

Вінницький Національний Технічний Університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра програмного забезпечення

Бакалаврська дипломна робота

на тему: Розробка програми для аналізу та класифікації музичних жанрів на основі аудіозаписів з використанням алгоритмів машинного навчання.

Виконав: студент IV курсу, групи 2ПІ-206
спеціальності

121 – Інженерія програмного забезпечення
(назва спеціальності)

Іванченко Андрій Володимирович
(прізвище та ім'я)

Керівник: к.т.н., доц. каф. ПЗ Чехмestрук Р.Ю.
(прізвище та ім'я)

Рецензент: к.т.н., доц. каф. ОТ Войцеховська О.В.
(прізвище та ім'я)

Допущено до захисту

Зав. кафедри Романюк О.Н.

«10» червня 2024 р.

ВНТУ – 2024

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра програмного забезпечення
Рівень вищої освіти перший (бакалаврський)
Галузь знань 12 – «Інформаційні технології»
Спеціальність 121 – «Інженерія програмного забезпечення»
Освітньо-професійна програма – «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ
Завідувач кафедри ПЗ
п.т.н., професор
О. Н. Романюк
«12» березня 2024 р.

ЗАВДАННЯ НА БАКАЛАВРСЬКУ ДИПЛОМНУ РОБОТУ СТУДЕНТУ

Іванченку Андрію Володимировичу

1. Тема роботи – розробка програми для аналізу та класифікації музичних жанрів на основі аудіозаписів з використанням алгоритмів машинного навчання.

Керівник роботи: Керівник роботи: Чехмestрук Роман Юрійович, к.т.н. доцент кафедри ПЗ, затверджені наказом вищого навчального закладу від 11 березня 2024 р. № 80.

2. Строк подання студентом роботи 3 червня 2024.

3. Вихідні дані до роботи:

4. Зміст текстової частини: вступ; аналіз стану застосунків для аналізу музичних аудіозаписів; порівняльний аналіз аналогів; аналіз методів розв'язання задачі; постановка задачі; аналіз вхідних даних системи; розробка інтерфейсу програмного застосунку; розробка моделі системи; розробка алгоритмів роботи системи; аналіз і обґрунтування засобів розробки; огляд роботи програмних модулів; тестування розробленого програмного забезпечення, створення інструкції користувача.

5. Перелік ілюстративного матеріалу: титульний слайд; актуальність; тема; мета, об'єкт та предмет дослідження; задачі дослідження; наукова новизна; практична цінність отриманих результатів; блок-схема загального алгоритму роботи веб-системи; аналоги програмного застосунку; вихідні дані моделі машинного навчання; інтерфейс системи; тестування та валідація веб-системи; апробація та публікація результатів; фінальний слайд.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
1-4	Чехместрук Р.Ю., к.т.н., доц. каф. ПЗ	13.03.24	03.06.24

7. Дата видачі завдання 13 березня 2024р

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів бакалаврської дипломної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз існуючих сервісів та додатків, що працюють з аудіозаписами	14.03.24 - 22.03.24	Виконано
2	Збір та підготовка даних для тренування моделі класифікації	24.03.24 - 30.03.24	Виконано
3	Розробка алгоритмів машинного навчання для класифікації	02.04.24 - 20.04.24	Виконано
4	Розробка зручного інтерфейсу користувача	21.04.24 - 10.05.24	Виконано
5	Проведення тестування та валідації розробленої моделі	11.05.24 - 24.05.24	Виконано
6	Оформлення матеріалів до захисту БДР	25.05.24 - 03.06.24	Виконано

Студент А. В. Іванюк

Керівник бакалаврської дипломної роботи Р. Ю. Чехместрук

АНОТАЦІЯ

УДК 004.912.032.26

Іванченко А. В. Розробка програми для аналізу та класифікації музичних жанрів на основі аудіозаписів з використанням алгоритмів машинного навчання: бакалаврська дипломна робота зі спеціальності 121 Інженерія програмного забезпечення, освітня програма – інженерія програмного забезпечення. Вінниця: ВНТУ, 2024. 55 с.

Укр. мовою. Бібліогр. : назв 22; рис. : 42; табл. 3.

У бакалаврській дипломній роботі проведено детальний аналіз методів та засобів машинного навчання для реалізації додатку для класифікації музичних аудіозаписів за жанрами. Запропоновано новий підхід до визначення жанру музичного аудіозапису – за допомогою алгоритмів машинного навчання. Це дозволяє користувачам визначити жанр власного аудіозапису.

Розроблено програмні засоби для створення застосунку, що забезпечує аналіз та класифікацію музичних жанрів на основі аудіозаписів. Застосунок надає користувачам можливість отримати інформацію про жанр музики на підставі її звукових характеристик. Застосунок розроблено з використанням мови Python та середовища PyCharm. Для створення бекенд частини використовувався фреймворк Django.

Отримані в бакалаврській дипломній роботі результати можуть бути використані розробниками для створення застосунків, які включають аналіз музичних композицій та надають інформацію про їхні жанри. Це може бути корисним у різних сферах, включаючи музичні платформи, освітні додатки та рекомендаційні системи.

Ключові слова: музика, жанр, машинне навчання.

ABSTRACT

UDC 004.912.032.26

Ivanchenko A. V. Development of a program for the analysis and classification of music genres based on audio recordings using machine learning algorithms: Bachelor's thesis in the specialty 121 Software Engineering, educational program – software engineering. Vinnytsia: VNTU, 2024. 53 pages.

In Ukrainian. Bibliography: titles: 22; figures: 42; tables: 3.

This bachelor's thesis provides a detailed analysis of methods and tools of machine learning for implementing an application to classify music audio recordings by genre. A new approach to defining the genre of a music audio recording using machine learning algorithms is proposed. This allows users to determine the genre of their own audio recordings.

Software tools have been developed to create an application that analyzes and classifies music genres based on audio recordings. The application allows users to obtain information about the music genre based on its sound characteristics. The application is developed using the Python programming language and the PyCharm environment. The backend part was created using the Django framework.

The results obtained in this bachelor's thesis can be used by developers to create applications that include the analysis of music compositions and provide information about their genres. This can be useful in various fields, including music platforms, educational applications, and recommendation systems.

Keywords: music, genre, machine learning.

ЗМІСТ

ВСТУП.....	3
1 АНАЛІЗ РОЗВИТКУ ЗАСТОСУНКІВ ДЛЯ АНАЛІЗУ ТА КЛАСИФІКАЦІЇ МУЗИЧНИХ АУДІОЗАПИСІВ ТА ПОСТАНОВКА ЗАДАЧ ДОСЛІДЖЕННЯ	6
1.1 Аналіз стану застосунків для аналізу музичних аудіозаписів.....	6
1.2 Порівняльний аналіз аналогів.....	7
1.3 Аналіз методів розв’язання задачі.....	12
1.4 Постановка задач проєкту.....	14
1.5 Висновки	15
2 РОЗРОБКА МОДЕЛІ ТА АЛГОРИТМІВ ПРОГРАМНОГО ПРОДУКТУ..	16
2.1 Аналіз вхідних даних системи	16
2.2 Розробка інтерфейсу програмного додатку	18
2.3 Розробка моделі системи.....	20
2.4 Розробка алгоритмів роботи системи	21
2.5 Висновки	23
3. ОГЛЯД ПРОГРАМНИХ МОДУЛІВ РЕАЛІЗАЦІЇ ВЕБ-ЗАСТОСУНКУ ДЛЯ КЛАСИФІКАЦІЇ МУЗИЧНИХ АУДІОФАЙЛІВ ЗА ЖАНРАМИ З ВИКОРИСТАННЯМ МАШИННОГО НАВЧАННЯ	24
3.1 Варіантний аналіз і обґрунтування вибору засобів для реалізації програмного забезпечення.....	24
3.2 Огляд роботи моделі машинного навчання.....	25
3.3 Огляд роботи програмного модуля для розділення музичного аудіозапису на складові та подальшого аналізу.....	27
3.4 Висновки	30
4 ТЕСТУВАННЯ ПРОГРАМИ.....	31
4.1 Тестування і валідація системи.....	31
4.2 Розробка інструкції користувача	49
4.3 Висновки	51
ВИСНОВКИ	52
ЛІТЕРАТУРА.....	54
ДОДАТКИ.....	56
Додаток А – Технічне завдання	57
Додаток Б – Протокол перевірки на плагіат	60
Додаток В – Лістинг програми	61
Додаток Г – Ілюстративна частина	80

ВСТУП

Обґрунтування вибору теми дослідження. Сучасний музичний асортимент вражає своєю безмежною різноманітністю жанрів та піджанрів, що відображають широкий спектр культурних, соціальних та індивідуальних впливів. Ця надзвичайна диверсифікація створює складнощі у точному визначенні жанру музичних композицій, оскільки багато жанрів мають характеристики, які перекриваються та не можуть бути чітко визначені.

Актуальність. У сучасному світі більшість виконавців самі визначають жанр своїх творів, проте це визначення може бути суб'єктивним та недостатньо точним. У багатьох випадках, особливо у випадку музики, яка поєднує в собі різні стилі та впливи, визначення жанру стає складним завданням, що потребує об'єктивного підходу. Тому розробка програмних засобів системи визначення музичних жанрів аудіозаписів є надзвичайно актуальною задачею.

Зв'язок роботи з науковими програмами, планами, темами. Робота виконувалася згідно плану виконання наукових досліджень на кафедрі програмного забезпечення

Мета роботи – впровадження ефективного методу класифікації музичних композицій на основі аналізу їхніх характеристик та особливостей. Це дозволить підвищити точність і об'єктивність визначення музичних жанрів, що є важливим для музичних платформ, виконавців та користувачів.

Головними завданнями, які необхідно вирішити під час реалізації проєкту, є:

- аналіз існуючих сервісів та додатків, що працюють з аудіозаписами;
- збір та підготовка даних для тренування моделі класифікації;
- розробка алгоритмів машинного навчання для класифікації;
- розробка зручного інтерфейсу користувача;
- проведення тестування та валідації розробленої моделі.

Об'єктом дослідження є процес обробки аудіоданих та аналізу звукових сигналів, а також методи та алгоритми машинного навчання, які застосовуються для класифікації музичних композицій.

Предметом дослідження є розробка програмних засобів, які дозволять автоматизувати процес визначення жанру музичних творів на основі обробки аудіозаписів та аналізу їхніх акустичних та музичних властивостей.

Методи дослідження. У процесі дослідження використовувались: аналітичний метод – для збору та аналізу даних про існуючі музичні сервіси та алгоритми класифікації; метод машинного навчання – для розробки та тренування моделей класифікації музичних жанрів; метод програмування – для реалізації програмного забезпечення; метод тестування – для перевірки функціональності та точності розробленої моделі класифікації.

Новизна отриманих результатів. У бакалаврській дипломній роботі було розроблено новий підхід до класифікації музичних жанрів, що базується на використанні алгоритмів машинного навчання. Запропоноване рішення дозволяє автоматизувати процес визначення жанру музичних композицій, забезпечуючи високу точність та об'єктивність результатів.

Практичне значення одержаних результатів полягає в створенні інструменту, який може бути використаний для автоматизованого визначення жанру музичних композицій у різноманітних ситуаціях, таких як аналіз великих обсягів аудіоданих, рекомендації музики на музичних платформах або створення персоналізованих плейлистів для користувачів.

Особистий внесок здобувача. Весь внесок у цю роботу був зроблений особисто автором, що включає в себе ідею проекту, його проектування, розробку, тестування, аналіз потреб користувачів та ринку, щоб забезпечити максимальну ефективність та корисність продукту.

Структура та обсяг роботи. Пояснювальна записка до бакалаврської дипломної роботи складається з чотирьох розділів.

У першому розділі розглянуто поточний стан машинних технологій в аналізуванні та класифікації музичних аудіозаписів та описано важливість даної

розробки у музичному світі. Також проведено порівняння запропонованого рішення із аналогами та сформульовано завдання проєкту.

У другому розділі надано детальний опис реалізації проєкту, включаючи використані мови програмування, технології та інструменти. Також описано алгоритм та графічний інтерфейс, які використовуються для реалізації програмного забезпечення.

Третій розділ присвячений огляду роботи програмних модулів, необхідних для програмного застосунку. У розділі описані алгоритми та моделі, використані для розробки модулів, а також технології та інструменти, використані в процесі розробки.

У четвертому розділі проведено тестування функціональних та графічних компонентів програмного забезпечення, розробленого в рамках бакалаврської дипломної роботи, а також надано керівництво користувача для системи.

Апробація результатів. здійснена на конференції XXIV Всеукраїнської науково-технічної конференції молодих вчених, аспірантів та студентів (2024) і опубліковані в тезах доповіді цієї конференції.

Пояснювальна записка містить 55 сторінок тексту, 42 рисунка, 3 таблиці, чотири додатки, 22 найменування в списку використаних джерел.

1 АНАЛІЗ РОЗВИТКУ ЗАСТОСУНКІВ ДЛЯ АНАЛІЗУ ТА КЛАСИФІКАЦІЇ МУЗИЧНИХ АУДІОЗАПИСІВ ТА ПОСТАНОВКА ЗАДАЧ ДОСЛІДЖЕННЯ

1.1 Аналіз стану застосунків для аналізу музичних аудіозаписів.

На сьогоdnішньому етапі розвитку технологій важливо постійно вдосконалювати методи аналізу та вимірювань у сфері музичної класифікації. Традиційні методи, хоч і залишаються актуальними, часто обмежені і не завжди забезпечують достатню точність та деталізацію. У цьому контексті використання сучасних технологій аналізу аудіоданих, здатних аналізувати звукові сигнали та визначати їхні особливості, стає надзвичайно актуальним.

Один із ключових аспектів цього розвитку – це зростання швидкості та потужності обчислювальних систем. Сучасні комп'ютери та методи обробки даних дозволяють швидко та ефективно аналізувати великі обсяги аудіоданих, що є необхідним для визначення музичних жанрів та їхніх характеристик.

Важливим елементом розробки програмного забезпечення для визначення музичних жанрів – є розробка алгоритмів аналізу аудіозаписів. Ці алгоритми є основою для автоматичного визначення жанру музичних композицій та включають в себе комплексний аналіз різноманітних аспектів звукового сигналу.

Під час розробки таких алгоритмів враховується низка параметрів, що визначають характеристики музичної композиції. Основні з них включають ритм, який описує темп та ритмічну структуру музики; темп, що визначає швидкість виконання композиції; тембр, що характеризує звукові властивості інструментів та голосу; гармонію, яка відображає акордову структуру та музичні зв'язки; а також інші параметри, які можуть включати в себе частоту, амплітуду, спектральний зміст тощо. Важливо враховувати, що розробка алгоритмів аналізу аудіозаписів є складною задачею, оскільки музичні композиції можуть мати велику різноманітність та складність. Такі алгоритми повинні бути достатньо гнучкими та ефективними для визначення жанрів у різних умовах, включаючи різні стилі та піджанри музики.

Крім того, важливо розробляти графічний інтерфейс, який би спрощував взаємодію з програмним забезпеченням та надавав зручність у користуванні. Це може включати в себе візуалізацію результатів аналізу, можливість налаштування параметрів алгоритмів та інші функції.

Отже, розробка програмних засобів для визначення музичних жанрів аудіозаписів є важливим напрямком розвитку сучасних технологій у музичній індустрії. Вона сприяє автоматизації процесу класифікації музики, що є важливим для музичних платформ, радіостанцій, а також для музикознавців та користувачів.

1.2 Порівняльний аналіз аналогів

Оскільки, з кожним днем все більше людей виявляють зацікавленість у музиці та її властивостях, при цьому мають зовсім різний музичний смак у великому розмаїтті музичних жанрів, компанії, що надають засоби для прослуховування аудіозаписів або перегляду музичних відео, поставили перед собою завдання програмно аналізувати музичні аудіозаписи та класифікувати їх. Це дозволяє забезпечити користувачів рекомендаціями, що відповідають їхньому смаку, а також використовується для інших цілей, таких як підбір контенту для певних аудиторій, аналіз трендів у музичній індустрії та інше.. Із відомих на даний час додатків було виділено:

- Shazam;
- Youtube;
- Spotify.

Shazam – це додаток для мобільних пристроїв та онлайн-платформа, що дозволяє користувачам визначати назву та виконавця пісні, яка грає в момент запуску додатка. Основна функція Shazam полягає у розпізнаванні аудіозапису, а також у наданні користувачам інформації про композицію, включаючи назву пісні, виконавця, альбом та інші деталі.

Робота додатку Shazam базується на унікальному алгоритмі аналізу звуку. При запуску, користувач спрямовує пристрій на джерело звуку, наприклад, на динамік відтворювача музики. Додаток за допомогою мікрофону записує

невеликий уривок звуку та передає його на сервери Shazam для подальшого аналізу. На серверах Shazam використовується спеціальний алгоритм, який порівнює аудіофрагмент з базою даних, що містить мільйони записів. Ця база даних містить інформацію про мільйони пісень, а також характеристики їхніх звукових сигналів. Після успішного збігу алгоритм повертає користувачеві інформацію про пісню, яка була розпізнана[2].

Крім того, Shazam також надає користувачам можливість прослуховувати пісні онлайн, переглядати їх текст та отримувати рекомендації щодо подібних композицій. Додаток також може зберігати історію розпізнаних пісень, щоб користувач міг зручно повертатися до них у майбутньому.

Робота сервісу Shazam показана на рисунку 1.1.

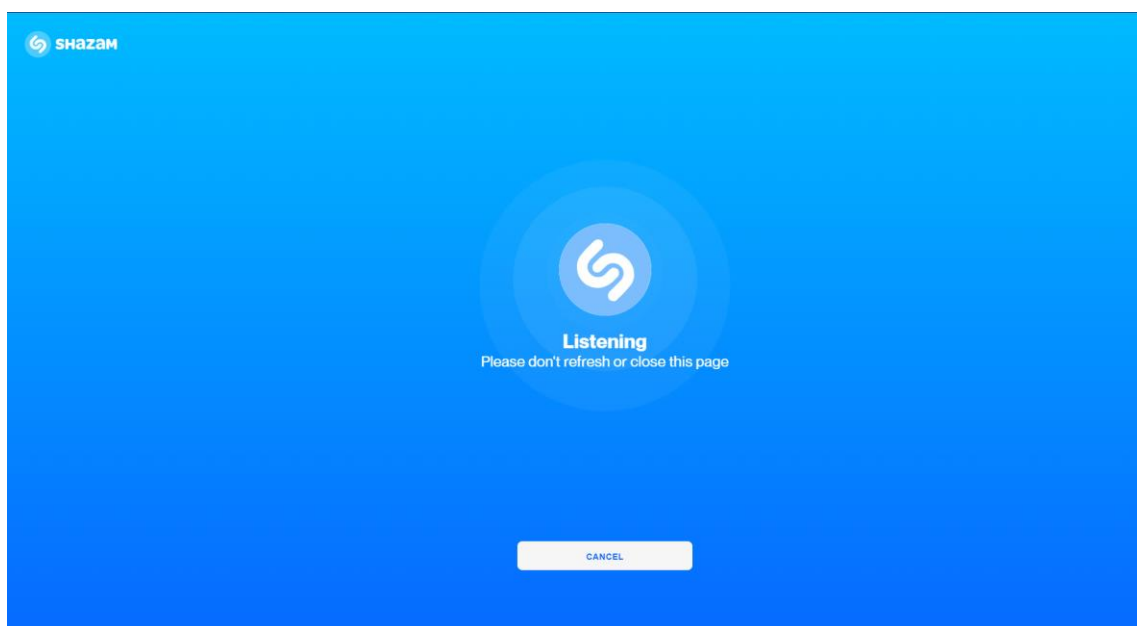


Рисунок 1.1 – Робота сервісу Shazam

Ще одним аналогом обраним для аналізу став всім звичний Youtube. Загалом, Youtube – це онлайн-платформа для перегляду та поширення відеоконтенту, включаючи музичні відео, концертні записи, музичні виступи, аудіо треки та інше [3]. Хоча основна функція YouTube полягає в перегляді відео, ця платформа також має безліч функцій та можливостей, які стосуються аналізу музичного контенту.

Однією з таких функцій є можливість рекомендаційного відтворення. YouTube використовує алгоритми рекомендацій, які аналізують історію перегляду та вподобань користувача для пропозиції відео, які можуть бути для нього цікавими. Це дозволяє користувачам знаходити нову музику, що відповідає їхнім смакам та уподобанням. До прикладу на рисунку 1.2 було показано рекомендаційний плейлист з музикою підібраний алгоритмами Youtube.



Рисунок 1.2 – Плейлист, що був підібраний алгоритмами Youtube за вподобаннями користувача

Третім аналогом, що було обрано для порівняння є Spotify – поширена онлайн-платформа для прослуховування музики. Spotify використовує ряд алгоритмів та методів для підбору та рекомендацій музики своїм користувачам, наприклад:

- Обробка природної мови – аналізує тексти пісень, відгуки користувачів та описи артистів, щоб зрозуміти контекст та настрій пісень.
- Аналіз аудіо – аналізує аудіозаписи для розпізнавання характеристик музики, таких як темп, ритм, тембр, гармонія, акустична складність тощо. На основі цих характеристик Spotify може рекомендувати користувачеві пісні, які подібні за музичними ознаками до тих, які він слухає.

На рисунку 1.3 зображено плейлист підібраний платформою Spotify на основі вподобань користувача.

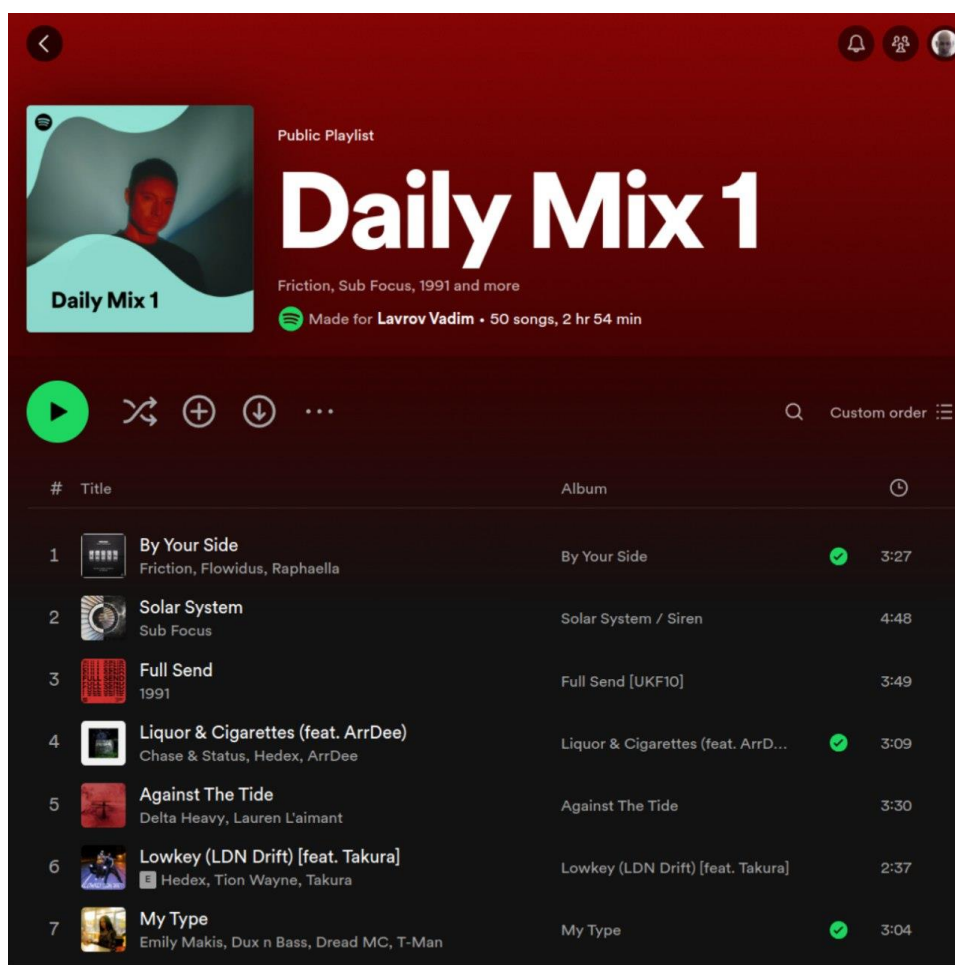


Рисунок 1.3 – Рекомендаційний плейлист Spotify

Слід зазначити, що представлені аналоги не мають прямого зв'язку із розроблюваним програмним засобом, тобто розроблюється унікальний алгоритм, що допоможе класифікувати будь-який музичний аудіозапис за жанром, навіть якщо він був створений користувачем, в той час як зазначені аналоги працюють із уже наявною інформацією про аудіозапис.

Результати порівняльного аналізу аналогів зведено в таблицю 1.1.

Таблиця 1.1 – Порівняння аналогів

Критерії оцінювання	Shazam	Youtube	Spotify	Розроблюваний програмний застосунок
Відкритий доступ до алгоритму дії і користування ним	+	-	-	+
Підтримка різних ОС	+	+	+	+
Можливість аналізу нових аудіозаписів	-	-	-	+
Можливість розширення функціоналу	+	-	-	+
Можливість інтеграції в інші додатки	+	+	-	+
Загальна оцінка	80%	40%	20%	100%

Відповідно до вказаних критеріїв, розробка власного програмного застосунку для аналізу музичних аудіозаписів виправдана. Новий програмний продукт має забезпечити користувачам доступ до нових алгоритмів та надати їм новий досвід аналізу музичних аудіозаписів. Крім того, застосунок повинен дозволяти користувачам прослуховувати аудіозаписи, аналізувати нові аудіозаписи, розширювати функціонал та інтегруватися з іншими застосунками. Ці функціональні можливості мають покрити недоліки аналогів і забезпечити зручніше та ефективніше використання користувачами.

Отже, розробка програмного застосунку для класифікації музичних аудіозаписів за жанром відкриває нові двері у музичному світі. Виконавці та платформи для прослуховування зможуть використовувати розроблений застосунок для точнішого аналізу аудіозаписів та визначення вподобань користувачів.

1.3 Аналіз методів розв’язання задачі

У процесі розробки програмних засобів для класифікації музичних аудіозаписів за жанрами необхідно застосовувати комплексний підхід, який включає аналіз різноманітних методів та їх оптимізацію для отримання найкращих результатів.

Один із ключових аспектів розробки системи визначення жанрів музичних аудіозаписів – це вибір та розробка алгоритмів для аналізу аудіосигналів. Це вимагає дослідження та розуміння різних аспектів звукового сигналу, таких як ритм, темп, тембр, мелодійність та інші, з метою точного визначення жанру музичної композиції. Для досягнення цієї мети можуть бути використані різні методи, включаючи цифрову обробку сигналів, машинне навчання та аналіз даних. Важливо розробляти алгоритми, які можуть ефективно враховувати характеристики звукового сигналу та виявляти унікальні риси кожного музичного жанру для надання найточнішого класифікаційного результату.

Розглянемо їх детальніше:

Метод цифрової обробки сигналів (ЦОС) в контексті аналізу музичних записів включає в себе ряд технік, які дозволяють аналізувати та екстрагувати різні аспекти звукового сигналу з метою визначення музичних характеристик, включаючи жанр.

Одним із основних методів ЦОС у цьому контексті є аналіз спектру звуку. Цей метод використовує перетворення Фур'є для перетворення аудіосигналу з часової області в частотну область. Результатом є спектрограма, яка відображає енергію сигналу в залежності від частоти та часу. З аналізу спектрограми можна отримати важливу інформацію про ритм, мелодійність, гармонію та інші аспекти музичного сигналу, що може бути корисною для визначення його жанру.

Ще одним методом ЦОС є аналіз амплітуди та частоти. Цей метод дозволяє визначити основні характеристики звуку, такі як гучність, частотний діапазон та динаміку. Наприклад, низькі частоти можуть вказувати на наявність басів, тоді як високі частоти можуть свідчити про наявність вокалу або інструментальних партій.

Також важливим методом ЦОС є аналіз темпу та ритму. Цей метод дозволяє визначити швидкість та ритм музичного сигналу, що є ключовими характеристиками багатьох музичних жанрів. Наприклад, темп може бути швидким для танцювальної музики або повільним для балад.

Метод аналізу даних у контексті визначення музичних жанрів аудіозаписів включає в себе різноманітні підходи та техніки для обробки та інтерпретації інформації, отриманої з музичних записів. Розглянемо два ключових пункти у цьому методі:

- Аналіз характеристик аудіосигналу: Включає в себе вимірювання та аналіз різних характеристик аудіосигналу, таких як спектральна енергія, темп, ритм, тембр і т.д. Алгоритми аналізу сигналу, такі як перетворення Фур'є, можуть бути використані для визначення унікальних особливостей кожного жанру та використані для класифікації записів за їхніми характеристиками.
- Аналіз текстової інформації: У додаток до аналізу аудіосигналу, можна також аналізувати текстову інформацію, що супроводжує музичні записи, таку як

назва пісні, виконавець, текст пісні, теги та інше. Ця інформація може допомогти в зрозумінні контексту та характеристик певного жанру та використовується для поліпшення точності класифікації.

Ще одним з можливих підходів до аналізу музичних записів є використання алгоритмів машинного навчання, які тренуються на великій кількості аудіозаписів з різних жанрів. Ці алгоритми можуть використовувати різноманітні ознаки аудіозаписів, такі як спектральні характеристики, темп, ритм, тембр, акустичні властивості та інші, для класифікації їх за жанрами. Використання машинного навчання дозволяє автоматизувати процес визначення жанрів музики та забезпечити високу точність класифікації.

На практиці може бути ефективним поєднання різних методів та підходів для досягнення найкращих результатів у визначенні музичних жанрів аудіозаписів.

Отже, оцінивши різні можливі методи розв'язання поставленої задачі, ми прийняли рішення сконцентруватися на використанні методів машинного навчання. Це було обрано з метою забезпечення точної та надійної класифікації музичних аудіозаписів за їхніми жанрами.

1.4 Постановка задач проєкту

Провівши аналіз аналогів та визначивши їх сильні та слабкі сторони, а також оцінивши методи можливого вирішення задачі, було сформульовано наступні завдання для реалізації проєкту:

- збір та підготовка даних для тренування моделі класифікації;
- розробка алгоритмів машинного навчання для класифікації;
- розробка алгоритмів роботи програми із аудіозаписами;
- навчання моделі машинного навчання класифікувати музичні аудіозаписи за жанрами;
- розробка зручного інтерфейсу користувача;
- проведення тестування та валідації розробленої моделі.

1.5 Висновки

У першому розділі було проведено аналіз аналогів, зокрема, популярних сервісів, таких як Shazam, YouTube та Spotify, що також працюють з музичними аудіозаписами. Під час цього аналізу були виявлені переваги та недоліки цих платформ. Наприклад, Shazam відомий своєю високою швидкістю розпізнавання музики та широким функціоналом, включаючи можливість створення списків відтворення та отримання інформації про виконавців. У той же час, YouTube надає безкоштовний доступ до великого обсягу музичного контенту, а Spotify відзначається високою якістю аудіопотоку та рекомендаціями, відповідними особистим музичним смакам користувачів.

Однак під час аналізу стало очевидним, що прямі аналоги для розробленої системи визначення музичних жанрів аудіозаписів відсутні. Це вимагало детальнішого огляду алгоритмів, які використовуються в подібних музичних платформах, та вибору того, що найбільш підходить для досягнення оптимальних результатів. На підставі цього аналізу були сформульовані завдання для подальшої розробки.

2 РОЗРОБКА МОДЕЛІ ТА АЛГОРИТМІВ ПРОГРАМНОГО ПРОДУКТУ

2.1 Аналіз вхідних даних системи

Для реалізації алгоритму машинного навчання на основі аудіозаписів було використано засоби мови програмування Python[4]. Python – це інтерпретована мова програмування, яка використовується для широкого спектру завдань, включаючи розробку веб-додатків, аналіз даних, машинне навчання, наукові обчислення та тривимірне моделювання. Вона відома своєю простотою та зручністю використання, що робить її популярним вибором серед розробників будь-якого рівня досвіду. Безпосередньо у створенні алгоритмів машинного навчання, крім базових можливостей мови програмування Python, було використано такі бібліотеки як:

- librosa[5] – бібліотека для аналізу та обробки аудіоданих;
- scikit-learn[6] – бібліотека машинного навчання;
- numpy[7] – бібліотека обчислення числових даних;
- matplotlib[8] – бібліотека для створення та зберігання графіків;
- pandas[9] – бібліотека для обробки та аналізу даних.

Дані для навчання моделі було отримано із відкритої теки інформації GTZAN Dataset[10]. У теці зібрано та оброблено для ефективного навчання тисяча пісень – по 100 пісень кожного із 10 жанрів.

Після навчання розробленої моделі та її тестування необхідно розробити користувацький інтерфейс. Було вирішено відтворити розроблений застосунок у форматі веб-сторінки.

Для розробки веб-сторінки було використано засоби мов HTML, CSS та JS.

HTML – це мова розмітки гіпертекстових документів, яка використовується для створення веб-сторінок[11]. Вона визначає структуру та вміст веб-сторінок за допомогою різноманітних тегів і атрибутів.

CSS – це мова стилів, яка використовується для оформлення та стилізації веб-сторінок, що написані на HTML[11]. Для зручності та ефективності розробки було використано фреймворк Bootstrap.

Bootstrap – це потужний фреймворк для розробки веб-інтерфейсів, що містить набір зручних інструментів, компонентів та стилів, які значно спрощують створення стильних та адаптивних веб-сторінок[12].

JS (JavaScript) – це високорівнева, інтерпретована мова програмування, що використовується для створення інтерактивних веб-сайтів та веб-додатків[13]. Вона дозволяє додавати функціональність до веб-сторінок, взаємодіючи з користувачем, обробляючи події та змінюючи вміст сторінки динамічно без необхідності перезавантаження. Крім основних засобів мови JS було використано фреймворк jQuery.

jQuery – це швидкий, легкий та потужний JavaScript фреймворк, який спрощує взаємодію з DOM веб-сторінки, а також реалізує анімації, обробку подій, взаємодію з AJAX та інші функціональності[14].

Після розробки фронтенду було необхідно забезпечити веб-сторінку серверною стороною. Для розробки бекенду було використано засоби мови Python, а саме – фреймворк Django.

Django – це відкритий веб-фреймворк для Python, який дозволяє швидко створювати веб-додатки складної структури та високої якості[15].

Отже, для реалізації проєкту машинного навчання для класифікації музичних аудіозаписів було використано засоби мови програмування Python із бібліотеками для простішої реалізації машинного навчання, та із фреймворком Django для реалізації бекенду. Для розробки фронтенду сайту було використано мови HTML, CSS та JS із використанням фреймворків Bootstrap та jQuery. Також було ретельно відібрано та підлаштовано музичні аудіозаписи для кращого навчання моделі машинного навчання. Із такими вхідними даними було розпочато розробку даного проєкту.

2.2 Розробка інтерфейсу програмного додатку

Розробка інтерфейсу веб-сайту була сконцентрована на зрозумілості послідовності дій для майбутнього користувача. Для задоволення естетичних показників було використано темні тони у реалізації інтерфейсу.

Після входу на сайт користувача зустрічає головна сторінка, що зображено на рисунку 2.1.

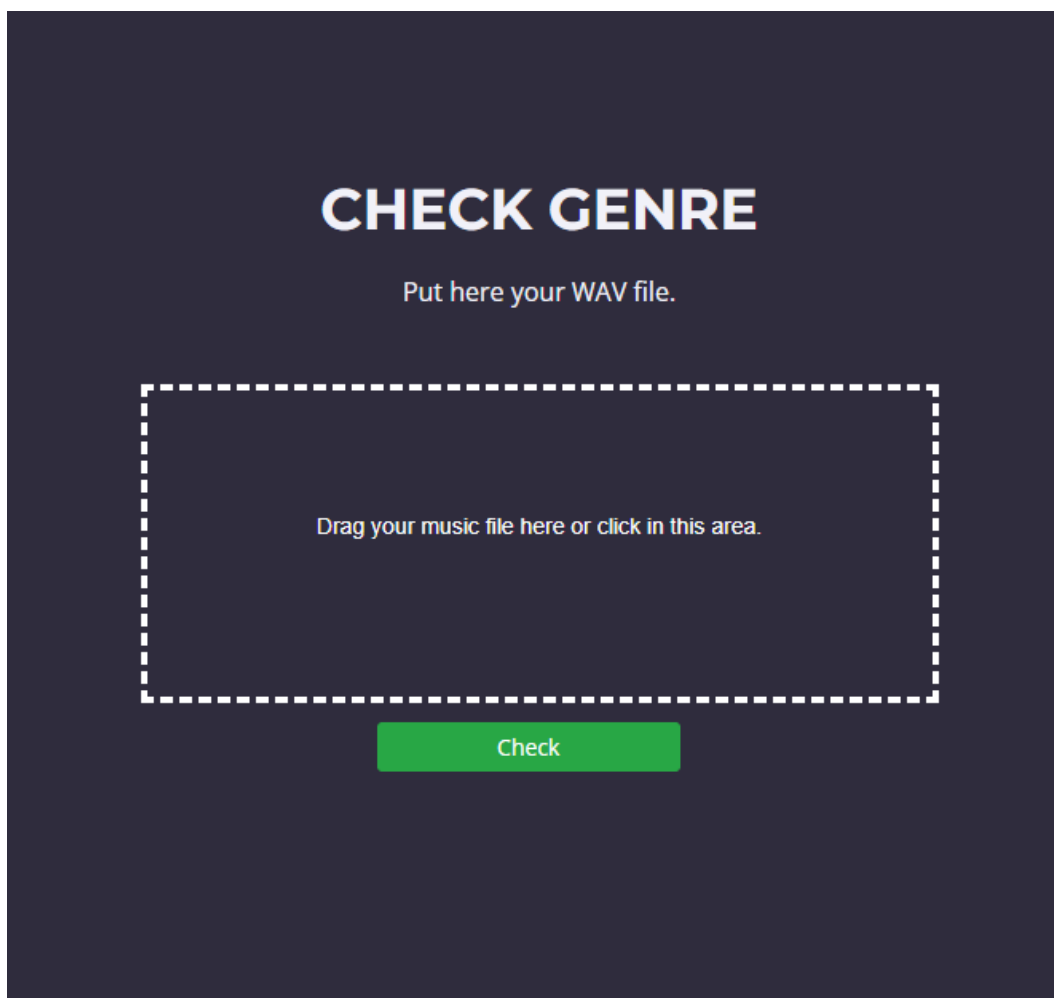


Рисунок 2.1 – Головна сторінка сайту

Оскільки, теперішній стан застосунку не включає нічого крім аналізу музичних аудіозаписів, було вирішено не обтяжувати інтерфейс користувача зайвими картинками та сконцентруватись на зрозумілості. На головній сторінці можна побачити вказівки правильних дій та вікно для загрузки аудіофайлу WAV формату. Аудіозапис можна як перетягнути у відведену зону, так і клікнути на неї

для ручної загрузки аудіофайлу із файлів користувача. При загрузці файлу неправильного формату сайт вкаже користувачу на помилку та надасть інструкції, приклад загрузки неправильного файлу вказано на рисунку 2.2.

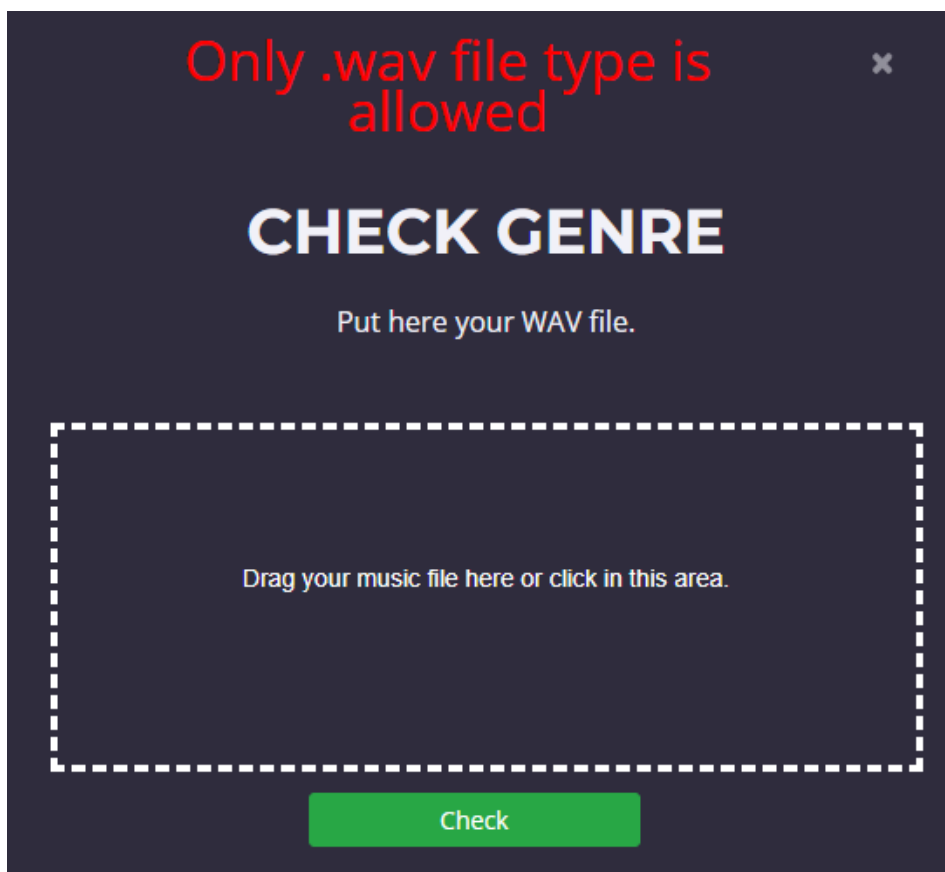


Рисунок 2.2 – Помилка при завантаженні неправильного формату файлу

Після завантаження аудіофайлу правильного формату та натисканні кнопки «Check» модель починає оброблювати завантажений файл та виділяти найбільш притаманний жанр для нього із відомих для розробленої моделі.

Після завершення аналізу аудіофайлу користувач переходить на сторінку результатів, де відображається прогнозований жанр музики для обраного аудіофайлу. Наприклад, у випадку аналізу пісні гурту Motörhead – Orgasmatrone, на сторінці результатів може бути вказано, що прогнозований жанр цієї аудіозапису – «Rock».

На рисунку 2.3 зображено вигляд сторінки результатів з вказаним прогнозованим жанром аудіозапису. Це дозволяє користувачеві швидко та зручно отримати інформацію про класифікацію обраного аудіофайлу.

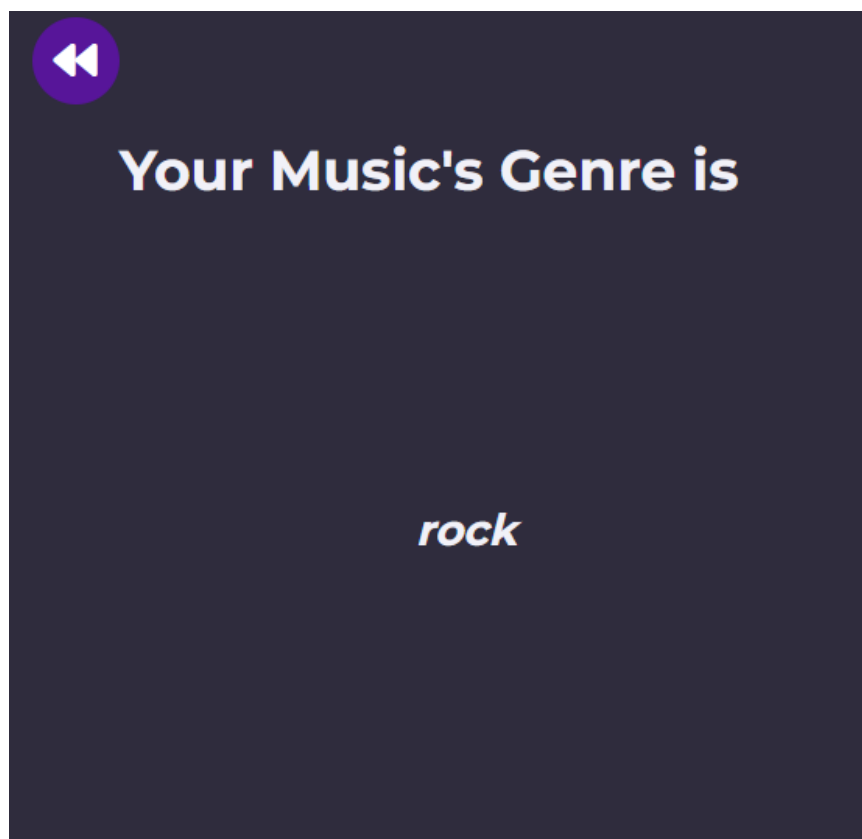


Рисунок 2.3 – Результат аналізу аудіозапису

Отримавши результат користувач має можливість перейти назад на головну сторінку та здійснити ще один аналіз.

Отже, для розробленого проєкту було розроблено мінімалістичний графічний інтерфейс у форматі веб-сторінки у темних тонах. При створенні інтерфейсу було відкинуто непотрібні елементи та залишено лише функціональні для легшого розуміння послідовності дій.

2.3 Розробка моделі системи

Для наочної демонстрації роботи веб-застосунку було створено UML діаграму діяльності, яку продемонстровано на рисунку 2.4. Першим кроком

користувач має обрати файл із наявних та завантажити його. Файл перевіряється на відповідність до потрібного формату і після успішного проходження перевірки аналізується та користувач отримує результат на сторінці з результатами.

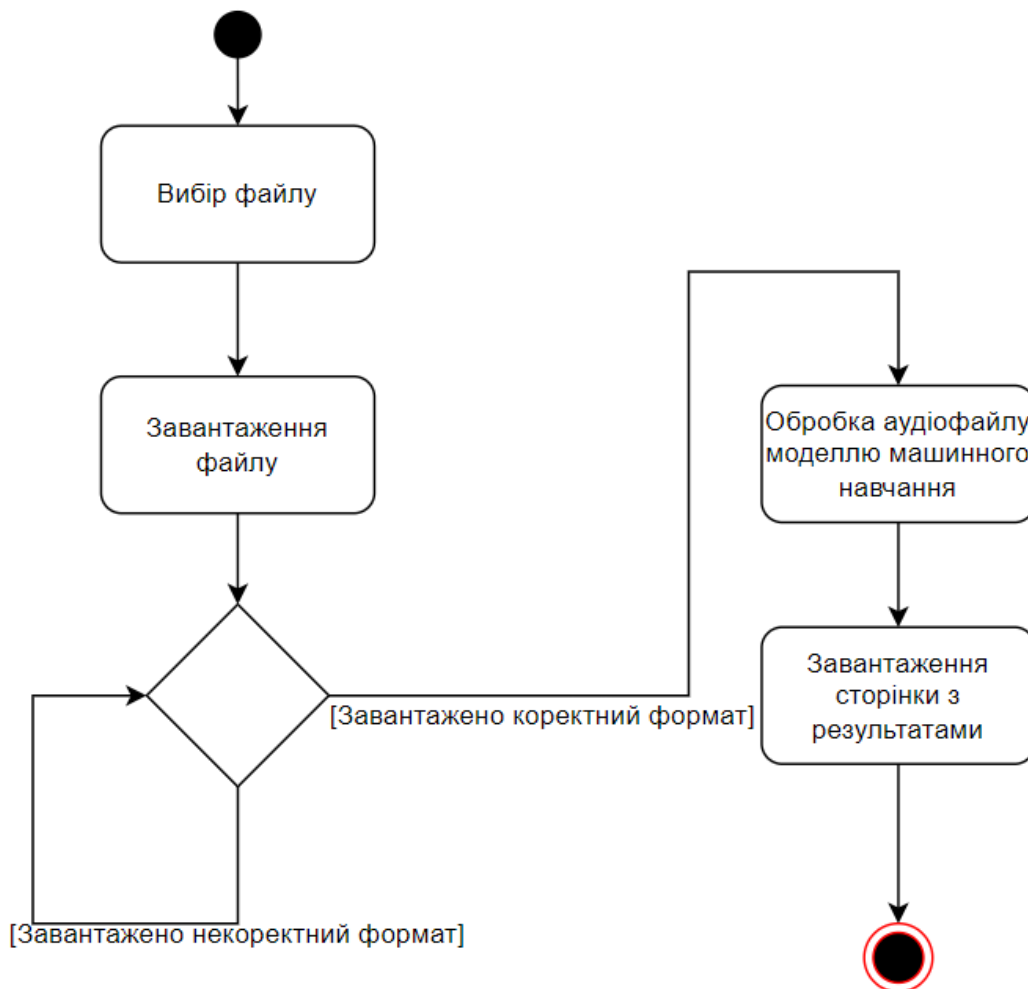


Рисунок 2.4 – UML діаграма діяльності

2.4 Розробка алгоритмів роботи системи

Для реалізації програмного застосунку була необхідність розробки та візуалізації загального алгоритму роботи розроблюваного програмного застосунку. Такий алгоритм було розроблено та продемонстровано на рисунку 2.5.

Після завантаження веб-сторінки користувачеві надається можливість обрати бажаний аудіофайл для подальшого аналізу. Якщо обраний файл відповідає встановленому формату (WAV), програмна частина веб-сайту розпочинає обробку

цього файлу. Для аналізу аудіофайлу застосунок розбиває його на окремі складові, такі як темп, ритм, хроматичні властивості тощо, для подальшого порівняння з характеристиками різних жанрів, які є у базі даних моделі.

Саме порівняння аудіофайлу з характеристиками різних жанрів є основним етапом аналізу. Після аналізу модель визначає, з яким жанром аудіофайлу спостерігається найбільше збігів і повертає цей жанр як результат класифікації. Після цього веб-сайт отримує відповідь від моделі та виводить результати аналізу користувачеві, який може переглянути прогнозований жанр музики для обраного аудіофайлу.

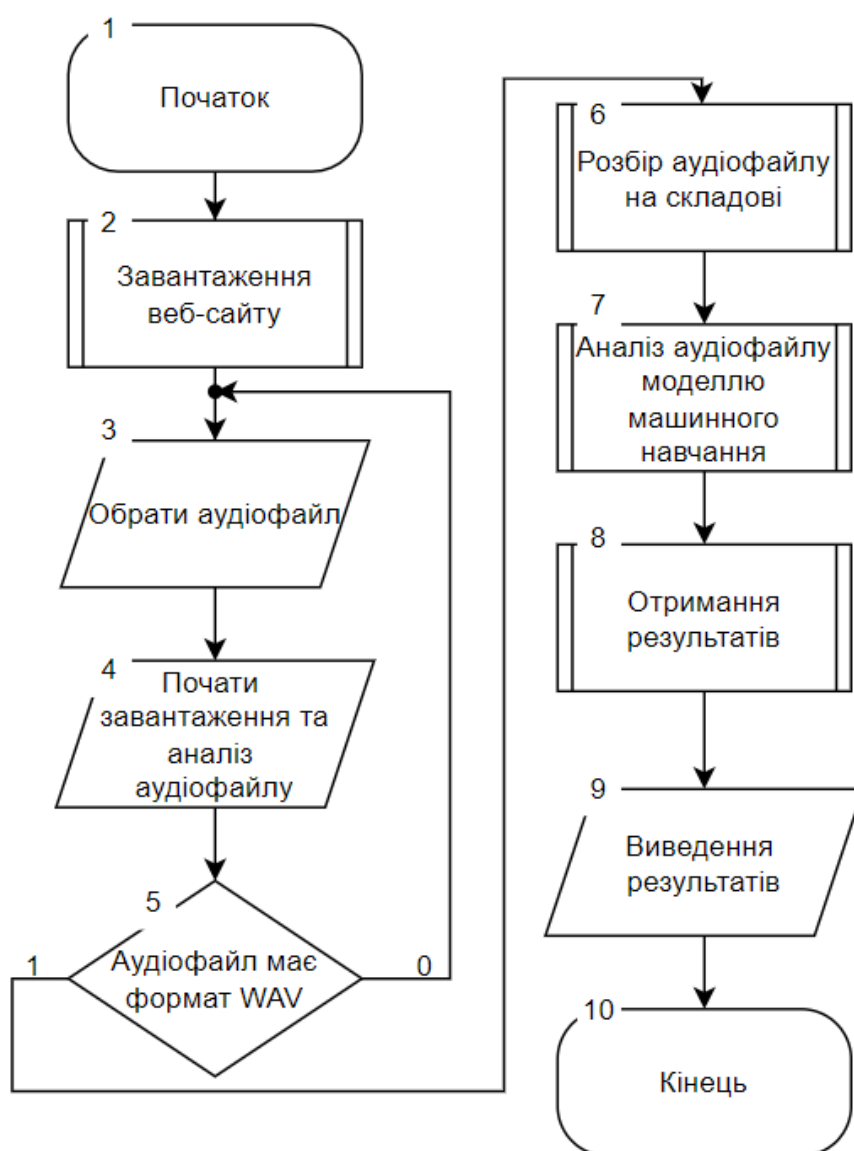


Рисунок 2.5 – Загальний алгоритм програмного застосунку

Для кращого розуміння модуля розбору файлу на складові було розроблено додатковий алгоритм, що продемонстровано на рисунку 2.6. Спочатку алгоритм отримує завантажений файл формату WAV та зчитує інформацію про звукові хвилі, що кодують аудіо. Далі зчитані дані перетворюються у числовий формат, що дозволяє представити звукові сигнали у вигляді числових значень. Далі алгоритм виконує екстракцію аудіоознак з отриманих числових значень. Це такі ознаки як частота, амплітуда, спектограма та інші. На основі цих даних алгоритм аналізує аудіофайл, порівнюючи із відомими програмі файлами.

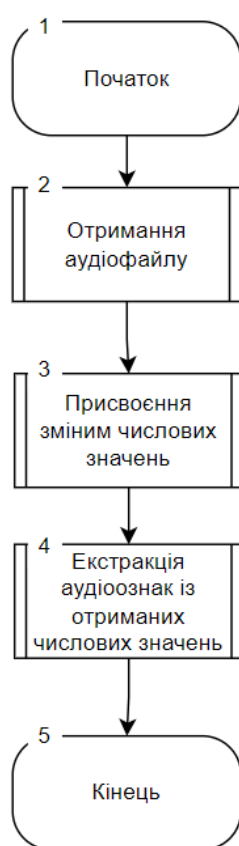


Рисунок 2.6 – Алгоритм роботи програмного модуля отримання аудіоознак

2.5 Висновки

Отже, у другому розділі було проаналізовано вхідні дані, розроблено інтерфейс та алгоритми роботи програми, які також були візуалізовані та продемонстровані на малюнках. Також було розроблено UML діаграму діяльності, що візуалізує діяльність програмного продукту.

3. ОГЛЯД ПРОГРАМНИХ МОДУЛІВ РЕАЛІЗАЦІЇ ВЕБ-ЗАСТОСУНКУ ДЛЯ КЛАСИФІКАЦІЇ МУЗИЧНИХ АУДІОФАЙЛІВ ЗА ЖАНРАМИ З ВИКОРИСТАННЯМ МАШИННОГО НАВЧАННЯ

3.1 Варіантний аналіз і обґрунтування вибору засобів для реалізації програмного забезпечення

Під час розробки веб-застосунку для класифікації музичних аудіофайлів за жанрами з використанням машинного навчання, вибір оптимальних технологій для кожного етапу розробки є ключовим аспектом успішної реалізації проекту. Оптимальні технології забезпечують ефективність, швидкість та надійність системи, а також полегшують процеси розробки та підтримки.

Для реалізації машинного навчання була можливість вибрати такі бібліотеки як scikit-learn, TensorFlow[16] або PyTorch[17]. Оскільки TensorFlow та PyTorch, хоч і використовуються для машинного навчання, але спеціалізуються на методах глибокого навчання, а scikit-learn надає великий вибір алгоритмів класифікації та інших методів машинного навчання, що найкраще підходить для виконання задачі проекту, то очевидно, що для реалізації машинного навчання в проекті, було обрано саме бібліотеку scikit-learn.

Для обробки аудіоданих було запропоновано до використання дві бібліотеки: Librosa та PyDub[18]. Проаналізувавши обидві було виявлено, що PyDub спеціалізується на таких операціях із аудіофайлами як обрізка, злиття чи інші, а Librosa надає інструменти для аналізу аудіоданих та екстрагування аудіоознак, що найкраще підходить для реалізації поставленої задачі.

Веб-розробка була виконана із використанням фреймворку Django. При виборі фреймворку для веб-розробки також було дві опції, однією із яких був фреймворк Flask[19], а іншою – обраний фреймворк. Flask надає базові інструменти для розробки веб-застосунків та може бути легко розширений за допомогою різних розширень. У виборі між фреймворками було зроблено акцент на особисті уміння розробника та його можливості.

У реалізації фронтенду, очевидно було використано HTML, CSS та JS. Для швидкої розробки стильного та адаптивного дизайну було використано Bootstrap. Для кращої роботи із документом HTML, спрощення реалізації обробки подій та анімацій було обрано бібліотеку jQuery.

Збір та початкова обробка даних для навчання моделі машинного навчання не виконувались. Дані для навчання моделі машинного навчання були використані із поширеного датасету – GTZAN Dataset. Датасет знаходиться у відкритому доступі та вільно поширюється між користувачами. У датасеті підготовано десять папок із різними жанрами, у кожній папці знаходиться по сотні підготованих та оброблених пісень, нарізаних по 30 секунд. Разом виходить тисяча пісень, що у часовому еквіваленті тривають близько восьми із половиною годин.

Отже, для машинного навчання було обрано бібліотеку Python – scikit-learn. Для програмної обробки аудіозаписів було використано бібліотеку Librosa. Веб-розробка виконувалась за допомогою фреймворка Django, а фронтенд розроблено засобами мов HTML, CSS із використанням фреймворку Bootstrap та JS із використанням бібліотеки jQuery. Дані для навчання моделі машинного навчання було взято із датапаку у відкритому доступі – GTZAN Dataset.

3.2 Огляд роботи моделі машинного навчання

У реалізації проєкту найпершою та найважливішою задачею, що вимагала програмної розробки було створення моделі машинного навчання. Для успішного навчання моделі необхідно грамотна робота із даними. Під час розробки проєкту було використано функцію, що розділяє дані на навчальний та тестовий набори.

Таке розділення даних допомагає переконатися, що модель вчиться на одних даних і оцінюється на зовсім інших. Це дозволяє уникнути перенавчання – ситуації, коли модель вивчає шум або деталі навчальних даних, що не підходять для узагальнення на нові дані.

Також розділення даних дозволяє оцінити, наскільки добре модель працює на нових, раніше невидимих даних. Тестовий набір даних використовується для

оцінки точності та інших метрик ефективності моделі. Розділення даних допомагає уникнути підвищення більш ніж потрібних очікувань щодо ефективності моделі.

Коли весь набір даних використовується для навчання, модель може показати дуже хороші результати, але це може бути результатом того, що модель була оптимізована саме під цей набір.

Наявність такого поділу даних на набори допомагає встановити, наскільки добре модель генералізується на нові, реальні дані, що можуть з'явитися в майбутньому. Це дає змогу прогнозувати ефективність моделі в реальних умовах застосування. Вказана функція рандомізує дані і розділяє їх у вказаних пропорціях між навчальним та тестовим наборами.

Для рандомізації розділення було використано конкретне зерно, що забезпечило відтворюваність результатів при кожному запуску.

У розробці моделі машинного навчання для класифікації даних про музичні аудіозаписи за жанрами використовувався алгоритм випадкового лісу, з бібліотеки `sklearn`.

У програмі створюється екземпляр класу випадкового лісу, який навчається на навчальних даних, зберігаючи навчену модель у змінній. Навчальні дані передаються у вигляді масивів. Після навчання моделі отримується важливість ознак, яка дозволяє зрозуміти, які атрибути дали найбільший внесок у класифікацію. Для цього використовується атрибут важливості ознак із класу випадкового лісу. Потім ознаки сортуються за їхньою важливістю, що допомагає відобразити їхню значущість у класифікаційному процесі. Таким чином модель розрізняє жанри та має можливість аналізувати аудіозаписи, звіряючи наявні ознаки у навчальному наборі із ознаками отриманими у результаті аналізу.

Також використовується клас із бібліотеки `sklearn`, що виконує масштабування ознак для навчального та тестового наборів даних, для забезпечення їх однакової шкали значень. Цей клас зводить усі значення у діапазон між одиницею та нулем, таким чином оптимізуючи та пришвидчуючи роботу програми.

У результаті отримуємо вихід даних у такому вигляді як це продемонстровано на рисунку 3.1.

÷	class_name ÷	tempo ÷	chroma_stft ÷	rmse ÷	spectral_centroid ÷	spectral_bandwidth ÷
0	blues	103.359375	0.380260	0.248262	2116.942959	1956.611056
1	blues	95.703125	0.306451	0.113475	1156.070496	1497.668176
2	blues	151.999081	0.253487	0.151571	1331.073970	1973.643437
3	blues	184.570312	0.269320	0.119072	1361.045467	1567.804596
4	blues	161.499023	0.391059	0.137728	1811.076084	2052.332563

Рисунок 3.1 – Приклад отриманих програмою даних

Модель машинного навчання оброблює дані у такому, зручному для неї вигляді та прикріплює отримані ознаки до відповідних жанрів, завдяки чому має змогу аналізувати зовсім нові аудіозаписи та давати результати.

Отже, було проаналізовано модель машинного навчання, що використовує бібліотеку sklearn для класифікації даних про музичні аудіозаписи. Також модель розділяє дані на два набори – тестовий та навчальний, для отримання точніших результатів у роботі моделі в майбутньому при зіткненні із невідомими їй аудіозаписами.

3.3 Огляд роботи програмного модуля для розділення музичного аудіозапису на складові та подальшого аналізу

Для ефективного використання навченої моделі машинного навчання необхідно провести конвертацію аудіосигналів з оброблюваного файлу у числові значення та витягнути з них аудіоознаки. Цей процес є критичним для правильного функціонування моделі та точності класифікації музичних аудіозаписів.

Під час цієї обробки, аудіосигнали перетворюються із аналогового аудіовигляду у вигляд числових значень, що можуть бути оброблені програмним забезпеченням. Після чого, проводиться екстрагування аудіоознак із цих числових даних, що включає в себе аналіз спектральних характеристик, таких як спектрограма та спектральні коефіцієнти, а також аналіз темпу, ритму та інших музичних характеристик.

Отримані аудіоознаки стають важливими вхідними даними для навченої моделі, яка використовує їх для класифікації музичних аудіозаписів за їхніми музичними характеристиками та жанрами. Детальна та точна конвертація аудіосигналів у відповідні аудіоознаки є важливим етапом підготовки даних перед використанням моделі машинного навчання.

Під час роботи програми та аналізу аудіозапису виконується перетворення аудіосигналів у числові значення та підготовка їх для подальшого використання у екстрагуванні аудіоознак. Ці аудіоознаки є ключовими для подальшої роботи моделі машинного навчання, оскільки вони дозволяють аналізувати та класифікувати аудіозаписи за їхніми музичними характеристиками.

Описана функція використовує бібліотеку Librosa для завантаження аудіофайлу і отримання аудіоданих та частоти дискретизації. Потім вона обчислює силу атаки звуку для аналізу темпу та ритму аудіозапису. Після визначення темпу та ритму функція розділяє аудіодані на гармонічні та перкусивні складові. Далі вона рахує середнє значення енергії аудіосигналу у кожному кадрі аудіозапису та обчислює кількість переходів через нуль у сигналі.

Крім того, функція обробляє музичний спектр, обчислюючи спектограму хроматичних властивостей звуку, визначаючи спектральний центроїд, розраховуючи спектральну ширину смуги, визначаючи спектральний ролофф (частоту, що визначає 1/10 енергії спектра) та обчислюючи коефіцієнти мел-частотних кепстральних коефіцієнтів аудіосигналу.

Після проведення всіх обчислень функція створює словник, де ключами є назви характеристик, а значеннями – їх значення, і повертає цей словник для подальшого аналізу та класифікації обраного аудіозапису.

Для взаємодії програми із звичайним користувачем було включено функцію, яка відповідає за процес завантаження аудіофайлу через веб-форму та його подальший аналіз та класифікацію з використанням моделі машинного навчання. Ця функція дозволяє користувачам легко та зручно класифікувати музичні аудіофайли за їхніми жанрами, використовуючи створену модель машинного навчання.

Функція перевіряє, чи було надіслано файл через форму та чи не є він порожнім. Якщо файл не був надісланий або він порожній, виводиться повідомлення про помилку, а користувач перенаправляється на головну сторінку.

Якщо файл успішно завантажений, функція створює екземпляр форми, який містить інформацію про завантажений файл. Після цього відбувається перевірка валідності форми. Якщо форма валідна, отримуються метадані аудіофайлу за допомогою функції, що оглядається, та здійснюється передбачення жанру аудіофайлу за допомогою функції передбачення жанру. Після отримання результату передбачення жанр передається у контекст та відображується на веб-сторінці.

Тепер варто оглянути функцію, що здійснює аналіз та передбачення жанру аудіозапису.

По-перше, вона завантажує збережену модель машинного навчання з файлу, розташованого в папці з моделями. Це файл, який містить навчену модель класифікатора, об'єкт нормалізації та інші необхідні дані для класифікації.

Після завантаження моделі функція витягує необхідні дані з цієї моделі, такі як сам класифікатор, об'єкт нормалізації та словник, який відображає зв'язок між передбаченими класами та їхніми іменами.

Далі функція перетворює передані метадані за допомогою об'єкту нормалізації і передає ці дані на вхід моделі. Після цього вона отримує передбачений клас для цих даних і використовує його для визначення відповідного жанру музики.

Тож, ця функція виконує ключові кроки: завантаження моделі, підготовку даних, передбачення класу та визначення жанру музики на основі цього класу.

Отже, у результаті було розроблено програмний модуль, який розділяє аудіофайли на складові, конвертує аудіосигнали у числові значення, нормалізує отримані числові значення, працює з користувачем, контролюючи валідність завантаженого файлу та передає валідний аудіофайл на обробку моделі машинного навчання, яка визначає жанр, до якого найбільш споріднений завантажений аудіофайл.

3.4 Висновки

Під час розробки веб-застосунку для класифікації музичних аудіофайлів за жанрами з використанням машинного навчання було обрано оптимальні технології та інструменти. Розроблена модель машинного навчання на базі бібліотеки `scikit-learn`, яка дозволяє класифікувати музичні аудіофайли за жанрами на основі їхніх характеристик. Бібліотека `Librosa` була використана для обробки аудіофайлів та отримання їхніх характеристик, таких як темп, ритм, спектральні характеристики та інші. Отримані дані були використані як вхідні параметри для моделі машинного навчання.

У результаті отримано веб-застосунок для класифікації музичних аудіофайлів за жанрами з використанням машинного навчання.

4 ТЕСТУВАННЯ ПРОГРАМИ

4.1 Тестування і валідація системи

Тестування системи[20] є важливою складовою розробки програмного забезпечення, оскільки дозволяє перевірити її працездатність, надійність та відповідність функціональним та нефункціональним вимогам. Оскільки, було розроблено веб-застосунок, додаток є кросплатформним[21] – здатним працювати на базі різних систем. Тестування проводилось на базі операційної системи Windows 11, використовуючи браузер Opera GX.

Першим ділом потрібно визначити чим саме являються представлені десять жанрів. Отож:

- поп – жанр характеризується простотою, цікавою інструментальною частиною і почуттям ритму, при цьому вокалу приділяється далеко не найосновніша увага. Головною і практично єдиною формою музичних композицій є пісня;
- диско – жанр танцювальної музики, якому властиві велика кількість ведучих ефектів, ритм-секції ударних і баса, а так само другорядне, фонове звучання струнних і духових інструментів;
- рок – жанр музики характеризується ритмічними відчуттями, які пов'язані з певним рухом. Деякі ознаки рок-композицій відносяться до вторинних, через що багато стилів музики помилково відносять до року;
- хіп-хоп – ритмічна й римована промова, яка спочатку виконується без супроводу, а пізніше виконується під ритмічний супровід;
- джаз – стиль музики поєднує в собі африканську і європейську культури. Відмінними рисами цього напрямку є імпровізація, витончений ритм і унікальні прийоми ритмічних фактур;
- класика – академічна музика, що наслідує передусім жанри (наприклад, соната, симфонія, опера і т. д.), принципи композиції (зокрема гармонії, форми, фактури тощо), інструментарій (переважно — інструменти

симфонічного оркестру), а також традиції виконання, що формувалися в європейській музиці в XVII — поч. XX ст.;

- реггі – відрізняється танцювальним характером. Характерною рисою реггі є відсутність виключно ритмічних або виключно мелодичних інструментальних партій (музиканти грають ритм та мелодію одночасно). Інші особливості: це бубнова партія, гітара, що грає на парних долях та мінливий бас;
- кантрі – оперує простим інструментарієм, окрім ударної установки характерне звучання надає гітара зі сталевими струнами (різновид гавайської гітари), губна гармоніка, нерідко використовуються також банджо, скрипка та піаніно хонкі-тонк, спеціально дещо розстроєне;
- метал – цьому жанру притаманні агресивні ритми на ударних та підсилені дисторшном гітари зі вставками соло.

Для початку було вирішено перевірити функціональність розробленої системи. Щоб це перевірити було підготовано музичні аудіофайли десяти різних жанрів, що підтримуються у системі, щоб точно перевірити її можливості. На рисунках 4.1-4.20 представлено результати визначення жанру реалізованою системою та їх спектральні аналізи.

Першою на перевірку була запущена пісня жанру блюз, виконавця Buckwheat Zydeco – Drivin Old Grey. Результати класифікації показано на рисунку 4.1.

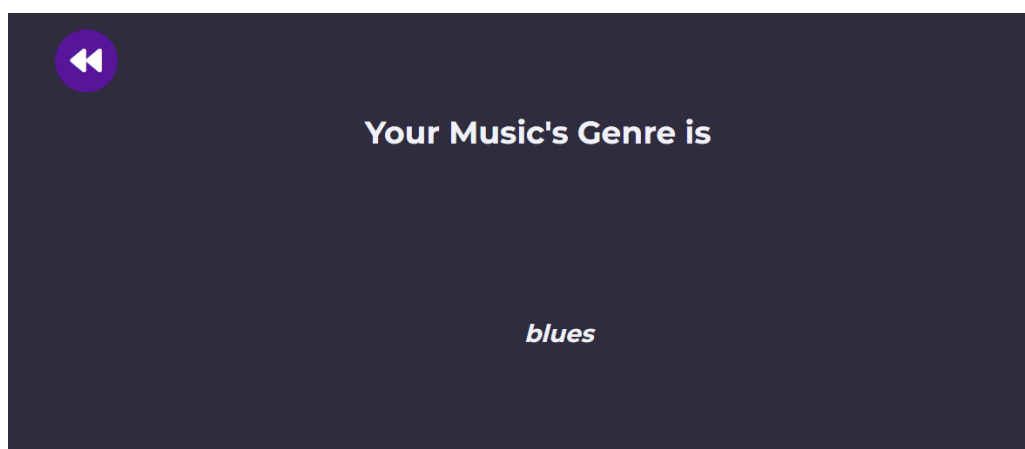


Рисунок 4.1 – Результат класифікації музичного аудіофайлу жанру блюз

Спектрограму композиції продемонстровано на рисунку 4.2.

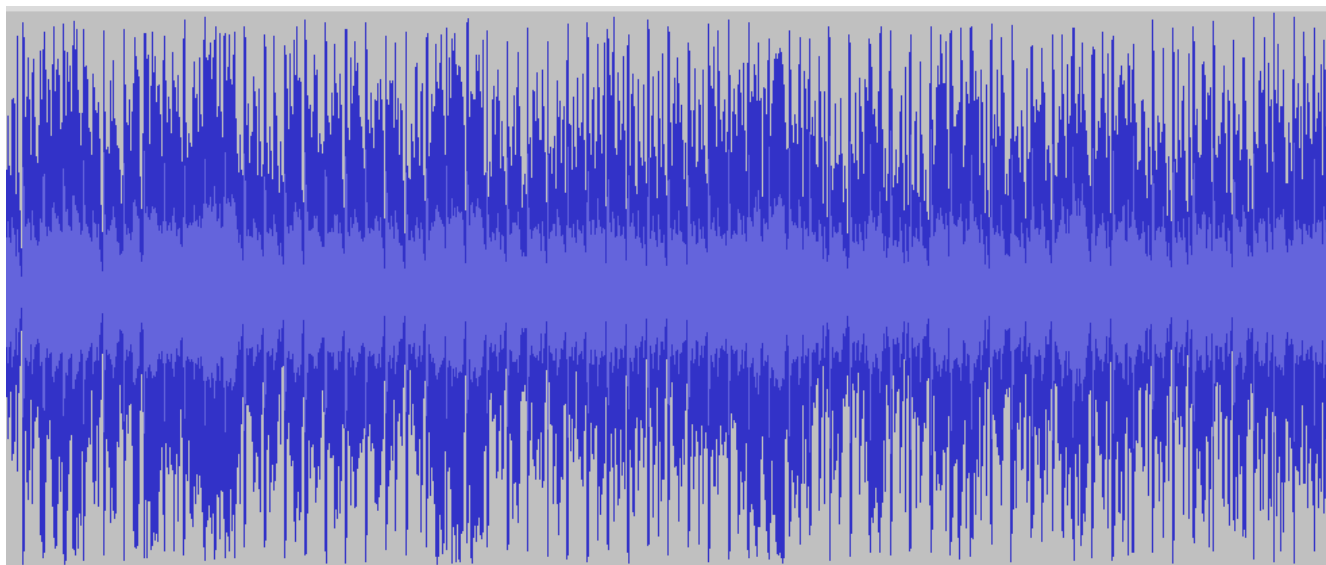


Рисунок 4.2 – Спектрограма композиції жанру блюз

Далі для системи було поставлено завдання визначити жанр класичної композиції Леонарда Берштейна – одного із епізодів мюзиклу On the Town(“На місто”). Результати класифікації було продемонстровано на рисунку 4.3.

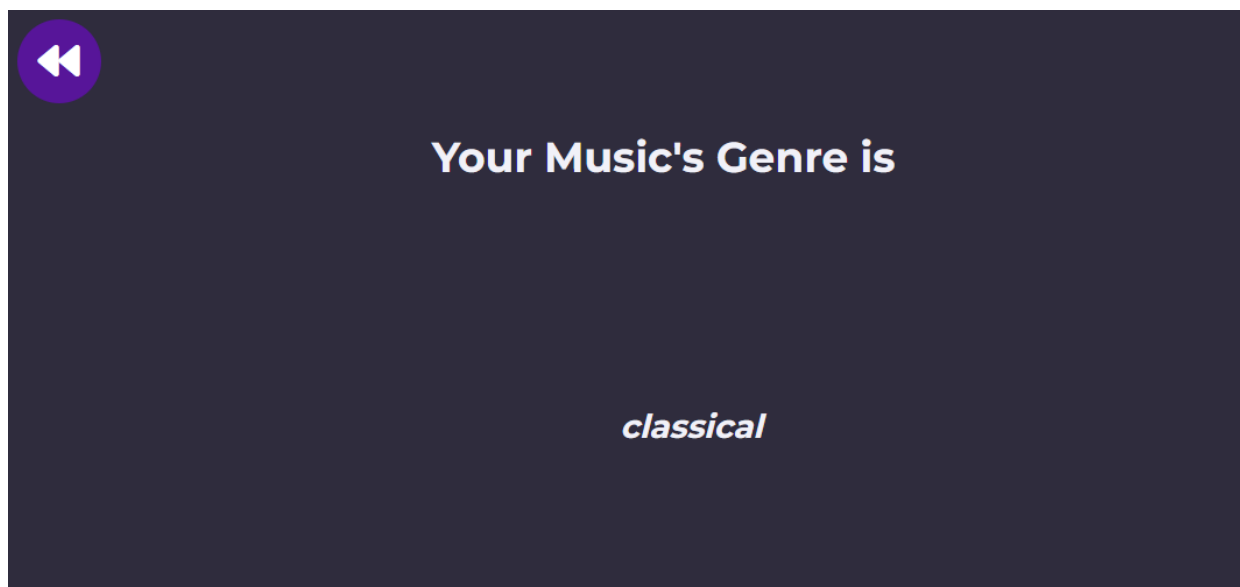


Рисунок 4.3 – Результат класифікації музичного аудіофайлу жанру класика

Спектрограма композиції жанру класика показує значні відмінності від спектрограми композиції жанру блюз. Отже, вищезгаданий аналіз композиції жанру класика представлений на рисунку 4.4.

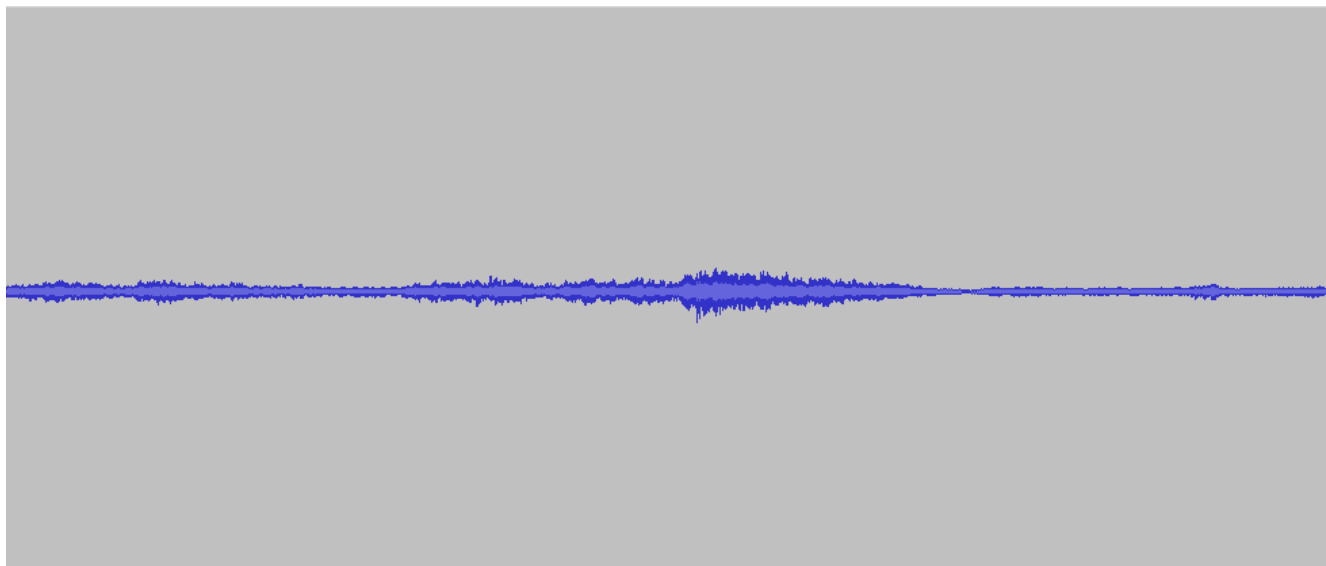


Рисунок 4.4 – Спектрограма композиції жанру класика

Таким самим чином було протестовано систему на кантрі пісні Джоні Кеша – The Man Come's Around. Результат показано на рисунку 4.5.

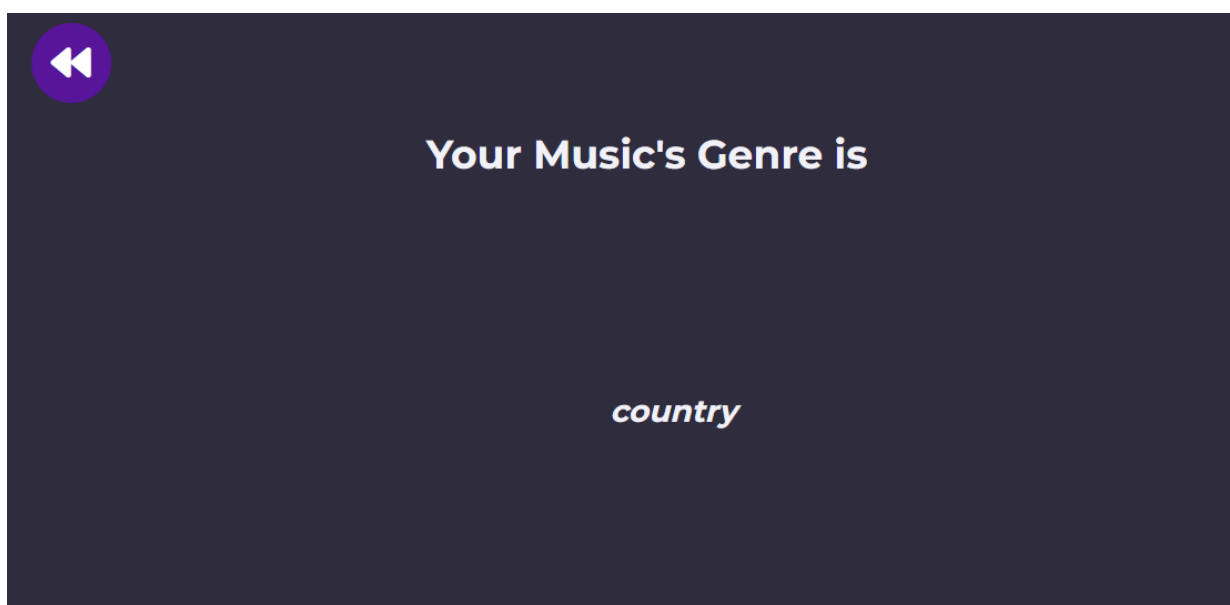


Рисунок 4.5 – Результат класифікації музичного аудіофайлу жанру кантрі

На рисунку 4.6 спостерігаємо спектрограму композиції жанру кантрі.

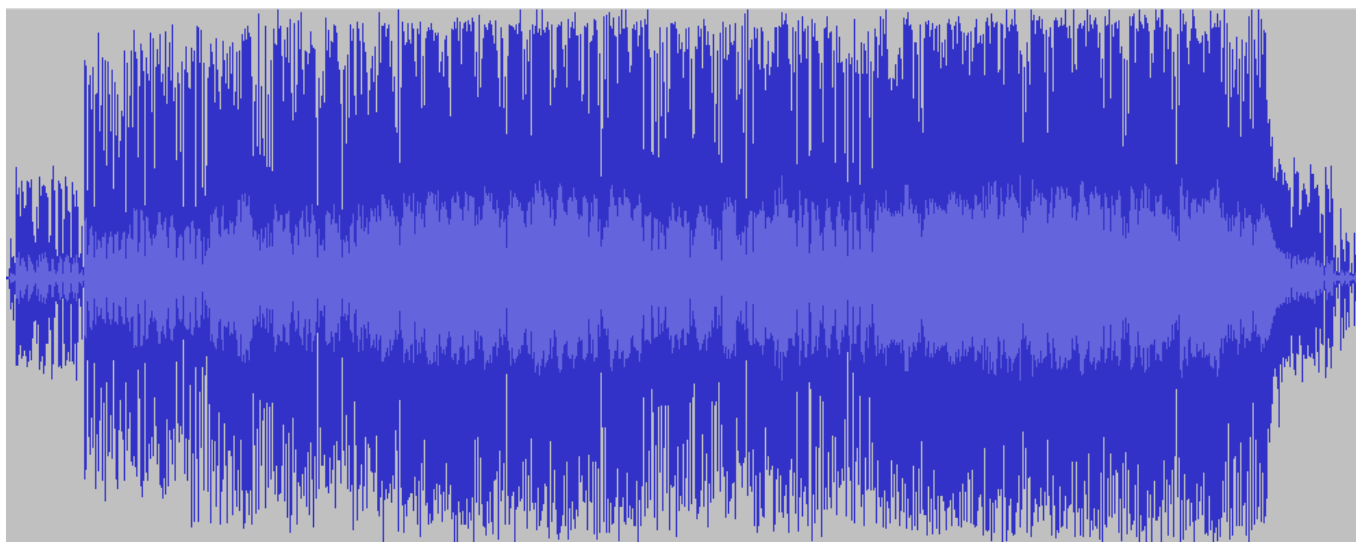


Рисунок 4.6 – Спектрограма композиції жанру кантрі

Для тестування класифікації жанру диско було обрано пісню I Will Survive виконавиці Глорії Гейнор. Результат показано на рисунку 4.7.

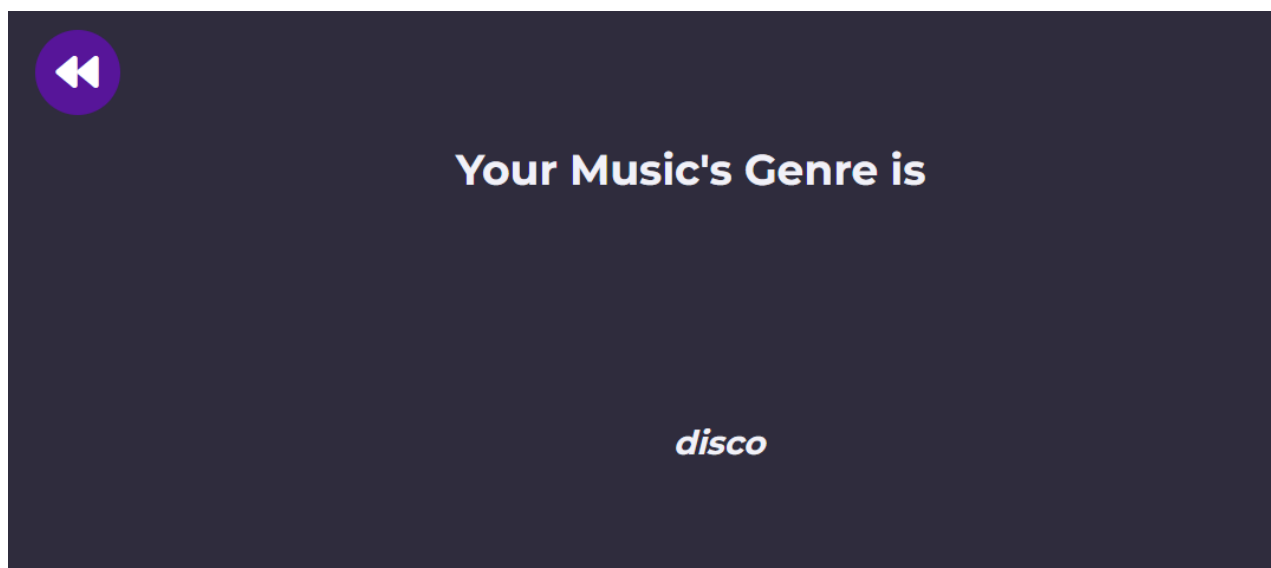


Рисунок 4.7 – Результат класифікації музичного аудіофайлу жанру диско

Оскільки диско – танцювальна музика її спектрограма вирізняється ритмічністю та швидким темпом, в той час, як спектрограма жанру кантрі більш хаотичний.

Спектрограму композиції жанру диско було показано на рисунку 4.8.

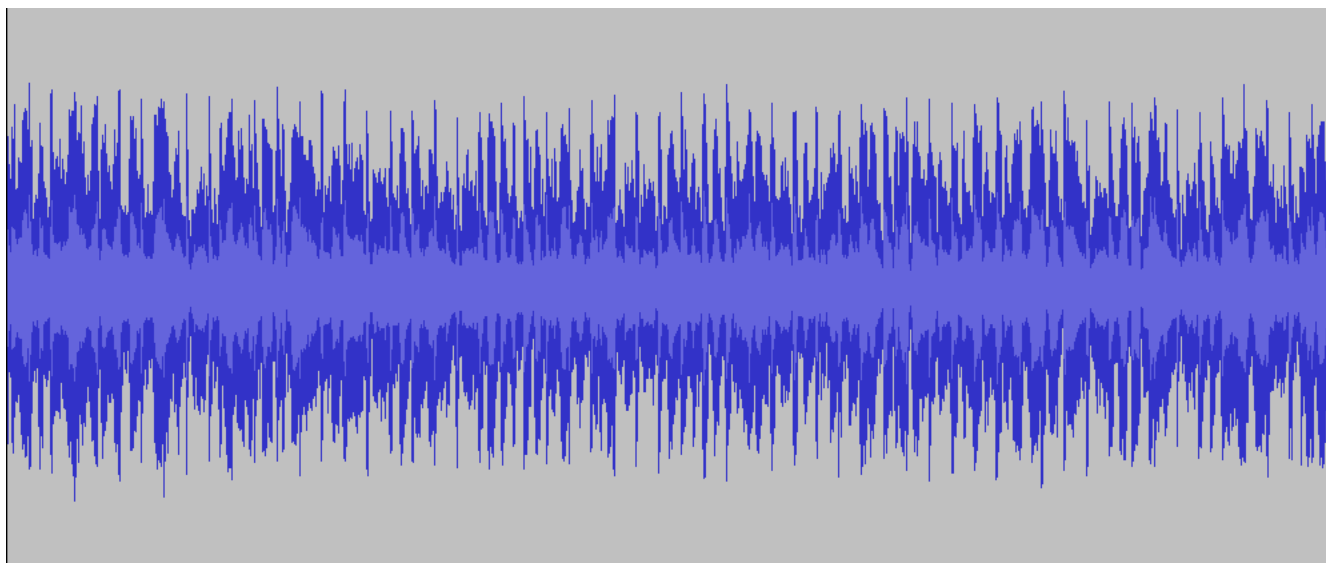


Рисунок 4.8 – Спектрограма композиції жанру диско

Для тестування уміння моделі розпізнавати жанр хіп-хоп було обрано композицію Hey Ladies гурту Beastie Boys. Результат продемонстровано на рисунку 4.9.

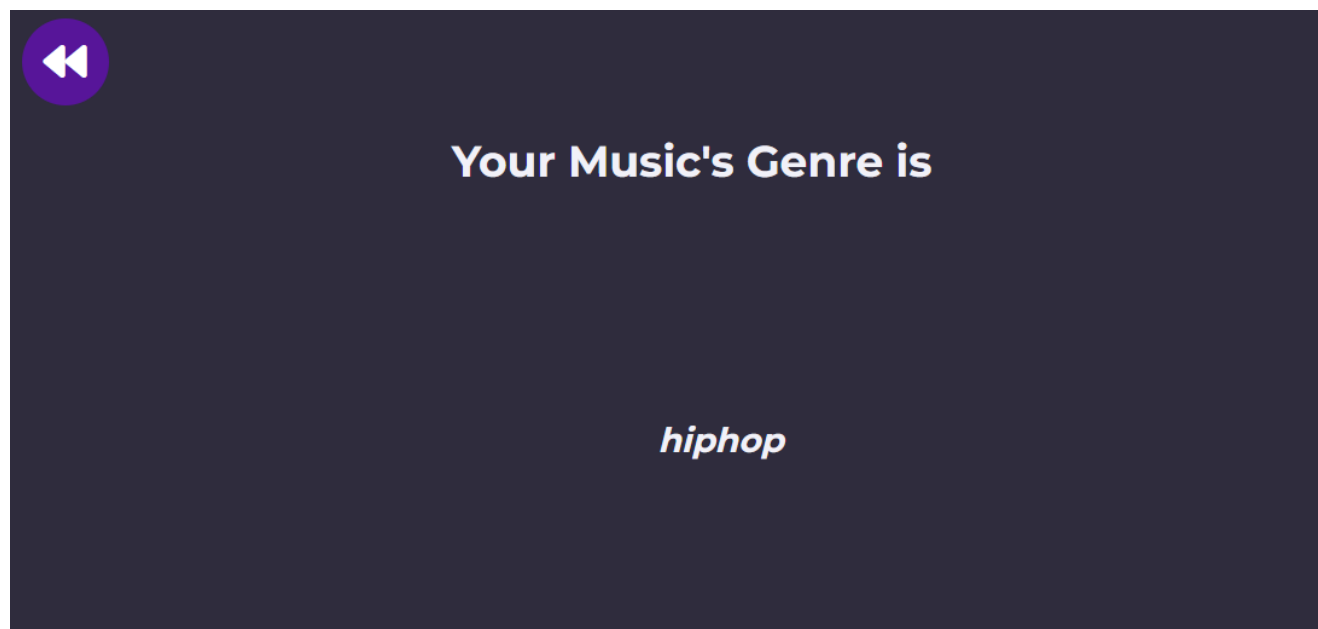


Рисунок 4.9 – Результат класифікації музичного аудіофайлу жанру хіп-хоп

Спектрограму жанру хіп-хоп показано на рисунку 4.10. Він виказує жанр вираженим повторенням певних фрагментів.

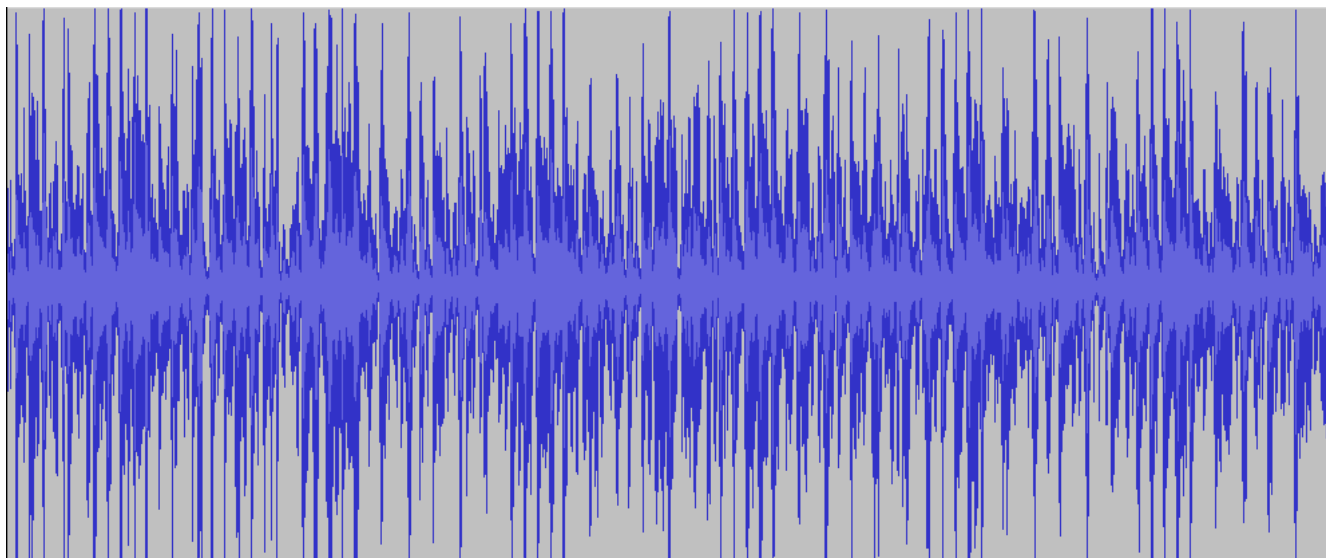


Рисунок 4.10 – Спектрограма композиції жанру хіп-хоп

Для перевірки розпізнавання жанру джаз було обрано композицію Xavier's Lair тріо Branford Marsalis. Результат отримано та продемонстровано на рисунку 4.11.

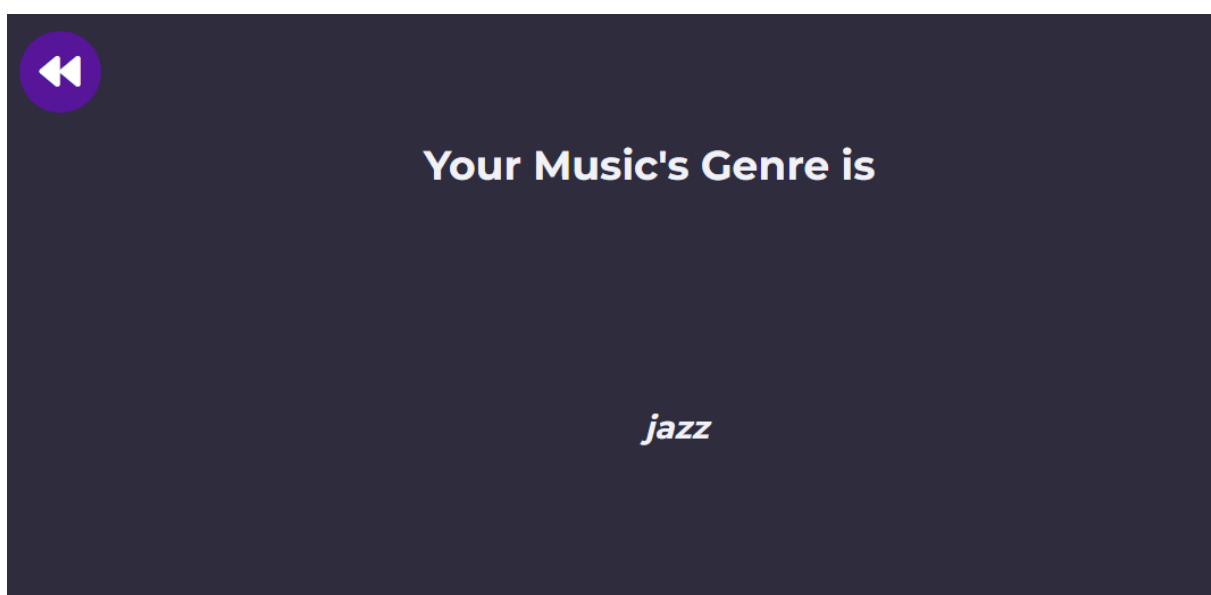


Рисунок 4.11 – Результати класифікації музичного аудіофайлу жанру джаз

Спектрограму жанру джаз продемонстровано на рисунку 4.12.

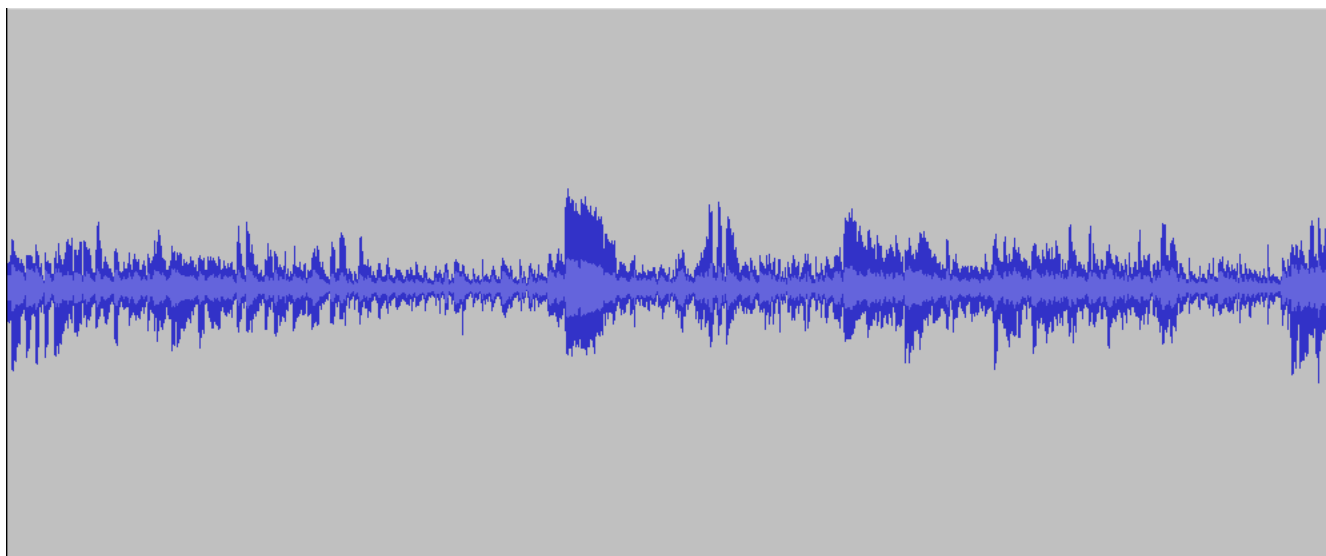


Рисунок 4.12 – Спектрограма композиції жанру джаз

Тестування моделі в її умінні класифікувати жарт метал було перевірено легендарною композицією названих батьків жанру метал – Black Sabbath – Paranoid. Модель успішно справилась із завданням. Результати показані на рисунку 4.13.

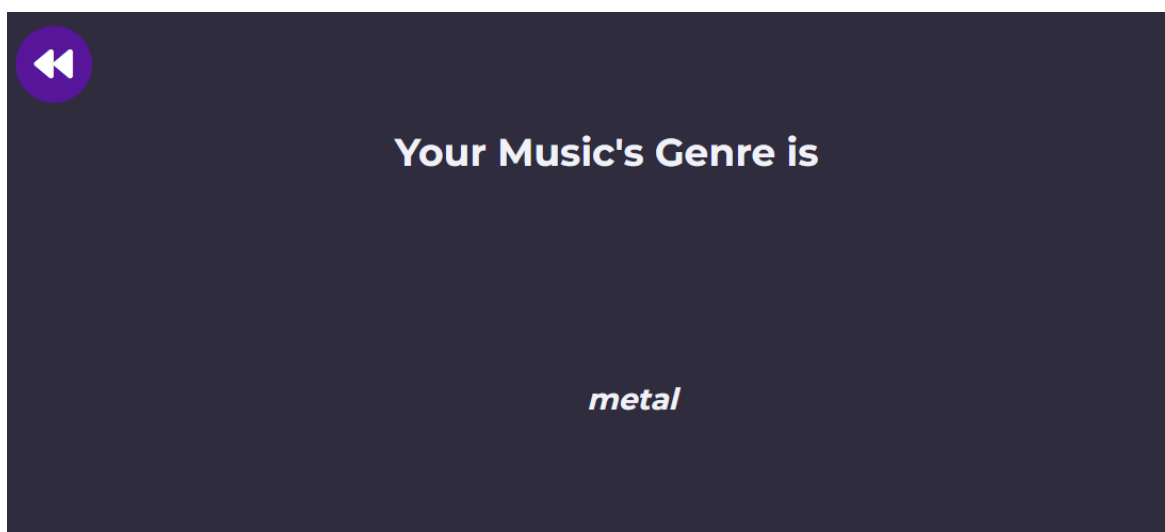


Рисунок 4.13 – Результат класифікації музичного аудіофайлу жанру метал

Спектрограма жанру метал показує агресивність та ритмічність, що притаманно жанру. Спектрограму зображено на рисунку 4.14.

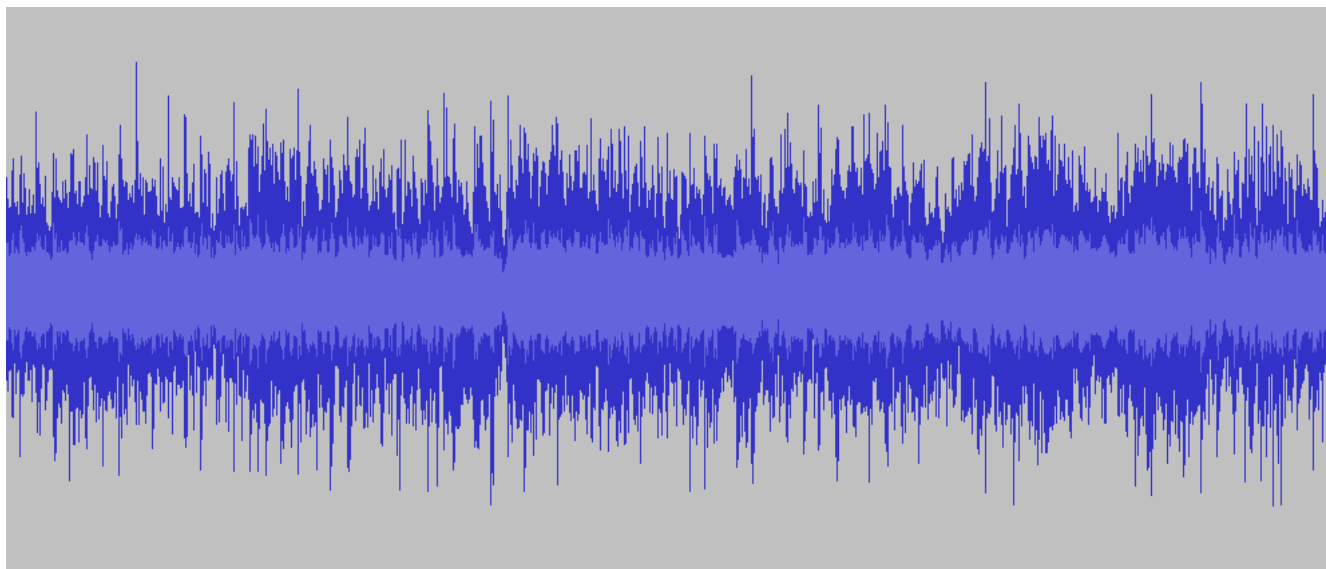


Рисунок 4.14 – Спектрограма композиції жанру метал

Для тестування розпізнавання жанру поп у піснях було завантажено аудіофайл Autumn Goodbye співачки Брітні Спірс. Результати тестування показано на рисунку 4.15.

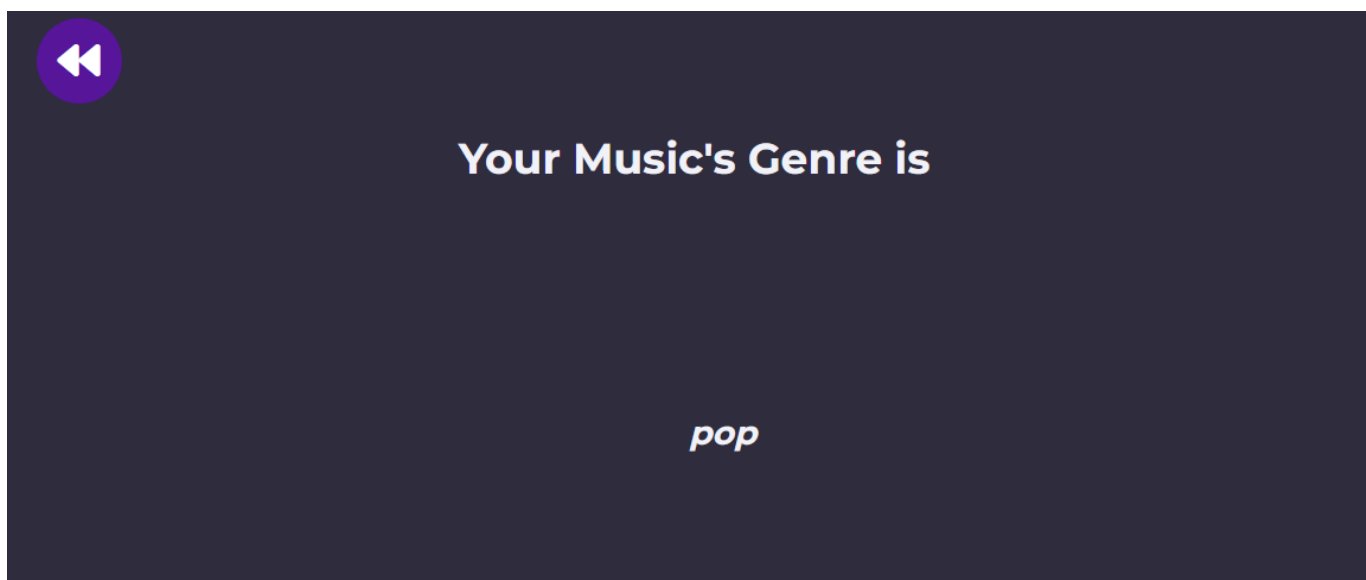


Рисунок 4.15 – Результат класифікації музичного аудіофайлу жанру поп

Спектрограму композиції жанру поп продемонстровано на рисунку 4.16.

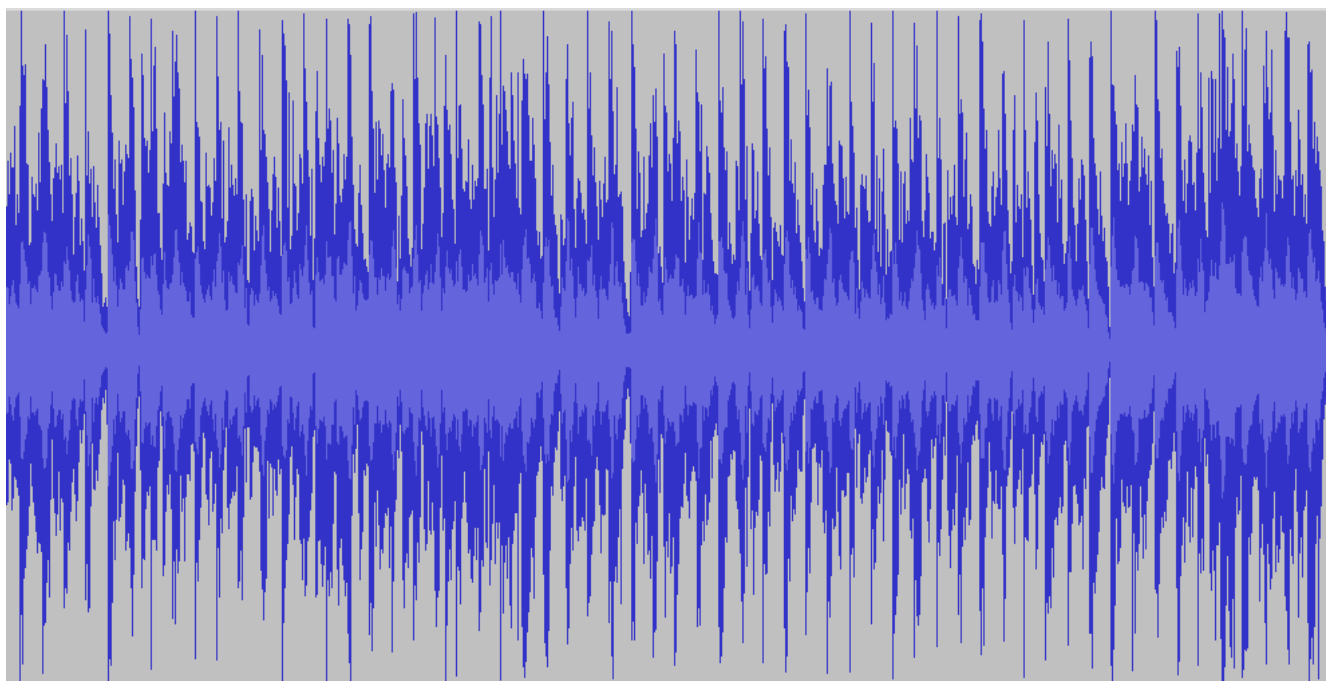


Рисунок 4.16 – Спектрограма композиції жанру поп

Жанр реггі було протестовано на пісні Макса Ромеро – Chase The Devil. Результати показано на рисунку 4.17.

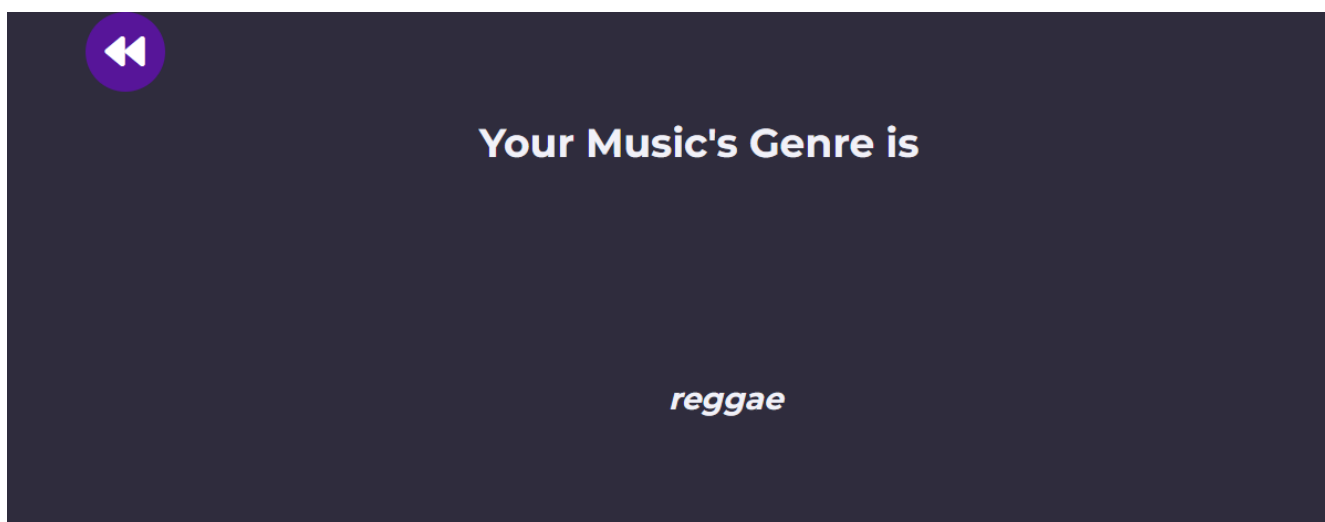


Рисунок 4.17 – Результат класифікації музичного аудіофайлу жанру реггі

Спектрограма композиції жанру реггі більше схожа на виконаний аналіз композиції жанру кантрі своєю хаотичністю, її було продемонстровано на рисунку 4.18.

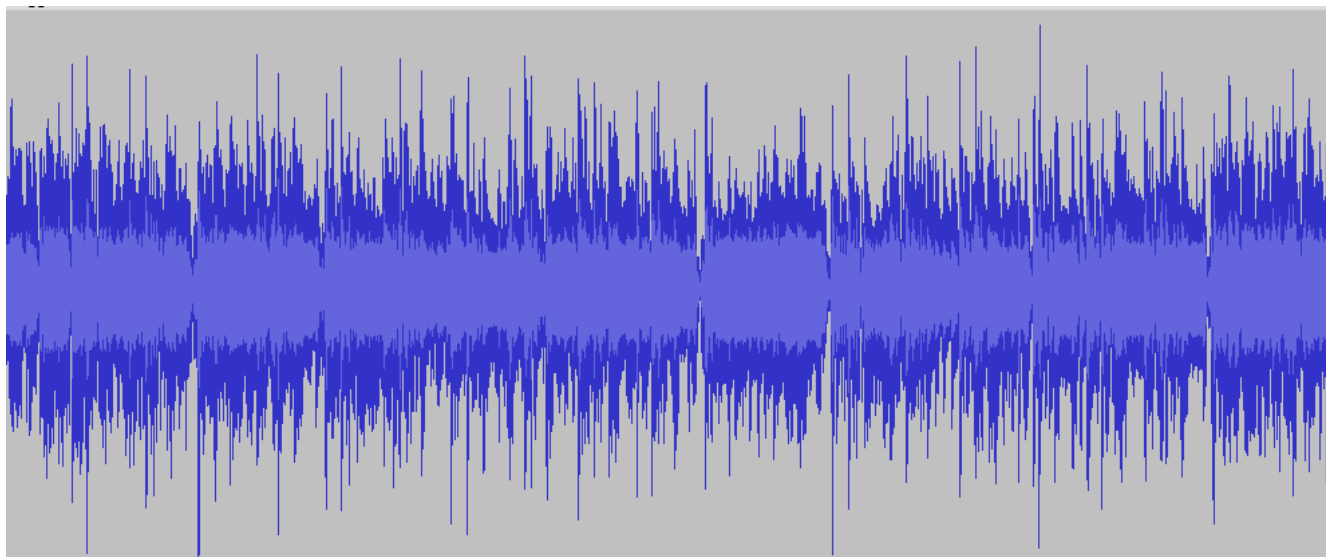


Рисунок 4.18 – Спектрограма композиції жанру реггі

Жанр рок протестовано на одній із культових піснi одного із культових гуртiв Stairway to heaven гурту Led Zeppelin. У результатi було отримано правильну класифiкацiю, яку продемонстровано на рисунку 4.19.

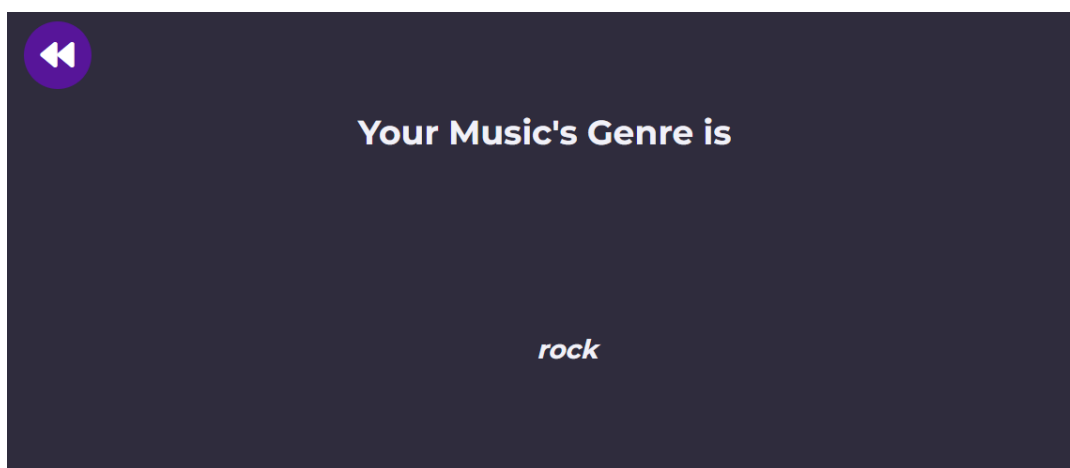


Рисунок 4.19 – Результат класифікації музичного аудіозапису жанру рок

Спектрограма саме цієї композиції вважається легендарною, бо, по чуткам, якщо її віддзеркалити – можна побачити послання сатани.

Тим не менше – спектрограму даної композиції показано на рисунку 4.20.

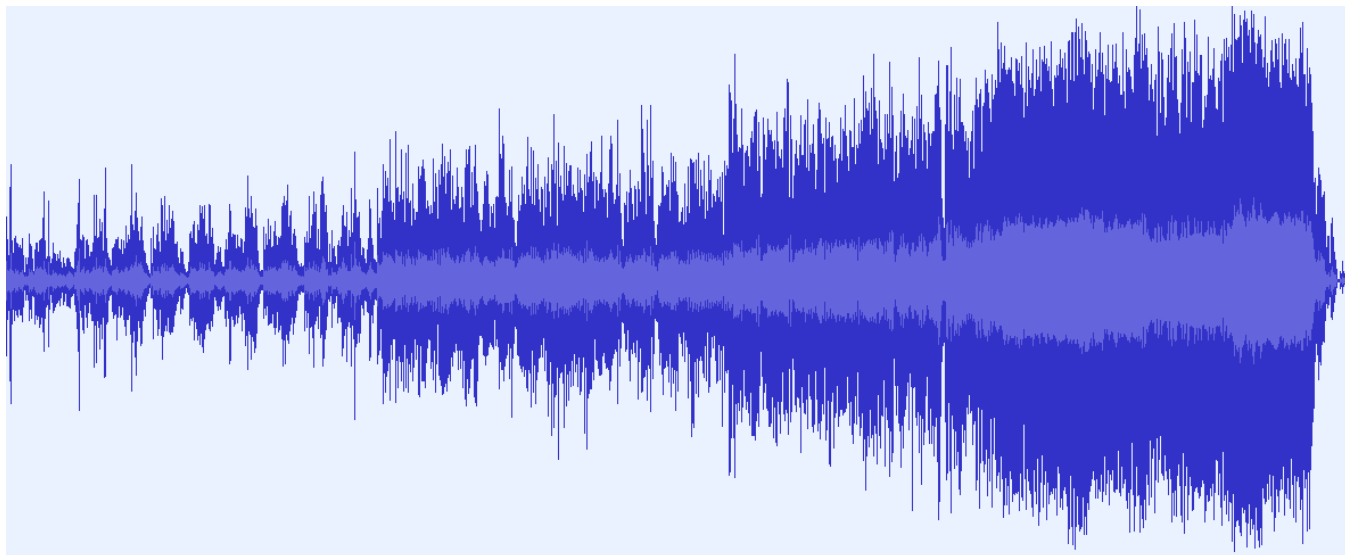


Рисунок 4.20 – Спектрограма композиції жанру рок

Система успішно виконує основне завдання - визначення жанру музичних аудіозаписів за допомогою алгоритмів машинного навчання. Тестування показало, що додаток правильно класифікує жанри в більшості випадків. Проте, деякі жанри, які мають схожі звукові характеристики, інколи визначаються з меншою точністю.

Система демонструє високу продуктивність, обробляючи аудіофайли середнього розміру (3-5 хвилин) за декілька секунд.

Інтерфейс додатку інтуїтивно зрозумілий та зручний для користувача. Процес завантаження аудіофайлу та отримання результату класифікації виконується у кілька простих кроків. Тестування користувачами показало позитивні відгуки щодо зручності використання додатку.

Після тестування уміння розробленою системи класифікувати аудіозаписи було вирішено перевірити результати при помилкових введеннях даних.

Після проведення тестування необхідно також провести валідацію системи. Першим ділом було перевірено реакцію системи на натиснення кнопки «Check» без завантаженого файлу.

Результат продемонстровано на рисунку 4.21.

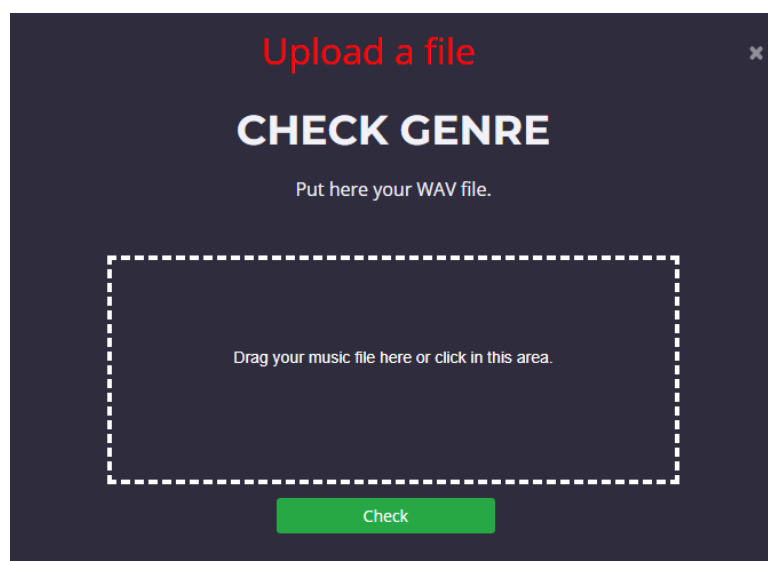


Рисунок 4.21 – Вигляд головної сторінки після натискання кнопки без завантаження аудіофайлу

Наступним було перевірено результат при завантаженні файлу невідповідного формату. Для тестування було використано формат .jpeg. Результат роботи системи продемонстровано на рисунку 4.22.

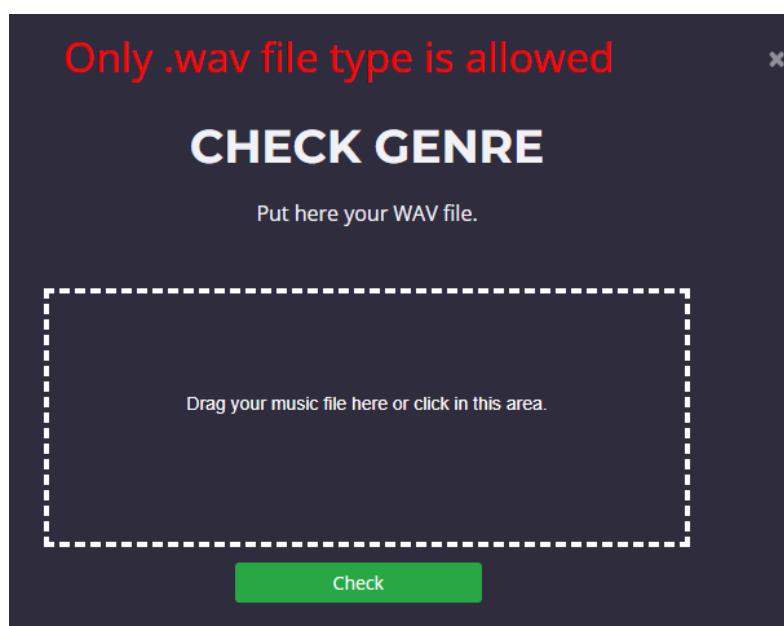


Рисунок 4.22 – Результат роботи системи при завантаженні файлу невідповідного формату

Хоча користувачу надано можливість завантажувати зразу декілька файлів, система підтримує обробку лише одного файлу за раз. У випадку завантаження декількох файлів за раз буде проаналізовано та виведено лише останній файл. Проведені тести продемонстровані на рисунках 4.23 та 4.24.

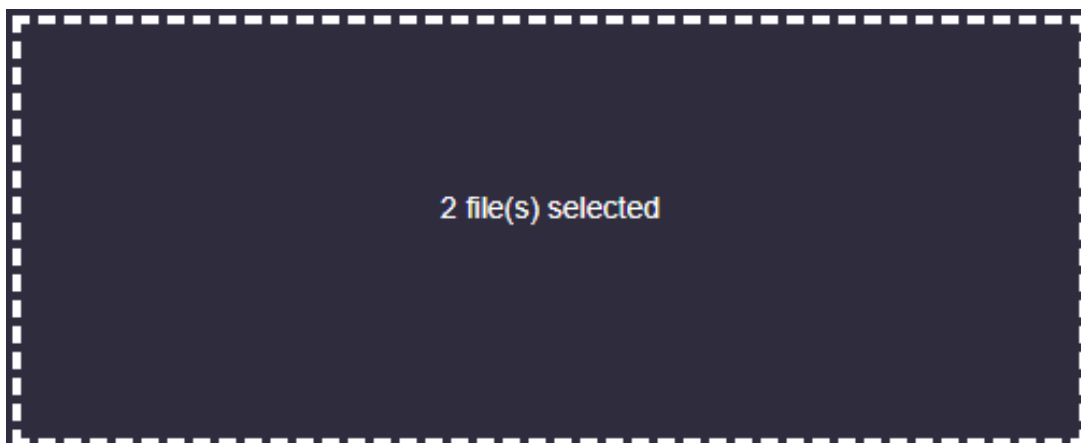


Рисунок 4.23 – Можливість завантаження декількох файлів

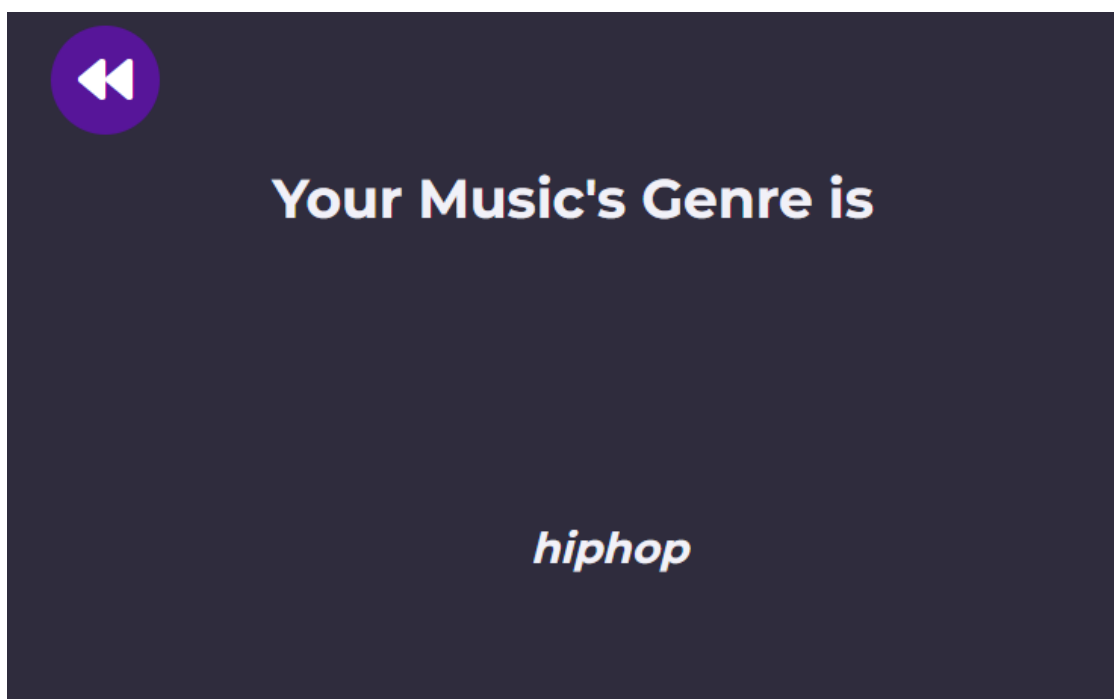


Рисунок 4.24 – Результат класифікації музичних аудіофайлів при виборі кількох варіантів

Наступні тести пов'язані із справжньою ціллю проєкту – визначення жанру власноруч записаних музичних аудіофайлів.

Почато було із власноруч записаних акустичної гітари та співу. Це кавер на відомого в Україні виконавця – Mistmorn'a, пісня – Палає цей світ у вогні. Результат аналізу моделі було показано на рисунку 4.25.

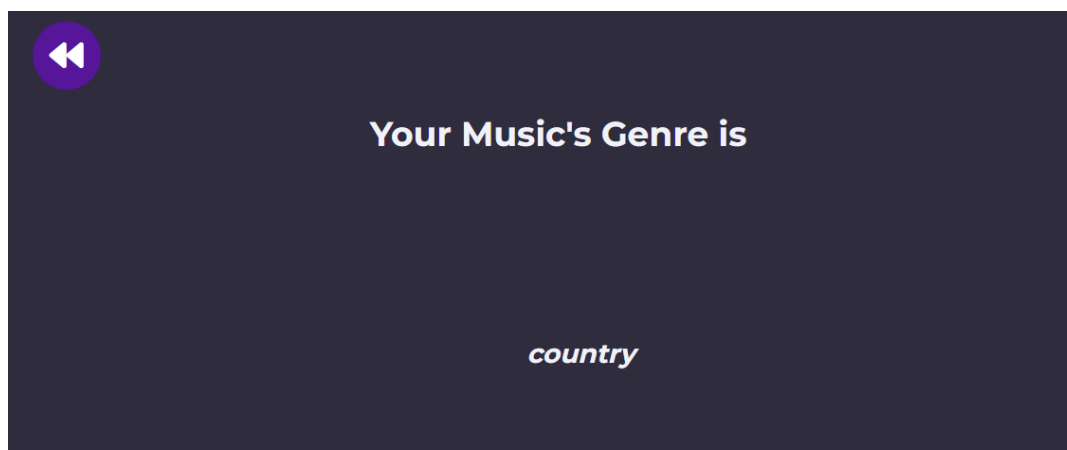


Рисунок 4.25 – Результат класифікації власного музичного аудіозапису Палає Світ Цей У Вогні

Бачимо, що система вирішила класифікувати цей аудіозапис як кантрі, воно й не дивно – кантрі зазвичай звучить з акустичної гітари та банджо. Також спектрограма композиції доводить, що рішення моделі – правильне, адже спектрограма схожа на аналіз композиції жанру кантрі.

Спектральний аналіз цієї композиції було продемонстровано на рисунку 4.26.

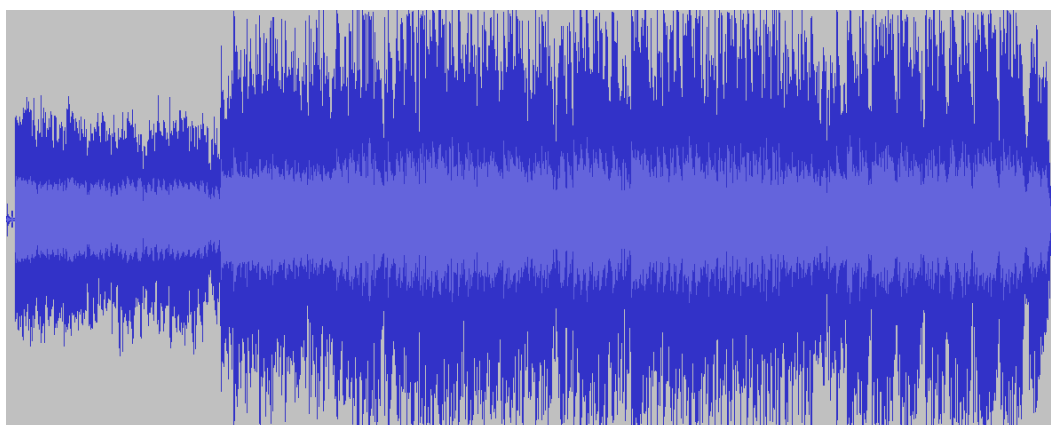


Рисунок 4.26 – Спектрограма композиції Палає Світ Цей У Вогні

Наступною було перевірено пісню *Cowboys From Hell*, оригінально записану гуртом *Pantera*, але виконану іншим виконавцем за допомогою електрогітари та басу.

Результат аналізу моделі класифікації продемонстровано на рисунку 4.27.

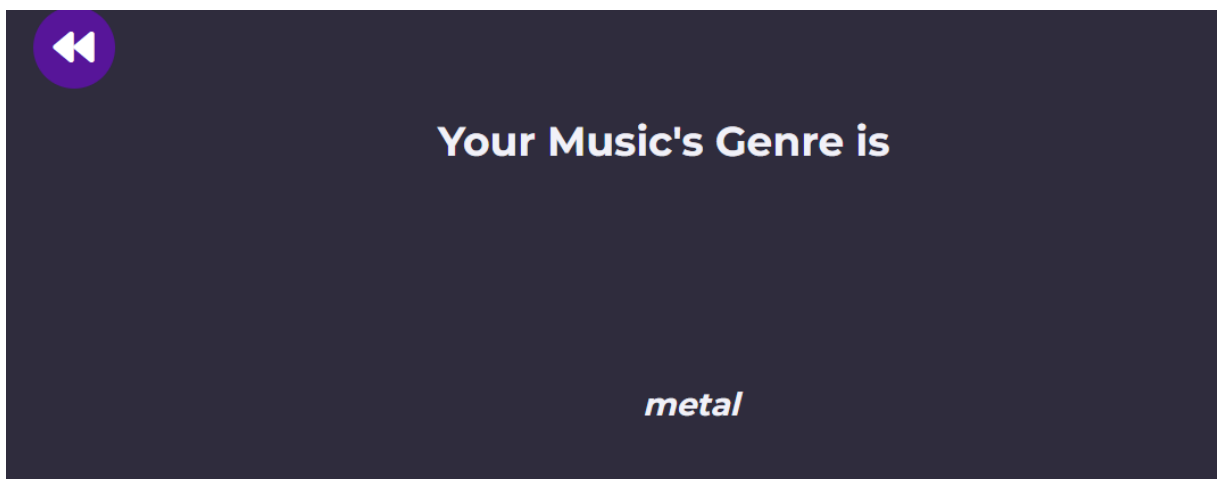


Рисунок 4.27 – Результат класифікації музичного аудіофайлу пісні *Cowboys From Hell* від іншого виконавця

Спостерігаємо, що система відреагувала на появу агресивних рифів та вокалу у аудіофайлі, що аналізується та визначила його жанр як метал. Також проаналізуємо дані спектрограми, яку показано на рисунку 4.28.

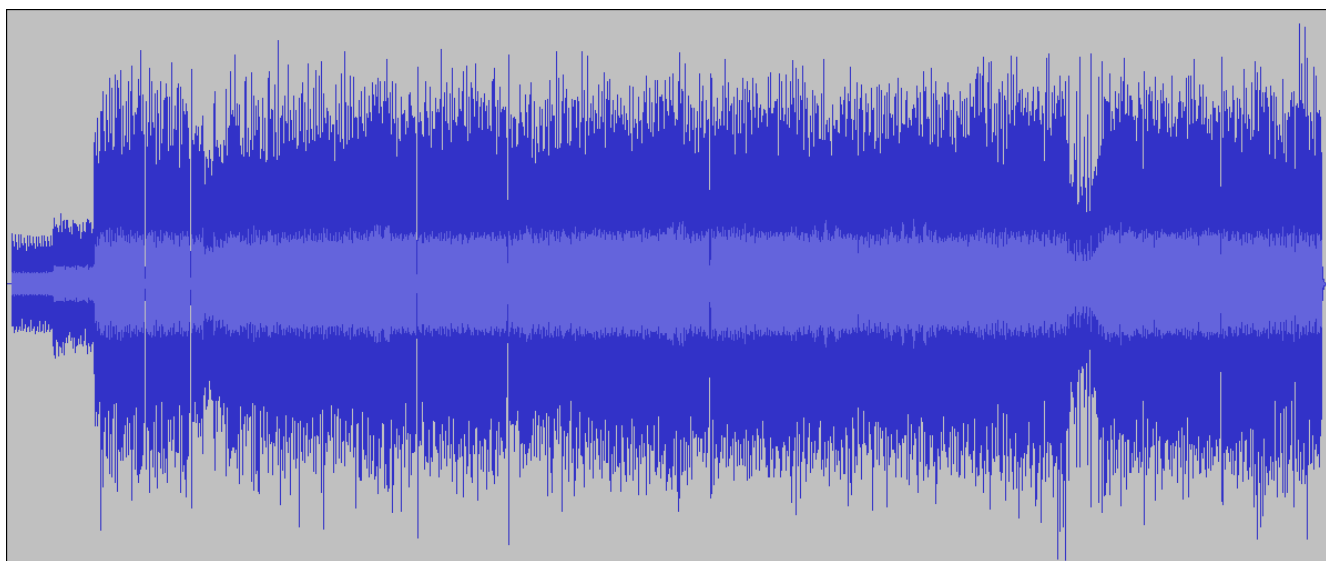


Рисунок 4.28 – Спектрограма композиції *Cowboys From Hell* від іншого виконавця

Останнім із каверів на відомі пісні було проаналізовано пісню Сльози Осені, оригінально написано та заспівано гуртом М8Л8ТХ. У результатах очікується отримання жанру метал, оскільки сама пісня відноситься до жанру блек-метал. Результати продемонстровано на рисунку 4.29.

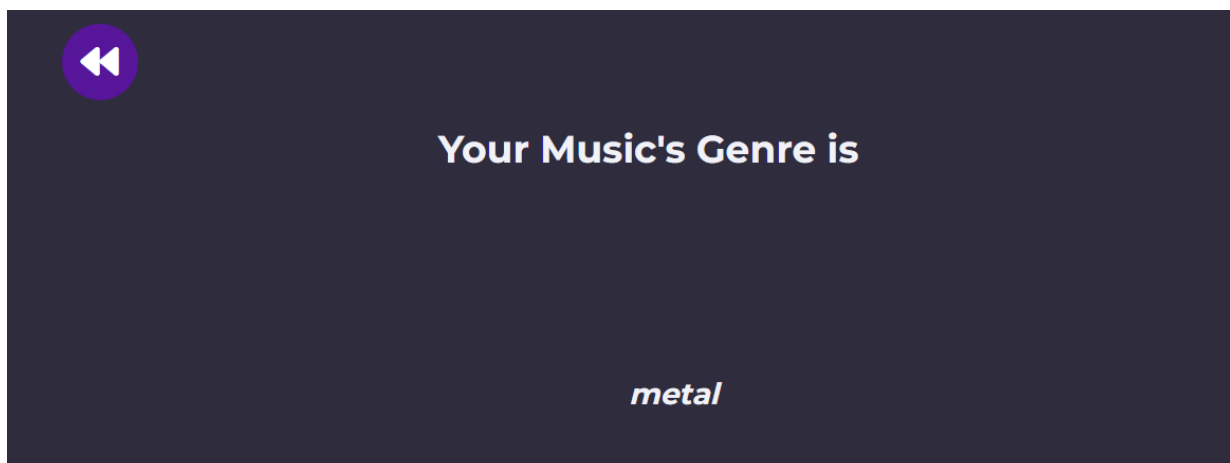


Рисунок 4.29 – Результат класифікації музичного аудіозапису пісні Сльози Осені виконаної іншим виконавцем

Як і було очікувано пісня, навіть від іншого виконавця має жанр метал. Спектро грама треку також доводить це, спостерігаємо потужні агресивні ритми на рисунку 4.30.

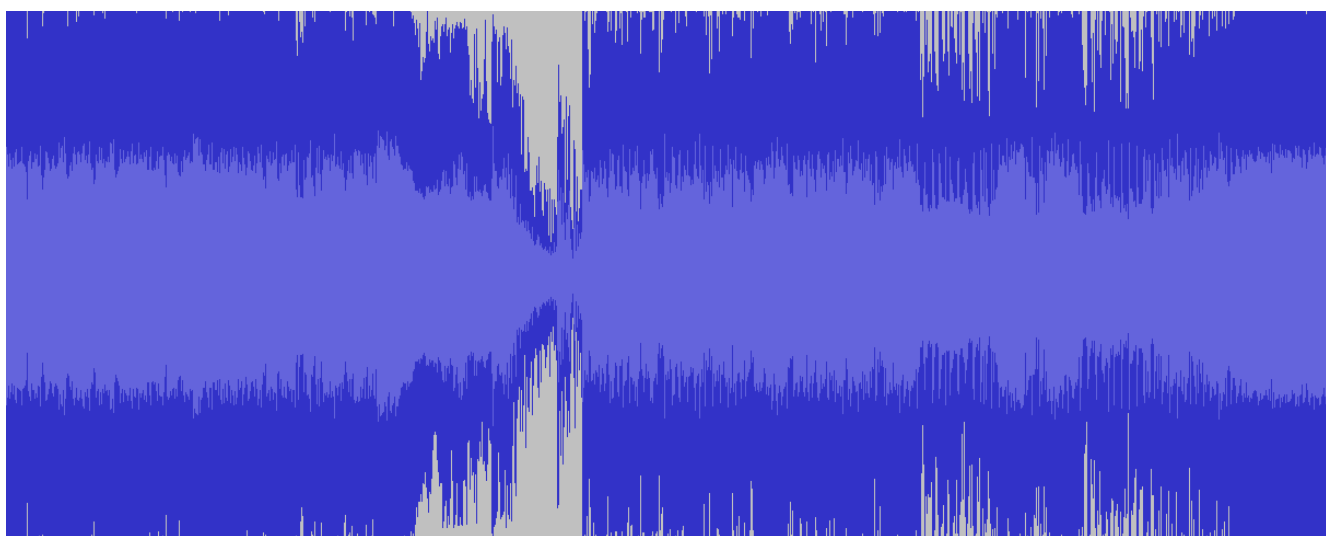


Рисунок 4.30 – Спектрограма треку Сльози Осені

Останнім тестом було вирішено проаналізувати пісню власного виконання та створення – Файно Вале. Результат показано на рисунку 4.31.

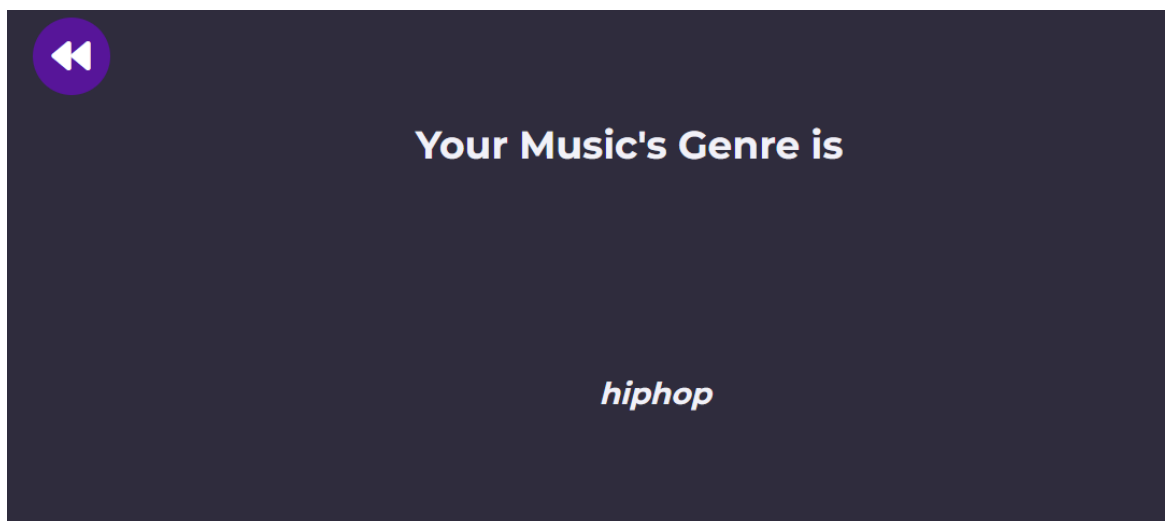


Рисунок 4.31 – Результати класифікації аудіозапису власного виконання та авторства

Таким чином було визначено, що пісня власного виконання має найбільше елементів хіп-хопу, аніж інших жанрів. Також розглянемо спектрограму треку на рисунку 4.32.

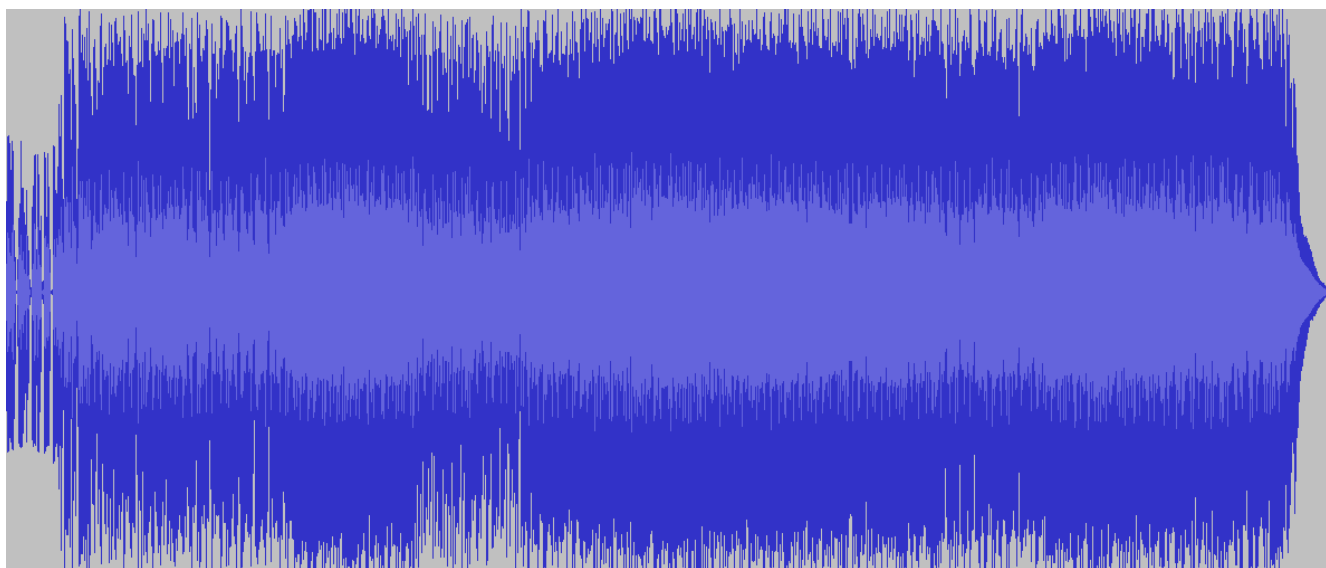


Рисунок 4.32 – Спектрограма композиції власного виконання

Отже, було проведено функціональне тестування розробленої системи та показано отримані результати. Таким чином було протестовано десять музичних аудіофайлів із різними жанрами для можливості переконатись у коректній роботі системи. Також було проведено тестування реакції системи на некоректний ввід даних.

Після проведеного тестування зроблено висновок – розроблена система працює коректно.

4.2 Розробка інструкції користувача

Інструкція користувача передбачає його ознайомлення з вимогами до програмного забезпечення та послідовність дій під час його використання.

Перед початком використання користувач має переконатися, що його пристрій відповідає мінімальним технічним вимогам, які зведено в таблицю 4.1 або рекомендованим технічним вимогам, які зведено в таблицю 4.2. Також, варто зауважити, що, оскільки розроблена система є веб-застосунком, то технічні вимоги – це вимоги, що забезпечують роботу браузера на пристрої.

Таблиця 4.1 – Мінімальні технічні вимоги для успішної експлуатації розробленої системи

Вимоги до конфігурації	
Процесор	Intel Core i3 або еквівалентний
Оперативна пам'ять	512 МБ
Вільне місце на диску	200 МБ
Веб-браузер	Google Chrome або аналог
Операційна система	Будь-яка, що підтримує приєднання до мережі Інтернет
Монітор	Наявний

Таблиця 4.2 – Рекомендовані технічні вимоги для успішної експлуатації розробленої системи

Рекомендована конфігурація	
Процесор	Intel Core i5 або еквівалентний
Оперативна пам'ять	8 ГБ RAM
Вільне місце на диску	1 ГБ
Веб-браузер	Google Chrome або аналог
Операційна система	Будь-яка, що підтримує приєднання до мережі Інтернет
Монітор	Наявний

1. Введення посилання: Користувачу необхідно відкрити веб-браузер та ввести адресу веб-сторінки. Це можна зробити шляхом введення URL-адреси в адресному рядку браузера або переходом за посиланням.
2. Завантаження аудіофайлу: Після входу на веб-сторінку користувачу слід обрати музичний аудіофайл у форматі WAV з власного пристрою. Це можна зробити шляхом натискання на кнопку, що відповідає за завантаження файлу. Після цього користувачеві буде запропоновано вибрати необхідний аудіофайл у файловій системі свого пристрою та підтвердити вибір.
3. Перевірка файлу: Після завантаження аудіофайлу користувач повинен натиснути на кнопку «Check», щоб розпочати процес класифікації аудіофайлу за жанрами.
4. Отримання результату: Після завершення обробки веб-сторінка автоматично показує користувачеві результати класифікації.

Отже, було розроблено інструкцію користувача та виділено технічні вимоги. Інструкція користувача містить послідовність дій, необхідних для коректної роботи з розробленим веб-застосунком, а також визначає технічні вимоги, які має відповідати пристрій користувача для успішного використання програмного забезпечення.

4.3 Висновки

Отож, у четвертому розділі було проведено тестування розробленої системи – протестовано можливість класифікації усіх наявних жанрів та продемонстровано результати. Також було протестовано реакцію системи на введення некоректних даних. Із заданими тестами система справилась навідмінно.

Після цього було розроблено інструкцію користувача, де було виділено потрібні технічні вимоги та розписано послідовність дій для успішної роботи користувача із веб-застосунком.

ВИСНОВКИ

Отже, у результаті виконання бакалаврської дипломної роботи були розроблені модулі для веб-застосунку для аналізу та класифікації музичних аудіозаписів за жанрами з використанням машинного навчання. Розробка виконувалася у середовищі програмування PyCharm, що дозволило зручно та ефективно вести роботу над проєктом.

Під час виконання проєкту був проведений аналіз сучасного стану проблеми, виявлено переваги та недоліки аналогів програмного продукту, таких як Shazam, YouTube та Spotify, і порівняно їх з власним розробленим продуктом. На основі цього порівняння було сформульовано завдання проєкту.

У процесі аналізу технологій було обгрунтовано вибір мови програмування Python та його бібліотек, таких як Librosa, scikit-learn для реалізації програмної частини, а також фреймворку Django, що забезпечив гарну роботу серверної частини. Для реалізації фронтенду було використано такі мови як HTML, CSS, зокрема бібліотеку Bootstrap, JS та її бібліотеку jQuery.

У результаті роботи було виконано всі поставлені задачі, зокрема:

- проведено аналіз існуючих сервісів та додатків, що працюють з аудіозаписами;
- зібрано та підготовлено дані для тренування моделі класифікації;
- розроблено алгоритм для роботи програми із аудіозаписами;
- навчено модель машинного навчання класифікувати музичні аудіозаписи за жанром;
- реалізовано зручний інтерфейс користувача;
- проведено тестування та валідацію розробленої моделі.

Також були розроблені схеми роботи основних алгоритмів програми та створено UML діаграму діяльності, що ілюструє модель роботи системи.

Розроблена система має необмежені ресурси для розвитку – починаючи від розширення асортименту жанрів та закінчуючи інтегруванням у інші сервіси, що надають можливості для прослуховування та роботи із аудіозаписами.

Тестування показало повну готовність розробленої програми до роботи та відповідність поставленим задачам, що підтверджує успішну реалізацію проєкту.

ЛІТЕРАТУРА

1. Музика та її жанри [Електронний ресурс] – Режим доступу до ресурсу:
<https://madlife.webnode.com.ua/l/muzika-ta-jiji-zhanri/>
2. Shazam [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.shazam.com>
3. Youtube [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.youtube.com>
4. Luts M. Python. Довідник програміста – Київ 2023 – 29 с.
5. Librosa [Електронний ресурс] – Режим доступу до ресурсу:
<https://librosa.org/doc/latest/index.html>
6. Scikit-learn [Електронний ресурс] – Режим доступу до ресурсу:
<https://scikit-learn.org/stable/>
7. Harris C.R., Millman K.J., van der Walt S.J., et al. "Array programming with NumPy" // Nature, 2020 – 585 (7825), pp. 357-362.
8. Hunter J.D., Dale D. "Matplotlib: A 2D Graphics Environment" // Computing in Science & Engineering, 2021 – 13 (2), pp. 90-95.
9. McKinney W. "Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython" // O'Reilly Media, 2022 – 550 с.
10. Tzanetakis G., Cook P. "Musical genre classification of audio signals" // IEEE Transactions on Speech and Audio Processing, 2021 – 10 (5), pp. 293-302.
11. Duckett J. "HTML and CSS: Design and Build Websites" // Wiley, 2020 – 512 с.
12. Spurlock J. "Bootstrap: Responsive Web Development" // O'Reilly Media, 2022 – 320 с.
13. McMillan M. Data Structures and Algorithms with JavaScript: Bringing Classic Computing App – O'Reilly, Beijing 2014 – 20 с.
14. Yank S. "Learning jQuery: A Hands-On Guide to Building Rich Interactive Web Frontends" // Addison-Wesley, 2021 – 456 с.

- 15.Kaplan M. "Django for Beginners: Build websites with Python and Django" // Apress, 2020 – 320 с.
- 16.Géron A. "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems" // O'Reilly Media, 2022 – 856 с.
- 17.Stevens L., Antiga L., Viehmann T. "Deep Learning with PyTorch" // Manning Publications, 2020 – 520 с.
- 18.Войтенко І. "Audio Programming with Python" // Київ: Наукова думка, 2023 – 312 с.
- 19.Grinberg M. "Flask Web Development: Developing Web Applications with Python" // O'Reilly Media, 2023 – 400 с.
- 20.Pyne S. "Practical Software Testing" // Packt Publishing, 2021 – 424 с.
- 21.Pedroni N. "Cross-Platform Development in C++: Building Mac OS X, Linux, and Windows Applications" // Addison-Wesley, 2020 – 656 с.
- 22.Іванченко А. В., Чехмestрук Р. Ю. Розробка програми для аналізу та класифікації музичних жанрів на основі аудіозаписів з використанням алгоритмів машинного навчання. // Стан, досягнення та перспективи інформаційних систем і технологій: матеріали XXIV Всеукраїнської науково-технічної конференції молодих вчених, аспірантів та студентів, Одеса, 18-19 квітня 2024 р. Одеса: Видавництво ОНТУ, 2024. 498 с.

ДОДАТКИ

Додаток А – Технічне завдання

57

Додаток А – Технічне завдання
Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії

ЗАТВЕРДЖУЮ
Завідувач кафедри ПЗ
д.т.н., професор
О. Н. Романюк
«12» березня 2024 р.

Технічне завдання
на бакалаврську дипломну роботу «Розробка програми для аналізу та
класифікації музичних жанрів на основі аудіозаписів з використанням
алгоритмів машинного навчання» за спеціальністю 121 – Інженерія
програмного забезпечення

Керівник бакалаврської дипломної роботи:

 к.т.н., доц. каф. ПЗ Чехмestрук Р.Ю.
«12» березня 2024 р.

Виконав:

 студент гр. 2ПІ-206 А. В. Іванченко
«12» березня 2024 р.

Вінниця – 2024 року

1. Найменування та галузь застосування

Бакалаврська дипломна робота: «Розробка програми для аналізу та класифікації музичних жанрів на основі аудіозаписів з використанням алгоритмів машинного навчання.».

Галузь застосування – комерційні музичні платформи.

2. Підстава для розробки.

Підставою для виконання бакалаврської дипломної роботи (БДР) є індивідуальне завдання на БДР та наказ №80 від «11» березня 2024 р. ректора ВНТУ про закріплення тем БДР.

3. Мета та призначення розробки.

Метою роботи є впровадження ефективного методу класифікації музичних композицій на основі аналізу їхніх характеристик та особливостей. Це дозволить підвищити точність і об'єктивність визначення музичних жанрів, що є важливим для музичних платформ, виконавців та користувачів.

4. Вихідні дані для проведення НДР

Перелік основних літературних джерел, на основі яких буде виконуватись БДР.

1. Іванченко А. В., Чехмestрук Р. Ю. Розробка програми для аналізу та класифікації музичних жанрів на основі аудіозаписів з використанням алгоритмів машинного навчання. // Стан, досягнення та перспективи інформаційних систем і технологій: матеріали XXIV Всеукраїнської науково-технічної конференції молодих вчених, аспірантів та студентів, Одеса, 18-19 квітня 2024 р. Одеса: Видавництво ОНТУ, 2024. 498 с.

5. Технічні вимоги

Вихідні дані до роботи: середовище розробки PyCharm, мова розробки Python, фреймворк Django, алгоритми машинного навчання.

6. Конструктивні вимоги.

Графічна та текстова документація повинна відповідати діючим стандартам України.

7. Перелік технічної документації, що пред'являється по закінченню робіт:

- пояснювальна записка до БДР;
- технічне завдання;
- лістинги програми.

8. Вимоги до рівня уніфікації та стандартизації

При розробці програмних засобів слід дотримуватися уніфікації і ДСТУ.

9. Стадії та етапи розробки:

№ п\п	Назва етапів бакалаврської дипломної роботи	Строк виконання етапів роботи
1	Аналіз існуючих сервісів та додатків, що працюють з аудіозаписами	14.03.24 - 22.03.24
2	Збір та підготовка даних для тренування моделі класифікації	24.03.24 - 30.03.24
3	Розробка алгоритмів машинного навчання для класифікації	02.04.24 - 20.04.24
4	Розробка зручного інтерфейсу користувача	21.04.24 - 10.05.24
5	Проведення тестування та валідації розробленої моделі	11.05.24 - 24.05.24
6	Оформлення матеріалів до захисту БДР	25.05.24 - 03.06.24

10. Порядок контролю та прийняття.

Виконання етапів бакалаврської дипломної роботи контролюється керівником згідно з графіком виконання роботи. Прийняття бакалаврської дипломної роботи здійснюється ДЕК, затвердженою зав. кафедрою згідно з графіком.

Додаток Б – Протокол перевірки на плагіат

Додаток Б – Протокол перевірки на плагіат

60

ПРОТОКОЛ
ПЕРЕВІРКИ БАКАЛАВРСЬКОЇ ДИПЛОМНОЇ РОБОТИ
НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ

Назва роботи: Розробка програми для аналізу та класифікації музичних жанрів на основі аудіозаписів з використанням алгоритмів машинного навчання

Тип роботи: БДР

Відділ: кафедра програмного забезпечення, ФІТКІ

Науковий керівник: Чехмestрук Р.Ю.

Оригінальність	93,4
Схожість	6,6

Аналіз звіту подібності

• Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.

☐ Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цілісності роботи і відсутності самостійності її автора. Роботу направити на доопрацювання.

☐ Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату, або в ній містяться навмисні спотворення тексту, що вказують на спроби приховання недобросовісних запозичень.

Особа, відповідальна за перевірку

Черноволик Г.О.

Ознайомлені з повним звітом подібності, який був згенерований системою

Unicheck

Автор роботи

Іванченко А. В.

Керівник роботи

Чехмestрук Р.Ю.

Додаток В – Лістинг програми

Manage.py

```
def main():
os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'MusicClassification.settings')
try:
    from django.core.management import execute_from_command_line
except ImportError as exc:
    raise ImportError(
        "Couldn't import Django. Are you sure it's installed and "
        "available on your PYTHONPATH environment variable? Did you "
        "forget to activate a virtual environment?"
    ) from exc
execute_from_command_line(sys.argv)

if __name__ == '__main__':
    main()
```

Metadata.py

```
def getmetadata(filename):
import librosa
import numpy as np
y, sr = librosa.load(filename)
onset_env = librosa.onset.onset_strength(y, sr)
tempo = librosa.beat.tempo(onset_envelope=onset_env, sr=sr)
y_harmonic, y_percussive = librosa.effects.hpss(y)
tempo, beat_frames = librosa.beat.beat_track(y=y_percussive, sr=sr)
chroma_stft = librosa.feature.chroma_stft(y=y, sr=sr)
rmse = librosa.feature.rms(y=y)
spec_centroid = librosa.feature.spectral_centroid(y, sr=sr)[0]
spec_bw = librosa.feature.spectral_bandwidth(y=y, sr=sr)
spec_rolloff = librosa.feature.spectral_rolloff(y+0.01, sr=sr)[0]
zero_crossing = librosa.feature.zero_crossing_rate(y)
```

```

mfcc = librosa.feature.mfcc(y=y, sr=sr)
metadata_dict = {'tempo':tempo,'chroma_stft':np.mean(chroma_stft),'rmse':np.mean(rmse),
                 'spectral_centroid':np.mean(spec_centroid),'spectral_bandwidth':np.mean(spec_bw),
                 'rolloff':np.mean(spec_rolloff), 'zero_crossing_rates':np.mean(zero_crossing)}

for i in range(1,21):
    metadata_dict.update({'mfcc'+str(i):np.mean(mfcc[i-1])})

return list(metadata_dict.values())

```

predict.py

```

def predict_gen(meta1):
import pickle
import os
from django.conf import settings
path = os.path.join(settings.MODELS, 'models.p')
with open(path, 'rb') as pickled:
    data = pickle.load(pickled)
svmp = data['svmp']
norma = data['norma']
lgn = data['lgn']
x = norma.transform([meta1])
pred = svmp.predict(x)
return(lgn[pred[0]])

```

views.py

```

from django.shortcuts import render, redirect
from django.http import HttpResponseRedirect
from django.urls import reverse
from django.views.generic import ListView
from .apps import PredictorConfig
from .forms import DocumentForm
from .models import Document

```

```

from .Metadata import getmetadata
import warnings
from .predict import predict_gen
from django.contrib import messages
warnings.simplefilter('ignore')

class IndexView(ListView):
    template_name= 'music/index.html'
    def get_queryset(self):
        return True

def model_form_upload(request):

    documents = Document.objects.all()
    if request.method == 'POST':
        if len(request.FILES) == 0:
            messages.error(request,'Upload a file')
            return redirect("predictor:index")

        form = DocumentForm(request.POST, request.FILES)
        if form.is_valid():
            uploadfile = request.FILES['document']
            print(uploadfile.name)
            print(uploadfile.size)
            if not uploadfile.name.endswith('.wav'):
                messages.error(request,'Only .wav file type is allowed')
                return redirect("predictor:index")
            meta = getmetadata(uploadfile)

            genre = predict_gen(meta)
            print(genre)

            context = {'genre':genre}
            return render(request,'music/result.html',context)

```

```
else:
```

```
    form = DocumentForm()
```

```
    return render(request, 'music/result.html', {'documents': documents, 'form': form})
```

```
preprocess.ipynb
```

```
    import pandas as pd
```

```
import numpy as np
```

```
df = pd.read_csv('data.csv')
```

```
df = df.drop(['beats'], axis=1)
```

```
df.head()
```

```
    df['class_name'].unique()
```

```
df['class_name'] = df['class_name'].astype('category')
```

```
df['class_label'] = df['class_name'].cat.codes
```

```
lookup_genre_name = dict(zip(df.class_label.unique(), df.class_name.unique()))
```

```
lookup_genre_name
```

```
df['class_name'].unique()
```

```
cols = list(df.columns)
```

```
cols.remove('label')
```

```
cols.remove('class_label')
```

```
cols.remove('class_name')
```

```
#df[cols]
```

```
    %matplotlib notebook
```

```
from sklearn.model_selection import train_test_split
```

```
import matplotlib.pyplot as plt
```

```
X = df.iloc[:,1:28]
```

```
y = df['class_label']
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=3)
```

```
    from sklearn.preprocessing import MinMaxScaler
```

```
scaler = MinMaxScaler()
```

```
X_train_scaled = scaler.fit_transform(X_train)
```

```
X_test_scaled = scaler.transform(X_test)
```

```
    from sklearn.ensemble import RandomForestClassifier
```

```
%matplotlib notebook
```

```
clf = RandomForestClassifier(random_state=0, n_jobs=-1).fit(X_train_scaled, y_train)
```

```

importances = clf.feature_importances_
indices = np.argsort(importances)[::-1]
names = [X.columns.values[i] for i in indices]
plt.figure()
plt.title("Feature Importance")
plt.bar(range(X.shape[1]), importances[indices])
plt.xticks(range(X.shape[1]), names, rotation=90)
plt.show()

from sklearn.tree import DecisionTreeClassifier
clf = DecisionTreeClassifier(random_state=0).fit(X_train_scaled, y_train)
importances = clf.feature_importances_
indices = np.argsort(importances)[::-1]
names = [X.columns.values[i] for i in indices]
plt.figure()
plt.title("Feature Importance")
plt.bar(range(X.shape[1]), importances[indices])
plt.xticks(range(X.shape[1]), names, rotation=90)
plt.show()

from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors = 13)
knn.fit(X_train_scaled, y_train)
knn.score(X_test_scaled, y_test)

from Metadata import getmetadata
a = getmetadata("3.wav")
d1 = np.array(a)
data1 = scaler.transform([d1])
genre_prediction = knn.predict(data1)
print(lookup_genre_name[genre_prediction[0]])

from sklearn.svm import SVC
clf = SVC(kernel = 'linear', C=10).fit(X_train_scaled, y_train)
clf.score(X_test_scaled, y_test)

d1 = np.array(a)
data1 = scaler.transform([d1])
genre_prediction = clf.predict(data1)
print(lookup_genre_name[genre_prediction[0]])

```

```

import pickle

pick1 = {
    'norma':scaler,
    'svmp':knn,
    'lgn':lookup_genre_name
}

pickle.dump( pick1, open( 'models' + ".p", "wb" ) )

```

mystyle.css

```

body{
    overflow-x: hidden;
}

.up-form{
    position: relative;
    left: 50%;
    margin-top: -100px;
    margin-left: -250px;
    width: 500px;
    height: 200px;
    border: 4px dashed #fff;
}

.up-p{
    width: 100%;
    height: 100%;
    text-align: center;
    line-height: 170px;
    color: #ffffff;
    font-family: Arial;
}

.up-input{
    position: absolute;
    margin: 0;
    padding: 0;
    width: 100%;

```

```

    height: 100%;
    outline: none;
    opacity: 0;
}
.up-button{
    margin: 0;
    color: #fff;
    background: #16a085;
    border: none;
    width: 508px;
    height: 35px;
    margin-top: -20px;
    margin-left: -4px;
    border-radius: 4px;
    border-bottom: 4px solid #117A60;
    transition: all .2s ease;
    outline: none;
}
.up-button:hover{
    background: #149174;
    color: #0C5645;
}
.up-button:active{
    border:0;
}

.card {
    background-color: #262431;
}

.shadow{
    border: 2px solid #e3e3e3;
}

/* Center the loader */

```

```

#loader {
    position: absolute;
    left: 50%;
    top: 50%;
    z-index: 1;
    width: 150px;
    height: 150px;
    margin: -75px 0 0 -75px;
    border: 16px solid #f3f3f3;
    border-radius: 50%;
    border-top: 16px solid #3498db;
    width: 120px;
    height: 120px;
    -webkit-animation: spin 2s linear infinite;
    animation: spin 2s linear infinite;
}

@-webkit-keyframes spin {
    0% { -webkit-transform: rotate(0deg); }
    100% { -webkit-transform: rotate(360deg); }
}

@keyframes spin {
    0% { transform: rotate(0deg); }
    100% { transform: rotate(360deg); }
}

/* Add animation to "page content" */
.animate-bottom {
    position: relative;
    -webkit-animation-name: animatebottom;
    -webkit-animation-duration: 1s;
    animation-name: animatebottom;
    animation-duration: 1s
}

```

```
@-webkit-keyframes animatebottom {
  from { bottom:-100px; opacity:0 }
  to { bottom:0px; opacity:1 }
}
```

```
@keyframes animatebottom {
  from{ bottom:-100px; opacity:0 }
  to{ bottom:0; opacity:1 }
}
```

```
#myDiv {
  display: none;
  text-align: center;
}
```

scripts.js

```
(function($) {
  "use strict";

  /* Preloader */
  $(window).on('load', function() {
    var preloaderFadeOutTime = 500;
    function hidePreloader() {
      var preloader = $('.spinner-wrapper');
      setTimeout(function() {
        preloader.fadeOut(preloaderFadeOutTime);
      }, 500);
    }
    hidePreloader();
  });
});
```

```

/* Navbar Scripts */

// jQuery to collapse the navbar on scroll
$(window).on('scroll load', function() {
    if ($("#navbar").offset().top > 20) {
        $(".fixed-top").addClass("top-nav-collapse");
    } else {
        $(".fixed-top").removeClass("top-nav-collapse");
    }
});

$(function() {
    $(document).on('click', 'a.page-scroll', function(event) {
        var $anchor = $(this);
        $('html, body').stop().animate({
            scrollTop: $($anchor.attr('href')).offset().top
        }, 600, 'easeInOutExpo');
        event.preventDefault();
    });
});

$("#navbar-nav li a").on("click", function(event) {
    if (!$(this).parent().hasClass('dropdown'))
        $(".navbar-collapse").collapse('hide');
});

$("#js-rotating").Morphext({
    animation: "dissolve",
    separator: ",",
    speed: 2000,
    complete: function () {

```

```

    }
  });

```

```

var cardSlider = new Swiper('.card-slider', {
  autoplay: {
    delay: 4000,
    disableOnInteraction: false
  },
  loop: true,
  navigation: {
    nextEl: '.swiper-button-next',
    prevEl: '.swiper-button-prev'
  },
  slidesPerView: 3,
  spaceBetween: 20,
  breakpoints: {
    992: {
      slidesPerView: 2
    },
    768: {
      slidesPerView: 1
    }
  }
});

```

```

var imageSlider = new Swiper('.image-slider', {
  autoplay: {
    delay: 2000,
    disableOnInteraction: false
  },
  loop: false,
  navigation: {
    nextEl: '.swiper-button-next',

```

```

    prevEl: '.swiper-button-prev',
  },
  spaceBetween: 30,
  slidesPerView: 5,
  breakpoints: {
    380: {
      slidesPerView: 1,
      spaceBetween: 10
    },
    516: {
      slidesPerView: 2,
      spaceBetween: 10
    },
    768: {
      slidesPerView: 3,
      spaceBetween: 20
    },
    992: {
      slidesPerView: 4,
      spaceBetween: 30
    },
    1200: {
      slidesPerView: 5,
      spaceBetween: 30
    },
  }
});

```

```

$('.popup-link').magnificPopup({
  removalDelay: 300,
  type: 'image',
  callbacks: {
    beforeOpen: function() {
      this.st.image.markup = this.st.image.markup.replace('mfp-figure', 'mfp-figure ' +

```

```
this.st.el.attr('data-effect'));
```

```
    },
    beforeClose: function() {
        $('.mfp-figure').addClass('fadeOut');
    }
},
gallery:{
    enabled:true
}
});
```

```
$('.popup-youtube, .popup-vimeo').magnificPopup({
    disableOn: 700,
    type: 'iframe',
    mainClass: 'mfp-fade',
    removalDelay: 160,
    preloader: false,
    fixedContentPos: false,
    iframe: {
        patterns: {
            youtube: {
                index: 'youtube.com/',
                id: function(url) {
                    var m = url.match(/[\\?\\&]v=([\\?\\&]+)/);
                    if ( !m || !m[1] ) return null;
                    return m[1];
                },
                src: 'https://www.youtube.com/embed/%id%?autoplay=1'
            },
            vimeo: {
                index: 'vimeo.com/',
                id: function(url) {
                    var m = url.match(/(https?:\\V)?(www.)?(player.)?vimeo.com\\([a-z]*V)*([0-9]{6,11})[\\]??.*/);
```

```

        if ( !m || !m[5] ) return null;
        return m[5];
    },
    src: 'https://player.vimeo.com/video/%id%?autoplay=1'
}
}
}
});

```

```

$('.popup-with-move-anim').magnificPopup({
    type: 'inline',
    fixedContentPos: false,
    fixedBgPos: true,
    overflowY: 'auto',
    closeBtnInside: true,
    preloader: false,
    midClick: true,
    removalDelay: 300,
    mainClass: 'my-mfp-slide-bottom'
});

```

```

var a = 0;
$(window).scroll(function() {
    if ($('#counter').length) {
        var oTop = $('#counter').offset().top - window.innerHeight;
        if (a == 0 && $(window).scrollTop() > oTop) {
            $('.counter-value').each(function() {
                var $this = $(this),
                    countTo = $this.attr('data-count');
                $({
                    countNum: $this.text()
                }).animate({

```

```

        countNum: countTo
    },
    {
        duration: 2000,
        easing: 'swing',
        step: function() {
            $this.text(Math.floor(this.countNum));
        },
        complete: function() {
            $this.text(this.countNum);
            //alert('finished');
        }
    });
});
a = 1;
}
}
});

```

```

$("input, textarea").keyup(function(){
    if ($(this).val() != "") {
        $(this).addClass('notEmpty');
    } else {
        $(this).removeClass('notEmpty');
    }
});

```

```

$("#contactForm").validator().on("submit", function(event) {
    if (event.isDefaultPrevented()) {
        cformError();
        csubmitMSG(false, "Please fill all fields!");
    } else {
        event.preventDefault();
    }
});

```

```

        csubmitForm();
    }
});

```

```

function csubmitForm() {
    var name = $("#cname").val();
    var email = $("#cemail").val();
    var message = $("#cmmessage").val();
    var terms = $("#cterms").val();
    $.ajax({
        type: "POST",
        url: "php/contactform-process.php",
        data: "name=" + name + "&email=" + email + "&message=" + message + "&terms=" + terms,
        success: function(text) {
            if (text == "success") {
                cformSuccess();
            } else {
                cformError();
                csubmitMSG(false, text);
            }
        }
    });
}

```

```

function cformSuccess() {
    $("#contactForm")[0].reset();
    csubmitMSG(true, "Message Submitted!");
    $("input").removeClass('notEmpty');
    $("textarea").removeClass('notEmpty');
}

```

```

function cformError() {
    $("#contactForm").removeClass().addClass('shake animated').one('webkitAnimationEnd
mozAnimationEnd MSAnimationEnd oanimationend animationend', function() {
        $(this).removeClass();
    });
}

```

```
});  
}
```

```
function csubmitMSG(valid, msg) {  
    if (valid) {  
        var msgClasses = "h3 text-center tada animated";  
    } else {  
        var msgClasses = "h3 text-center";  
    }  
    $("#cmsgSubmit").removeClass().addClass(msgClasses).text(msg);  
}
```

```
$("#privacyForm").validator().on("submit", function(event) {  
    if (event.isDefaultPrevented()) {  
        // handle the invalid form...  
        pformError();  
        psubmitMSG(false, "Please fill all fields!");  
    } else {  
        // everything looks good!  
        event.preventDefault();  
        psubmitForm();  
    }  
});
```

```
function psubmitForm() {  
    var name = $("#pname").val();  
    var email = $("#pemail").val();  
    var select = $("#pselect").val();  
    var terms = $("#pterms").val();  
  
    $.ajax({  
        type: "POST",  
        url: "php/privacyform-process.php",  
        data: "name=" + name + "&email=" + email + "&select=" + select + "&terms=" + terms,
```

```

    success: function(text) {
        if (text == "success") {
            pformSuccess();
        } else {
            pformError();
            psubmitMSG(false, text);
        }
    }
});
}

```

```

function pformSuccess() {
    $("#privacyForm")[0].reset();
    psubmitMSG(true, "Request Submitted!");
    $("input").removeClass('notEmpty'); // resets the field label after submission
}

```

```

function pformError() {
    $("#privacyForm").removeClass().addClass('shake animated').one('webkitAnimationEnd
mozAnimationEnd MSAnimationEnd oanimationend animationend', function() {
        $(this).removeClass();
    });
}

```

```

function psubmitMSG(valid, msg) {
    if (valid) {
        var msgClasses = "h3 text-center tada animated";
    } else {
        var msgClasses = "h3 text-center";
    }
    $("#pmsgSubmit").removeClass().addClass(msgClasses).text(msg);
}

```

```

$('body').prepend('<a href="body" class="back-to-top page-scroll">Back to Top</a>');

```

```
var amountScrolled = 700;
$(window).scroll(function() {
    if ($(window).scrollTop() > amountScrolled) {
        $('a.back-to-top').fadeIn('500');
    } else {
        $('a.back-to-top').fadeOut('500');
    }
});

$(".button, a, button").mouseup(function() {
    $(this).blur();
});

})(jQuery);
```

Додаток Г – Ілюстративна частина

ВІННИЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА КОМП'ЮТЕРНОЇ
ІНЖЕНЕРІЇ

КАФЕДРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Бакалаврська дипломна робота на тему:

РОЗРОБКА ПРОГРАМИ ДЛЯ АНАЛІЗУ ТА КЛАСИФІКАЦІЇ МУЗИЧНИХ ЖАНРІВ НА ОСНОВІ АУДІОЗАПИСІВ З ВИКОРИСТАННЯМ АЛГОРИТМІВ МАШИННОГО НАВЧАННЯ



Автор:
студент 2ПІ-206 Іванченко А.В.
Науковий керівник:
к.т.н., доц. каф. ПЗ Чехместрук Р.Ю.

Вінниця 2024

Рисунок Г.1 – Вступний слайд

Актуальність роботи

У сучасному світі більшість виконавців самі визначають жанр своїх творів, проте це визначення може бути суб'єктивним та недостатньо точним. У багатьох випадках, особливо у випадку музики, яка поєднує в собі різні стилі та впливи, визначення жанру стає складним завданням, що потребує об'єктивного підходу. Тому розробка програмних засобів системи визначення музичних жанрів аудіозаписів є надзвичайно актуальною задачею.



Рисунок Г.2 – Актуальність роботи

Мета, предмет та об'єкт дослідження

Мета роботи – впровадження ефективного методу класифікації музичних композицій на основі аналізу їхніх характеристик та особливостей. Це дозволить підвищити точність і об'єктивність визначення музичних жанрів, що є важливим для музичних платформ, виконавців та користувачів.

Об'єкт дослідження – процес обробки аудіоданих та аналізу звукових сигналів, а також методи та алгоритми машинного навчання, які застосовуються для класифікації музичних композицій.

Предмет дослідження – розробка програмних засобів, які дозволять автоматизувати процес визначення жанру музичних творів на основі обробки аудіозаписів та аналізу їхніх акустичних та музичних властивостей.

Рисунок Г.3 – Мета, предмет та об'єкт дослідження

Аналіз аналогів

Критерії оцінювання	Shazam	Youtube	Spotify	Розроблюваний програмний застосунок
Відкритий доступ до алгоритму дії і користування ним	+	-	-	+
Підтримка різних ОС	+	+	+	+
Можливість аналізу нових аудіозаписів	-	-	-	+
Можливість розширення функціоналу	+	-	-	+
Можливість інтеграції в інші додатки	+	+	-	+
Загальна оцінка	80%	40%	20%	100%

Рисунок Г.4 – Аналіз аналогів

Постановка задачі проєкту

Було проведено аналіз аналогів та визначено їх сильні та слабкі сторони, оцінено методи можливого вирішення задачі, та сформульовано наступні завдання для реалізації проєкту:

- збір та підготовка даних для тренування моделі класифікації;
- розробка алгоритмів машинного навчання для класифікації;
- розробка алгоритмів роботи програми із аудіозаписами;
- навчання моделі машинного навчання класифікувати музичні аудіозаписи за жанрами;
- розробка зручного інтерфейсу користувача;
- проведення тестування та валідації розробленої моделі.



Рисунок Г.5 – Постановка задачі

Новизна

Розвитку набув новий підхід до класифікації музичних жанрів, що базується на використанні алгоритмів машинного навчання. Запропоноване рішення дозволяє автоматизувати процес визначення жанру музичних композицій, забезпечуючи високу точність та об'єктивність результатів.



Рисунок Г.6 – Наукова новизна

Практична цінність

Полягає в створенні інструменту, який може бути використаний для автоматизованого визначення жанру музичних композицій у різноманітних ситуаціях, таких як аналіз великих обсягів аудіоданих, рекомендації музики на музичних платформах або створення персоналізованих плейлистів для користувачів.



Рисунок Г.7 – Практична цінність

Використані технології



django



Рисунок Г.8 – Використані технології

Дані для навчання моделі машинного навчання

GTZAN dataset – колекція із 1000 музичних аудіозаписів, що поділяються на 10 жанрів – по 100 аудіозаписів на жанр. Колекція оптимізована для навчання моделі машинного навчання, всі аудіозаписи зведені та обрізані на 30-ти секундні відрізки із найбільш явними ознаками жанру.



Рисунок Г.9 – Дані для навчання моделі

Основні аспекти реалізації проєкту

- Дані розділені на навчальний і тестовий набори для уникнення перенавчання і точного оцінювання моделі.
- Використаний алгоритм випадкового лісу з бібліотеки sklearn.
- Використовується клас sklearn для масштабування ознак до діапазону між 0 і 1, що оптимізує роботу моделі.
- Важливість ознак аналізується для розуміння їх внеску у класифікацію.

Рисунок Г.10 – Основні аспекти реалізації проєкту

Принцип роботи програмного модуля

Конвертація аудіосигналу у числові значення Екстрагування аудіоознак

Аналіз темпу і ритму, розділення аудіоданих на гармонічні та перкусивні складові, обчислення середнього значення енергії та кількості переходів через нуль у сигналі.

Обробка спектру, аналіз спектрограми, спектрального центроїду, ширини смуги, ролоффу та мел-частотних кепстральних коефіцієнтів.

Створення словника з характеристиками.

Передбачення класу і визначення жанру музики

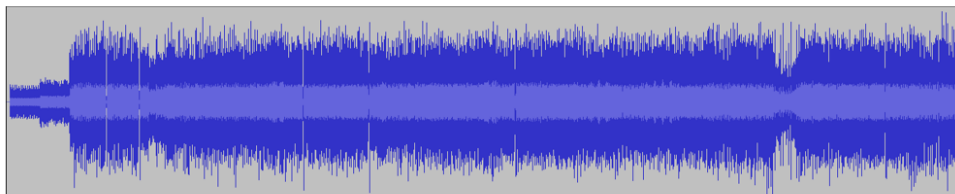
Рисунок Г.11 – Принцип роботи програмного модуля

Блок-схема
роботи
програмного
модуля



Рисунок Г.12 – Блок-схема роботи програмного модуля

Аналіз композиції Cowboys from Hell



Спектрограма композиції

Результати аналізу

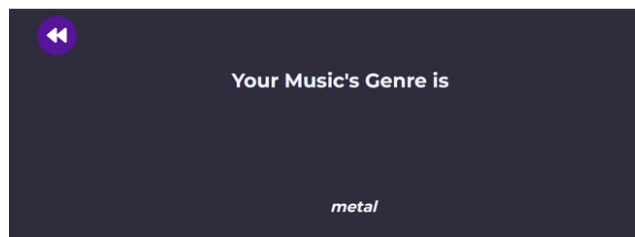
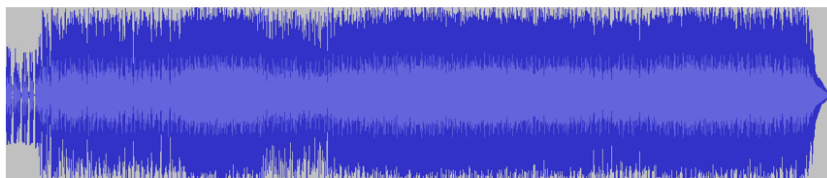


Рисунок Г.13 – Тестування. Аналіз композиції Pantera – Cowboys from hell

Аналіз на прикладі композиції власного виконання



Спектрограма композиції

Результати аналізу

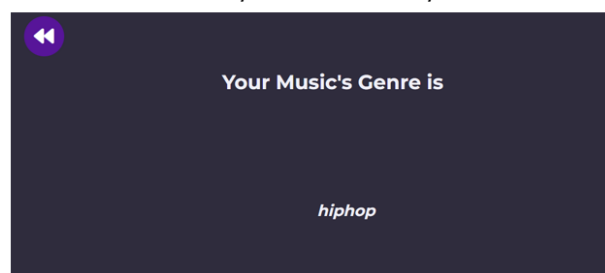


Рисунок Г.14 – Тестування. Аналіз композиції власного виконання

Апробація результатів

Здійснена на конференції XXIV Всеукраїнської науково-технічної конференції молодих вчених, аспірантів та студентів (2024) і опубліковані в тезах доповіді цієї конференції



Рисунок Г.15 – Апробація результатів

ВИСНОВКИ

В результаті роботи були розроблені модулі для веб-застосунку для аналізу та класифікації музичних аудіозаписів за жанрами з використанням машинного навчання.

Було проведено

- Аналіз існуючих сервісів.
- Підготовку даних для моделі.
- Розроблено алгоритм для роботи з аудіозаписами.
- Навчено модель класифікувати жанри.
- Реалізовано зручний інтерфейс для користувача.
- Тестування та валідацію моделі.

Система готова до роботи та має потенціал для подальшого розвитку та інтеграції з іншими сервісами.

Рисунок Г.16 – Висновки. Фінальний слайд