кнопкою Import для завантаження локальних аудіофайлів та елементами управління обліковим записом.

Основна робоча область головного екрана містить три секції контенту. Перша секція Recently Played відображає дев'ять останніх прослуханих треків з інформацією про назву, автора, жанр та мову. Друга секція Your Playlists містить створені користувачем плейлисти та збережені пісні Liked Songs. Третя секція

Quick Picks пропонує персоналізовані рекомендації на основі вподобань користувача.

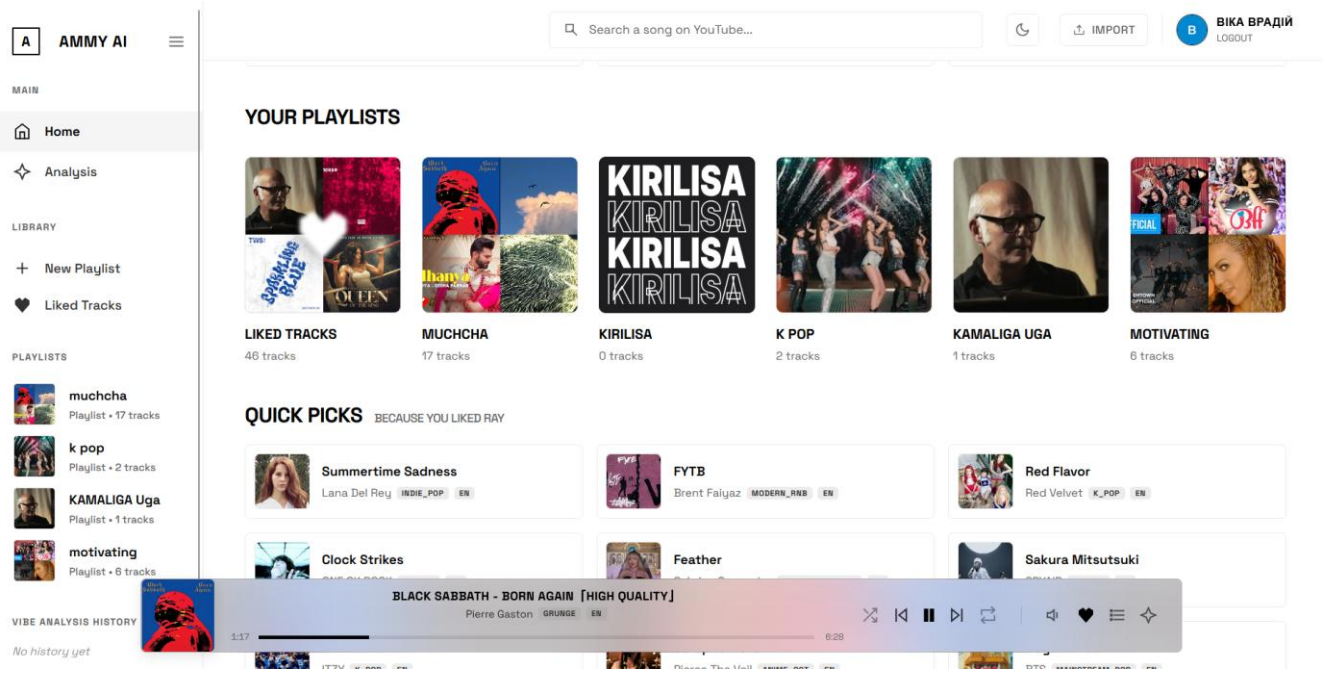

Рисунок 4.3 – Головний екран веб-додатка (Home View)

Верхній рядок пошуку інтегрований з YouTube Data API та забезпечує автодоповнення з випадаючим списком підказок (див. рис. 4.4). При натисканні клавіші Enter система переходить до повноекранної сторінки результатів пошуку (Search Results), яка відображає до 40 треків у вигляді нумерованого списку із мініатюрами, метаданими (автор, жанр, мова, тривалість).

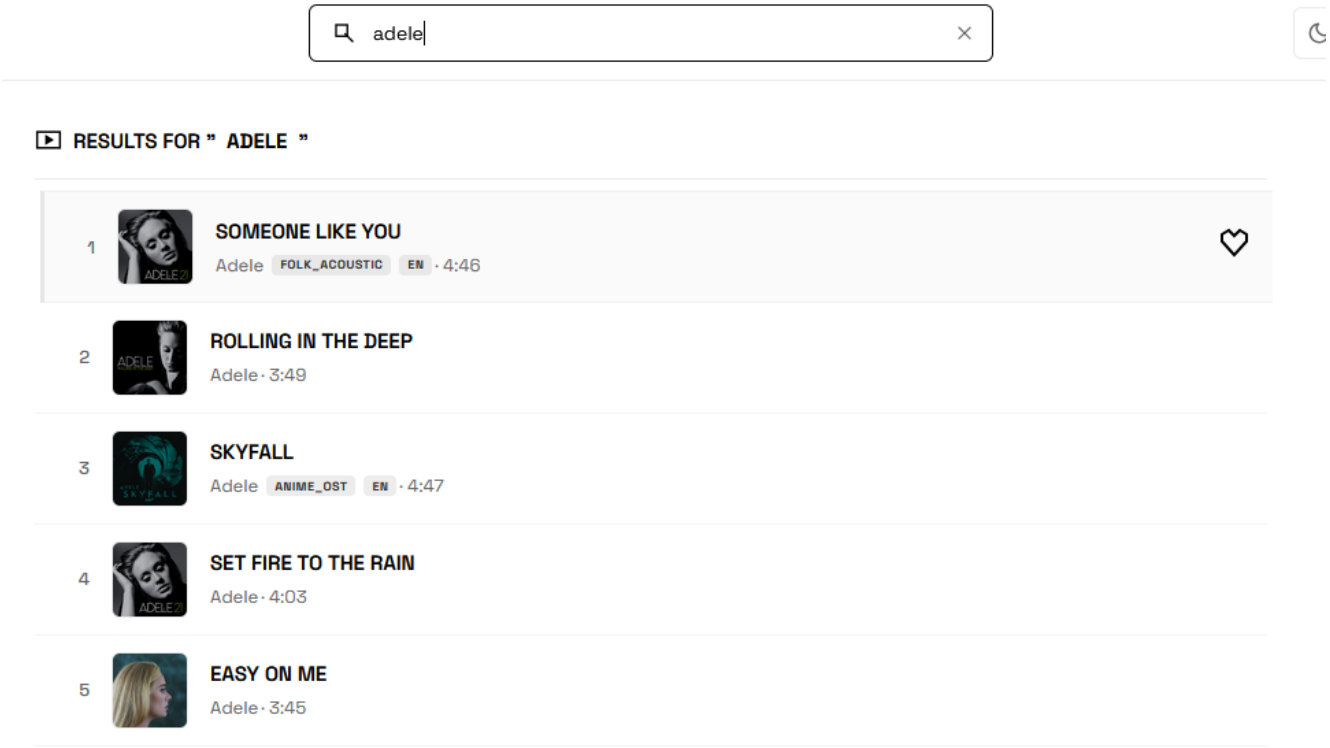

Рисунок 4.4 – Сторінка результатів пошуку із нумерованим списком треків

Одним із ключових вікон є панель аналізу (Analysis View) (див. рис. 4.5). Коли користувач завантажує власний локальний аудіофайл через діалогове вікно імпорту або обирає трек з пошуку для детального розбору, система відображає результати аналізу у вигляді віджетів:

– жанровий профіль відображається у вигляді прогрес-барів з імовірностями;

– емоційне забарвлення виводиться у вигляді тегів;

– гармонічна структура відображається на бічній панелі;

– текст пісні за наявності.

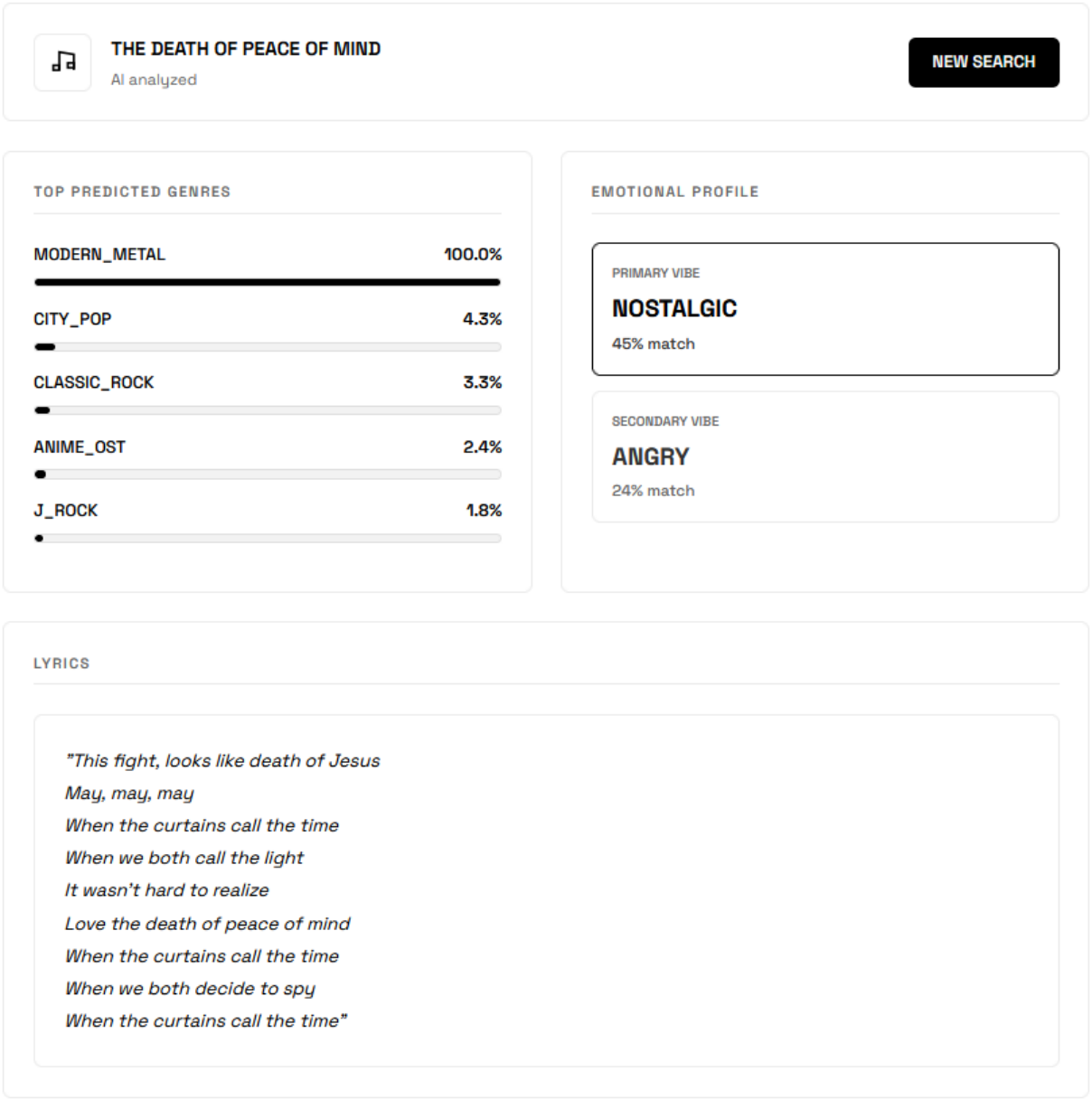

Рисунок 4.5 – Інтерфейс панелі аналізу аудіотреку (Analysis View)

Процес генерування рекомендацій інтегрований у життєвий цикл вбудованого аудіоплеєра (Player) (див. рис. 4.6).

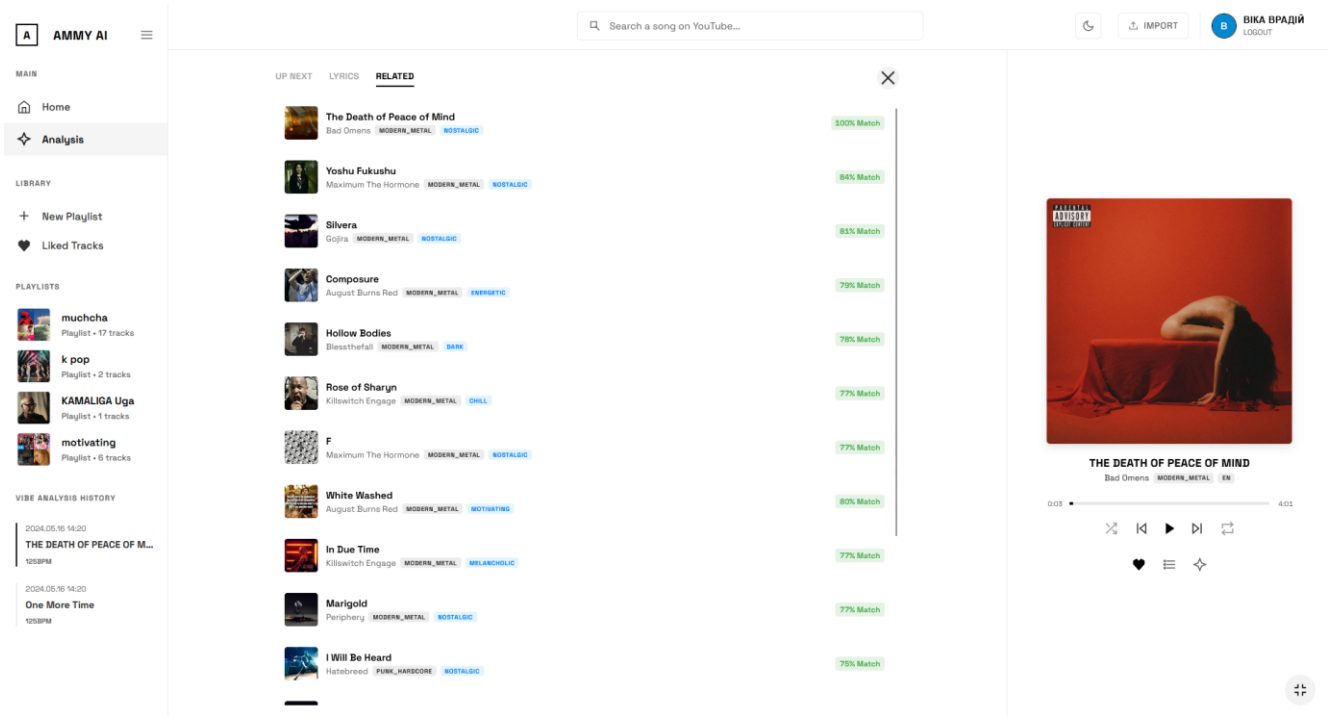

Рисунок 4.6 – Інтерфейс вбудованого аудіоплеєра та системи рекомендацій (Related)

Як тільки користувач натискає кнопку відтворення, клієнтський додаток не лише ініціює стрімінг аудіо, але й відправляє асинхронний запит до бекенду на пошук схожих композицій. Завдяки реалізованому патерну Fire-and-Forget, інтерфейс не блокується під час інтенсивних обчислень. Після успішного відпрацювання in-memory індексу HarmonicIndex, плеєр автоматично заповнює чергу відтворення Up Next релевантними треками, відібраними з дотриманням правил жанрової та авторської диверсифікації.

Детальний перегляд плейлиста реалізований у вигляді двопанельного макету (див. рис. 4.7). Ліва панель містить прокручуваний список треків з нумерацією, мініатюрами, метаданими та кнопкою видалення при наведенні курсору, а також кнопкою сортування.

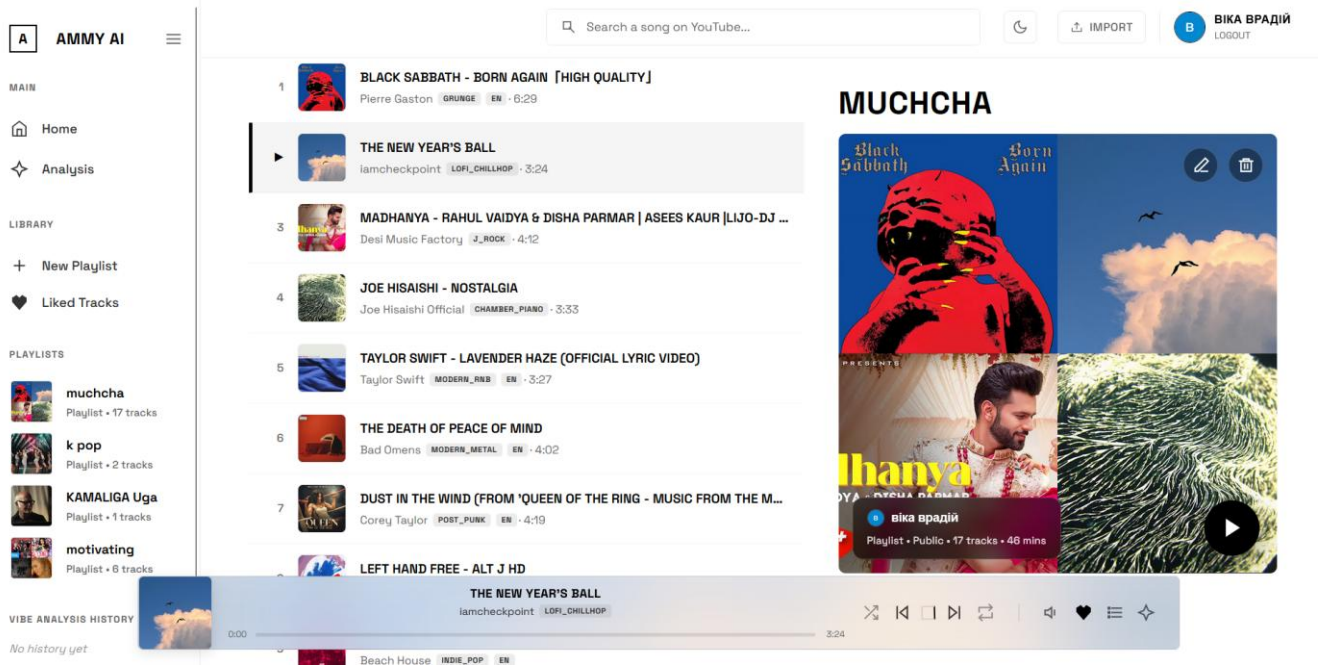

Рисунок 4.7 – Інтерфейс перегляду плейлиста (Playlist View)

Права панель включає великий заголовок плейлиста, динамічну обкладинку-колаж, інформаційний бейдж із аватаром користувача, кількістю треків і загальною тривалістю, а також кнопки редагування (перейменування, видалення).

4.3 Тестування інтерфейсу

Тестування системи проводилось методом «чорного ящика» (Black-box testing) — без доступу до внутрішнього коду, виключно через зовнішні інтерфейси: HTTP API та браузерний клієнт. Перевірці підлягали три аспекти: коректність обробки запитів на рівні API, стабільність клієнтського інтерфейсу та швидкодія рекомендаційної підсистеми.

Перевіркою було охоплено три ключові модулі системи. Для ендпоінту */analyze* тестувалась валідація вхідних файлів: при завантаженні файлів з непідтримуваними розширеннями (.txt, .mp4) сервер переривав обробку та повертав статус 400 Bad Request з відповідним повідомленням. Коректні файли форматів .mp3 та .wav проходили повний аналіз та формували JSON-профіль треку без помилок.

Для ендпоінту */auth* було перевірено механізми захисту облікових записів. Спроба реєстрації з вже існуючою електронною адресою блокувалась на рівні сервера. Також підтверджено коректну роботу двофакторної верифікації: згенерований код автоматично анулюється через 10 хвилин після відправки.

Для ендпоінту */analyze_url* було протестовано граничні сценарії взаємодії з YouTube. Встановлений ліміт завантаження у 5 хвилин відпрацьовував коректно: спроби обробити надто довгі відео автоматично перериваються із поверненням статусу «504 Gateway Timeout», що захищає сервер від вичерпання ресурсів.

Оскільки клієнтська частина реалізована як Single Page Application, основну увагу було приділено стабільності глобального стану та реакції інтерфейсу на нестандартні вхідні дані. Перевірено безперервність відтворення аудіо під час навігації: переходи між розділами додатку відбуваються без перезавантаження сторінки завдяки клієнтській маршрутизації, тому аудіоплеєр зберігає стан і поточний трек не переривається.

Окремо було протестовано стійкість інтерфейсу до нестандартних метаданих. Композиції з надмірно довгими назвами, відсутніми текстами або відсутніми обкладинками не порушують макет сторінки: текстові блоки коректно усікаються, замість відсутніх зображень відображаються резервні заглушки. Також перевірено алгоритм динамічної зміни теми оформлення, що витягує домінуючі кольори з обкладинок альбомів: аналіз пікселів виконується без блокування головного потоку браузера, частота кадрів інтерфейсу залишається стабільною, кнопки плеєра реагують без затримок.

Також було виміряно час відгуку ендпоінту */recommendations/{youtube_id}* при повному навантаженні каталогу. На базі з 5 410 треків повний цикл обчислення зваженої подібності за вісьмома сигналами займає в середньому 15 мілісекунд.

Асинхронний механізм Fire-and-Forget підтвердив свою ефективність на практиці: головний Event Loop FastAPI не блокується під час матричних обчислень, оскільки вони виконуються у окремому пулі потоків. Сервер одночасно віддає

аудіопотік поточному користувачу та приймає нові HTTP-запити від інших клієнтів без затримок.

Висновки до розділу 4

Експериментальна оцінка ансамблю ResNet і MLP на 5410 реальних треках підтвердила загальну точність класифікації 81,1% для 37 піджанрів. Аналіз хибних спрацьовувань показав, що помилки відбуваються переважно між стилістично спорідненими жанрами, що є прийнятним для задач рекомендації та нівелюється на етапі мультисигнального ранжування.

Функціональне тестування методом чорного ящика підтвердило коректність роботи всіх ключових модулів: валідація вхідних файлів, захист облікових записів та обробка граничних сценаріїв взаємодії з YouTube працюють відповідно до очікувань. Інтерфейс стійкий до нестандартних метаданих, а час генерації рекомендацій для каталогу з 5410 треків не перевищує 15 мілісекунд завдяки in-memory індексу HarmonicIndex.

ВИСНОВКИ

У кваліфікаційній роботі вирішено актуальну науково-практичну задачу розробки інтелектуальної системи рекомендацій музичних композицій на основі контентно-гармонічного аналізу, що функціонує без використання поведінкових даних користувачів та забезпечує подолання проблеми «холодного старту».

За результатами виконання роботи отримано такі основні результати:

– проведено аналіз предметної сфери та класифікацію методів побудови рекомендаційних систем. Встановлено, що алгоритми колаборативної фільтрації мають два критичних обмеження: популяризаційне упередження та нездатність рекомендувати нові композиції через відсутність накопиченої статистики взаємодій. Порівняльний аналіз Spotify, YouTube Music та Apple Music підтвердив, що жодна з платформ не поєднує глибокий акустичний аналіз аудіосигналу із семантичним аналізом тексту пісень, що обґрунтовує доцільність обраного підходу;

– досліджено та обрано комплекс методів цифрової обробки аудіосигналів: мел-спектрограми як вхідні зображення для згорткових мереж, Chroma CQT та простір Tonnetz для гармонічного аналізу, MFCC для опису тембральних характеристик. Для кожного часового ряду обчислено шість статистичних показників, що формують вектор понад 150 числових ознак;

– розроблено та навчено ансамбль нейронних мереж ResNet-18 та MLP на власному датасеті обсягом 16 650 музичних композицій, розподілених за 37 піджанрами. Зважене об'єднання результатів двох моделей забезпечило точність класифікації 81,1% на незалежній тестовій вибірці з 5 410 реальних треків. Аналіз матриці хибних спрацьовувань підтвердив, що помилки відбуваються переважно між стилістично спорідненими піджанрами, що є прийнятним для задач рекомендації;

– реалізовано модуль семантичного аналізу тексту пісень із використанням моделі Whisper для автоматичної транскрипції вокалу та моделі DistilBERT для

Zero-Shot класифікації емоційного забарвлення за 14 категоріями. Підхід не потребує попереднього навчання на розмічених музичних даних та обробляє тексти довільною мовою після автоматичного перекладу;

– розроблено алгоритм мультисигнального ранжування рекомендацій на основі зваженої суми восьми незалежних сигналів подібності. Впроваджено механізм стохастичної диверсифікації для запобігання ефекту «бульбашки фільтрів». Реалізований in-memory індекс на основі NumPy-структур забезпечує час генерації рекомендацій 15 мілісекунд для каталогу з 5 410 треків;

– здійснено повну програмну реалізацію системи у вигляді клієнт-серверного веб-застосунку на базі React, FastAPI та PostgreSQL. Проведено функціональне тестування та тестування продуктивності методом «чорного ящика», які підтвердили коректність роботи всіх модулів та стабільність інтерфейсу.

Порівняння з комерційними аналогами підтверджує досягнення поставленої мети: розроблена система забезпечує повну стійкість до проблеми холодного старту, усуває популяризаційне упередження та забезпечує прозорість рекомендаційних рішень — характеристики, недосяжні в рамках колаборативної фільтрації, що застосовується у Spotify, YouTube Music та Apple Music.

Результати роботи можуть бути впроваджені як незалежний рекомендаційний сервіс для музичних платформ, орієнтованих на підтримку маловідомих виконавців. Перспективами подальшого розвитку є масштабування каталогу шляхом переходу на векторні бази даних, додавання контекстуальних рекомендацій залежно від часу доби та настрою користувача, а також дослідження застосування мультимодальних мовних моделей для глибшого аналізу музичного змісту.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Kowald D., Schedl M., Lex E. The Unfairness of Popularity Bias in Music Recommendation: A Reproducibility Study // Advances in Information Retrieval : Proceedings of the 42nd European Conference on IR Research (ECIR 2020). Cham : Springer, 2020. P. 1–5.

2. Wei Y., Wang X., Li Q. et al. Contrastive Learning for Cold-Start Recommendation // arXiv preprint arXiv:2107.05315. 2021.

3. Van den Oord A., Dieleman S., Schrauwen B. Deep content-based music recommendation // Advances in Neural Information Processing Systems (NIPS). 2013. P. 1–9.

4. Rumiantcev M. Ontology-Guided Multimodal Framework for Explainable Music Similarity and Recommendation // Big Data and Cognitive Computing. 2026. Vol. 10, no. 4. Art. 122.

5. Ganhör C., Moscati M., Hausberger A. et al. A Multimodal Single-Branch Embedding Network for Recommendation in Cold-Start and Missing Modality Scenarios // Proceedings of the 18th ACM Conference on Recommender Systems (RecSys 2024). 2024.

6. Akhil P. V., Joseph S. A survey of recommender system types and its classification // International Journal of Advanced Research in Computer Science. 2017. Vol. 8, no. 9. P. 486–491. DOI: 10.26483/ijarcs.v8i9.5017.

7. Resnick P., Iacovou N., Suchak M. et al. GroupLens: An Open Architecture for Collaborative Filtering of Netnews // Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW). 1994. P. 175–186.

8. Sarwar B., Karypis G., Konstan J. et al. Item-based collaborative filtering recommendation algorithms // Proceedings of the 10th International Conference on World Wide Web (WWW '01). 2001. P. 285–295. DOI: 10.1145/371920.372071.

9. Liu Y. Music Recommendation with Deep Learning Methods: A Survey. 2024. DOI: 10.54254/2755-2721/109/20241353.

10. Mangla P. Spotify Music Recommendation Systems // PyImageSearch : вебсайт. URL: https://pyimagesearch.com/2023/10/30/spotify-music-recommendation-systems/ (Last accessed: 23.05.2026).

11. Shittu E. Spotify personalizes audio experiences with machine learning // TechTarget : вебсайт. URL: https://www.techtarget.com/searchenterpriseai/feature/Spotify-personalizes-audio-experiences-with-machine-learning (Last accessed: 23.05.2026).

12. Carter B., Seefeld B., Ogle M. How Discover Weekly works // Tracknack Blog : вебсайт. URL: https://tracknack.com/blog/how-spotify-discover-weekly-works (Last accessed: 23.05.2026).

13. Lakhotia R. How Spotify Optimized Their Recommendation System // Scale Engineer : вебсайт. URL: https://scaleengineer.com/blog/how-spotify-optimized-their-recommendation-system (Last accessed: 23.05.2026).

14. Covington P., Adams J., Sargin E. Deep Neural Networks for YouTube Recommendations // Proceedings of the 10th ACM Conference on Recommender Systems (RecSys). 2016. P. 191–198. DOI: 10.1145/2959100.2959190.

15. YouTube Algorithm for Musicians (Get Recommended in 2026) // Halmblog Music : вебсайт. URL: https://www.halmblogmusic.com/youtube-algorithm-for-musicians/ (Last accessed: 23.05.2026).

16. The Apple Music Algorithm in 2026: A Comprehensive Guide // BeatsToRapOn : вебсайт. URL: https://beatstorapon.com/blog/the-apple-music-algorithm-in-2026-a-comprehensive-guide-for-artists-labels-and-data-scientists/ (Last accessed: 23.05.2026).

17. How the Apple Music Algorithm Works: A Guide for Independent Artists in 2026 // NotNoise : вебсайт. URL: https://notnoise.co/learn/how-apple-music-algorithm-works (Last accessed: 23.05.2026).

18. Inside Spotify's Recommendation System: A Complete Guide // Music Tomorrow : вебсайт. URL: https://www.music-tomorrow.com/blog/how-spotify-recommendation-system-works-complete-guide (Last accessed: 23.05.2026).

19. Huang Q., Jansen A., Lee J. et al. MuLan: A Joint Embedding of Music Audio and Natural Language // arXiv preprint arXiv:2208.12415. 2022.

20. Yin T. Music Track Recommendation Using Deep-CNN and Mel Spectrograms // Mobile Networks and Applications. 2023. Vol. 28. P. 2130–2137. DOI: 10.1007/s11036-023-02170-2.

21. Alotaibi S. et al. Sentiment Analysis and Lyrics Theme Recognition of Music Lyrics Based on Natural Language Processing // Journal of Electrical Systems. 2024. Vol. 20, no. 9s. P. 315–321.

22. Dinnissen K., Bauer C. Fairness and Transparency in Music Recommender Systems: Improvements for Artists // Proceedings of the 18th ACM Conference on Recommender Systems (RecSys 2024). 2024. DOI: 10.1145/3640457.3688024.

23. Ni Y., McVicar M., Santos-Rodriguez R., De Bie T. An end-to-end machine learning system for harmonic analysis of music // arXiv preprint arXiv:1107.4969. 2011.

24. McVicar M., Santos-Rodriguez R., Ni Y., De Bie T. Learning from Audio and Symbolic Representations for Harmonic Analysis of Music // IEEE Transactions on Audio, Speech, and Language Processing. 2013. Vol. 21, no. 7. P. 1475–1487. DOI: 10.1109/TASL.2013.2250268.

25. Agrawal Y., Shanker R. G. R., Alluri V. Transformer-based approach towards music emotion recognition from lyrics // Advances in Information Retrieval. Cham : Springer, 2021. P. 182–189.

26. Vaswani K., Agrawal Y., Alluri V. Multimodal Fusion Based Attentive Networks for Sequential Music Recommendation // 2021 IEEE Seventh International Conference on Multimedia Big Data (BigMM). 2021. P. 25–32.

27. Music Feature Extraction Method Based on Internet of Things Technology and Its Application // Computational Intelligence and Neuroscience. 2022. Art. 8615152. DOI: 10.1155/2022/8615152.

28. Rudd D. H., Huo H., Xu G. Leveraged Mel Spectrograms using Harmonic and Percussive Components in Speech Emotion Recognition // arXiv preprint arXiv:2312.10949. 2023.

29. McFee B. et al. librosa: Audio and Music Signal Analysis in Python // librosa : вебсайт. URL: https://librosa.org/doc/latest/tutorial.html (Last accessed: 23.05.2026).

30. Liu X., Goh K. M. ResNet: Enabling Deep Convolutional Neural Networks through Residual Learning // arXiv preprint arXiv:2510.24036. 2025.

31. Zhang J. Music Genre Classification with ResNet and Bi-GRU Using Visual Spectrograms // arXiv preprint arXiv:2307.10773. 2023.

32. Zhang A., Lipton Z. C., Li M., Smola A. J. Dive into Deep Learning // D2L.ai : вебсайт. URL: https://d2l.ai/chapter_multilayer-perceptrons/mlp.html (Last accessed: 23.05.2026).

33. Liu Y., Dasgupta A., He Q. Music Genre Classification: Ensemble Learning with Subcomponents-level Attention // arXiv preprint arXiv:2412.15602. 2024.

ДОДАТОК А

Програмна реалізація серверної частини інтелектуальної системи рекомендацій

```
import os
from pathlib import Path
from contextlib import asynccontextmanager
from fastapi import FastAPI, UploadFile, File
from fastapi.responses import JSONResponse
from fastapi.middleware.cors import CORSMiddleware
import torch
import librosa
import uvicorn
from src.utils.config import load_config
from src.features.extractor import extract_all_features
from src.features.nlp_extractor import NLPExtractor
from src.features.vibe_extractor import VibeExtractor
from src.models.cnn_model import MusicResNet, MusicMLP
from src.models.nlp_classifier import NLPClassifier

models = {}

@asynccontextmanager
async def lifespan(app: FastAPI):
    config = load_config()
    device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
    models.update({
        'resnet': MusicResNet(...).to(device), 'mlp':
MusicMLP(...).to(device),
        'nlp_classifier': NLPClassifier(...), 'nlp_extractor':
NLPExtractor(),
        'vibe_extractor': VibeExtractor(), 'config': config, 'device':
device
    ))
    yield

app = FastAPI(title="Luna Music AI", lifespan=lifespan)
```

```
app.add_middleware(CORSMiddleware, allow_origins=["*"],
allow_credentials=True, allow_methods=["*"], allow_headers=["*"])

@app.post("/analyze")
async def analyze_song(file: UploadFile = File(...)):
    temp_path = Path("temp_audio") / file.filename
    temp_path.parent.mkdir(exist_ok=True)
    try:
        if not file.filename.endswith((".mp3", ".wav", ".flac")):
            return JSONResponse(status_code=400, content={"error":
"Unsupported file type."))
        with open(temp_path, "wb") as buffer:
            buffer.write(await file.read())

        config, device = models['config'], models['device']
        y, sr_out = librosa.load(str(temp_path),
sr=config["audio"]["sample_rate"])
        num_features = extract_all_features(str(temp_path), config)

        with torch.no_grad():
            resnet_out =
models['resnet'](torch.from_numpy(...).float().to(device))
            mlp_out =
models['mlp'](torch.from_numpy(...).float().to(device))

        nlp_data =
models['nlp_extractor'].extract_text_and_language(str(temp_path))
        vibe_data =
models['vibe_extractor'].get_vibe(nlp_data.get("english_translation"))
        final_output =
models['nlp_classifier'].adjust_probabilities((resnet_out * 0.7) + (mlp_out
* 0.3), nlp_data, vibe_data)

        if temp_path.exists(): os.remove(temp_path)
        return {
            "filename": file.filename, "predictions": final_output,
```

```
            "language": nlp_data.get("language"), "is_instrumental":
nlp_data.get("is_instrumental"),
            "lyrics": nlp_data.get("original_text"), "vibes": vibe_data
        )
    except Exception as e:
        if temp_path.exists(): os.remove(temp_path)
        return JSONResponse(status_code=500, content={"error": str(e)})

if __name__ == "__main__":
    uvicorn.run("src.api.server:app", host="0.0.0.0", port=8000,
reload=True)
```

ДОДАТОК Б

Лістинг алгоритму математичної обробки та видобування акустичних ознак з музичних композицій

```
import numpy as np
import librosa

def process_and_extract_spectrogram(file_path: str, config: dict) ->
np.ndarray:
    sr = config["audio"]["sample_rate"]
    y, sr_out = librosa.load(file_path, sr=sr, mono=True)
    fragment_dur = config["audio"]["fragment_duration"]
    if len(y) > fragment_dur * sr_out:
        rms_e = librosa.feature.rms(y=y, frame_length=sr_out,
hop_length=sr_out)[0]
        best_i = np.argmax(np.convolve(rms_e, np.ones(int(fragment_dur)),
mode='valid'))
        start = best_i * sr_out
        y = y[start : start + int(fragment_dur * sr_out)]
    mel_spec = librosa.feature.melspectrogram(
        y=y, sr=sr_out, n_mels=config["features"]["n_mels"],
        hop_length=config["features"]["hop_length"],
n_fft=config["features"]["n_fft"]
    )
    mel_spec_db = librosa.power_to_db(mel_spec, ref=np.max)
    mel_spec_db = (mel_spec_db - mel_spec_db.min()) / (mel_spec_db.max() -
mel_spec_db.min() + 1e-8)
    if mel_spec_db.shape[1] > 256:
        return mel_spec_db[:, :256]
    return np.pad(mel_spec_db, ((0, 0), (0, 256 - mel_spec_db.shape[1])),
mode='constant')
```

ДОДАТОК В

Лістинг функції parse_dataset_file()

```
def parse_dataset_file(filepath: str) -> list[dict]:
    tracks, family, current_subgenre = [], "", ""
    with open(filepath, "r", encoding="utf-8") as f:
        lines = f.readlines()
    valid_genres = set(load_config()["dataset"]["genres"])
    for line in lines:
        line = line.strip()
        if not line or line.startswith("===") or line.startswith("Теги:")
or line.startswith("Лінк:"):
            continue
        if "FAMILY" in line.upper():
            family = re.sub(r'[^\w\s&-]', '', line).strip()
            continue
        if line.lower().replace(" ", "_") in valid_genres or line in
valid_genres:
            current_subgenre = line if line in valid_genres else
line.lower().replace(" ", "_")
            continue
        match = re.match(r'^\s*\d+\.\s+(.+?)\s*-\s+(.+)$', line)
        if match and current_subgenre:
            tracks.append({
                "name": match.group(2).strip(),
                "author": match.group(1).strip(),
                "source_genre": current_subgenre,
                "source_family": family,
            ))
    return tracks
```

ДОДАТОК Г

Лістинг коду функції extract_dense_fragment()

```
def extract_dense_fragment(y: np.ndarray, sr: int, duration: int = 60) ->
np.ndarray:
    target_samples = duration * sr

    if len(y) <= target_samples:
        return y

    hop = sr  # крок = 1 секунда
    rms = librosa.feature.rms(y=y, frame_length=2048, hop_length=hop)[0]

    # Ширина вікна в фреймах RMS
    window_frames = duration * sr // hop
    if window_frames >= len(rms):
        return y
    best_start = 0
    best_energy = 0
    for i in range(len(rms) - window_frames):
        energy = np.sum(rms[i:i + window_frames])
        if energy > best_energy:
            best_energy = energy
            best_start = i
    start_sample = best_start * hop
    end_sample = start_sample + target_samples

    return y[start_sample:end_sample]
```

ДОДАТОК Д

Лістинг коду extract_bpm_and_chords()

```
def extract_bpm_and_chords(y: np.ndarray, sr: int) -> tuple[float, str]:
    tempo, _ = librosa.beat.beat_track(y=y, sr=sr)
    tempo = float(tempo[0]) if isinstance(tempo, np.ndarray) and len(tempo)
> 0 else float(tempo)
    bpm = round(tempo, 1)

    y_harmonic, _ = librosa.effects.hpss(y)
    chroma_mean = np.mean(librosa.feature.chroma_cqt(y=y_harmonic, sr=sr),
axis=1)
    pitch_classes = ['C', 'C#', 'D', 'D#', 'E', 'F', 'F#', 'G', 'G#', 'A',
'A#', 'B']
    major_profile = np.array([6.35, 2.23, 3.48, 2.33, 4.38, 4.09, 2.52,
5.19, 2.39, 3.66, 2.29, 2.88])
    minor_profile = np.array([6.33, 2.68, 3.52, 5.38, 2.60, 3.53, 2.54,
4.75, 3.98, 2.69, 3.34, 3.17])

    best_corr, best_key, best_mode = -2, 0, "major"
    for shift in range(12):
        shifted = np.roll(chroma_mean, shift)
        for profile, mode in [(major_profile, "major"), (minor_profile,
"minor")]:
            corr = np.corrcoef(shifted, profile)[0, 1]
            if corr > best_corr:
                best_corr, best_key, best_mode = corr, shift, mode

    top_notes = [pitch_classes[i] for i in np.argsort(chroma_mean)[::-
1][:3]]
    return bpm,
f"{pitch_classes[best_key])_{best_mode).{'.'.join(top_notes))"
```