МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

Інститут КОМП’ЮТЕРНИХ НАУК ТА ІНФОРМАЦІЙНИХ технологій

кафедра систем штучного інтелекту



**Звіт**

про виконання комплексної проектної роботи

з дисциплін «Машинне навчання» та «Методи аналізу Великих даних»

на тему:

«Застосування методів машинного навчання для прогнозування жанру музики»

Виконав:

*ст. групи КНСШ-12*

*Ткачук О.Р.*

Перевірили:

*доцент каф. СШІ, к.е.н.,*

*Бойко Н.І.,*

*доцент каф. СШІ, к.т.н.,*

*Хавалко В.М.*

Львів – 2023

**ЗМІСТ**

[**ВСТУП** 3](#_Toc126941210)

[**РОЗДІЛ 1. ВИБІР ПРЕДМЕТНОЇ ОБЛАСТІ ТА ОПИС НАБОРУ ДАНИХ** 5](#_Toc126941211)

[**1.1.** **Постановка задачі** 6](#_Toc126941212)

[**1.2.** **Опис набору даних** 7](#_Toc126941213)

[**1.3.** **Процедури попереднього опрацювання даних** 14](#_Toc126941214)

[**РОЗДІЛ 2. ДОСЛІДНИЦЬКИЙ РОЗДІЛ** 20](#_Toc126941215)

[**2.1.** **Опис та обгрунтування обраних моделей машинного навчання** 20](#_Toc126941216)

[**2.2.** **Процедури підбору оптимальних параметрів роботи моделей** 24](#_Toc126941217)

[**2.3.** **Показники оцінювання ефективності роботи моделей** 29](#_Toc126941218)

[**РОЗДІЛ 3. МОДЕЛЮВАННЯ** 31](#_Toc126941219)

[**3.1.** **Результати аналізу та попереднього опрацювання даних** 31](#_Toc126941220)

[**3.2.** **Оцінювання ефективності роботи обраних моделей та їх порівняння** 44](#_Toc126941221)

[**3.3.** **Інтерпретація результатів роботи найкращої моделі** 51](#_Toc126941222)

[**ВИСНОВКИ** 52](#_Toc126941223)

[**СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ** 53](#_Toc126941224)

[**ДОДАТОК А Програмна реалізація** 55](#_Toc126941225)

[**ДОДАТОК Б Комбінований графік відношень ознак розділений по жанрах** 56](#_Toc126941226)

# **ВСТУП**

Музика є одним із найскладніших видів мистецтва, який відрізняється різноманітністю емоційних сигналів і тембральних якостей. Для музикантів розуміння «міксу», який робить певний жанр або стиль унікальним, може зайняти роки, щоб освоїти його. Навіть для найдосвідченішого знавця музики може бути складно визначити, який жанр чи стиль звучить на певній композиції. Тепер, завдяки прогресу в машинному навчанні (ML), ми на шляху до автоматизації процесу класифікації музичних жанрів. У цій статті досліджується, як методи класифікації ML можуть допомогти нам зрозуміти широкий спектр музичних стилів і впорядкувати складний ландшафт музики.

Розвиток цифрової музики призвів до створення багатьох нових музичних стилів і жанрів. Сьогодні кожен знайде щось для себе: від електронних музичних композицій, у яких багато синтетичних ритмів і експериментальних звуків, до позачасового спокою класичної музики. Хоча багато різних атрибутів допомагають визначити, що робить певний трек відповідним до певної категорії, експертна думка залишається одним із засобів для класифікації жанрів. Проте завдяки прогресу в алгоритмах ML і наборах даних машинне навчання тепер може аналізувати звукові сигнали, щоб виявляти характеристики, які повертають їх до значущих класів, таких як хіп-хоп/реп або народна музика.

Крім того, нещодавні розробки дозволили моделям ML точно класифікувати пісні за різними темпами та тактовими розмірами зі швидкістю, що наближається до рівня точності людини. Оскільки ця технологія швидко стає все більш досконалою, музикантам більше не доведеться покладатися виключно на власні знання при визначенні музичних типів, до яких належать різні треки. Знання того, до якого жанру пісня належить, дає артистам цінне уявлення про її потенційну аудиторію. Зрештою, класифікація за допомогою ML може революціонізувати те, як люди взаємодіють із музикою та відкривають для неї нові форми так, як ніколи раніше не вважалося можливим.

Метою даної роботи являється аналіз різних методів машинного навчання для класифікації музики за жанром.

Предметом дослідження являється класифікація музики за допомогою методів автоматизованого машинного навчання.

Об’єктом дослідження є класифікаційні моделі машинного навчання.

# **РОЗДІЛ 1. ВИБІР ПРЕДМЕТНОЇ ОБЛАСТІ ТА ОПИС НАБОРУ ДАНИХ**

Аналіз необроблених музичних даних відіграє важливу роль при завантаженні треків у цифрові бібліотеки та потокові сервіси. Аналізуючи такі дані, як тональність, темп, удари в хвилину та інструменти, аудіоінженери можуть легко каталогізувати та організовувати необроблені музичні записи в цифрові колекції з легким пошуком. Цей процес може значно зменшити кількість ручної роботи, необхідної для завантаження та зберігання записів у цифрових аудіотеках. Крім того, метадані, пов’язані з музичними записами, використовуються для ідентифікації та індексування пісень на основі їхніх музичних характеристик – для створення індивідуальних налаштувань списку відтворення, які відповідають різноманітним уподобанням користувача. Корисність аналізу необроблених музичних даних не обмежується завантаженням треків; це також покращує точність алгоритмів штучного інтелекту для класифікації жанру або виконання операцій розтягування/стиснення часу.

Загалом аналіз необроблених музичних даних є важливим для багатьох аспектів створення та зберігання музики. Завдяки складним програмним забезпеченням, які фіксують складні параметри зі звуків, у поєднанні з алгоритмами машинного навчання для розпізнавання шаблонів у кількох зразках інженери можуть точніше ідентифікувати інструменти та створювати легкозасвоювані бібліотечні системи.

Завантаження музичних даних передбачає доступ до онлайнової бази даних потокового сервісу, наприклад Spotify[1]. Дані зазвичай складаються з треків, виконавців і альбомів разом із пов’язаними з ними характеристиками, такими як жанр, популярність виконавця та дата випуску[2]. Після завантаження в пам’ять його можна проаналізувати, щоб отримати уявлення про структуру музики в цілому, вплив культурних тенденцій на звички слухання та різноманітність жанрів, які існують у популярній музиці. Застосовуючи різні алгоритми та методи статистичного аналізу до даних, деякі ключові закономірності та характеристики стають очевидними.

## **Постановка задачі**

Класифікація музики – це процес віднесення музики до різних жанрів, стилів і категорій. Це можна розглядати як важливий спосіб організації музики для кращої навігації чи пошуку. З розвитком технологій це завдання стало більш складним через різноманітність музичних жанрів і піджанрів. Алгоритми машинного навчання дозволяють класифікувати будь-який тип аудіосигналів із необроблених аудіофайлів в один або кілька попередньо визначених класів.

Щоб досягти успішних результатів у сфері автоматизованої класифікації музики, різні алгоритми машинного навчання потрібно оцінювати за кількома наборами даних. Різні методи виділення ознак використовуються для захоплення важливих елементів, які характеризують певний музичний твір, таких як ритм, гармонія, тембр тощо. Потім ці функції використовуються як вхідні дані для алгоритмів навчання, таких як глибокі нейронні мережі, які можуть отримувати знання щодо кожного музичного жанру або стиль і розпізнавати його характеристики в невидимих зразках даних.

Ще одна проблема під час роботи з класифікацією музики полягає в тому, як знайти релевантну структуру у великих наборах даних, щоб побудувати потужні моделі, здатні ідентифікувати закономірності між багатьма окремими аудіозаписами. Крім того, враховуючи велику різноманітність музичних творів, які часто дуже схожі за своєю природою та можуть мати характеристики, що збігаються, робить видалення шуму та вибір функцій важливими кроками для попередньої обробки зразків перед подачею їх у будь-яку систему класифікатора.

Нарешті, серед інших аспектів, таких як показники оцінки продуктивності та методи оптимізації витрат, необхідно розробити надійні протоколи оцінки, щоб побудувати надійні системи, які можуть точно призначати нові елементи до правильної відповідної категорії за відповідного набору даних. Через ці згадані труднощі ця область все ще активно досліджується, навіть незважаючи на значний прогрес, досягнутий за останні роки завдяки вдосконаленню апаратних можливостей, а також новим розробкам спеціалізованих пакетів програмного забезпечення для програм машинного навчання обробки аудіосигналів.

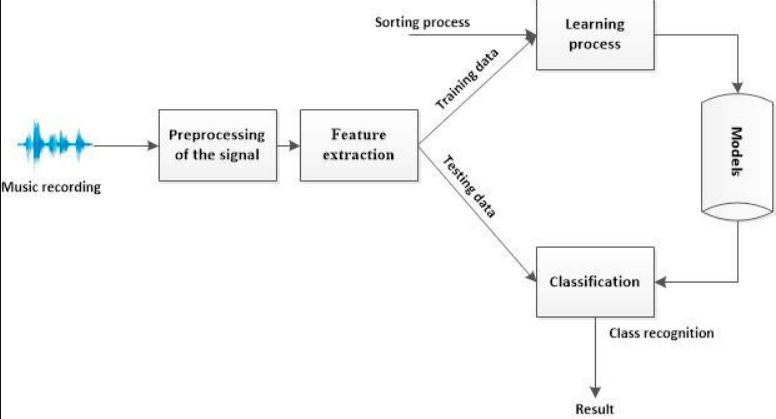


Рис. 1.1. Приклад системи класифікації музики

## **Опис набору даних**

Метою опису набору даних є надання огляду контексту, у якому було створено даний набір даних. Він може окреслити передумови, мету та ключові характеристики набору даних, щоб дослідникам було легше зрозуміти, який тип інформації та аналізу можна з нього отримати. Опис також може містити важливі відомості про те, коли та де були отримані дані, а також будь-які обмеження чи проблеми, які слід враховувати під час роботи з ними. Це особливо важливо для великих складних наборів даних, які можуть вимагати спеціальних знань, щоб зрозуміти їх. Зрештою, наявність ретельного опису допомагає переконатися, що дослідники можуть точно інтерпретувати свої результати та ефективно використовувати дані.

Для початку виведемо інформацію про датасет:

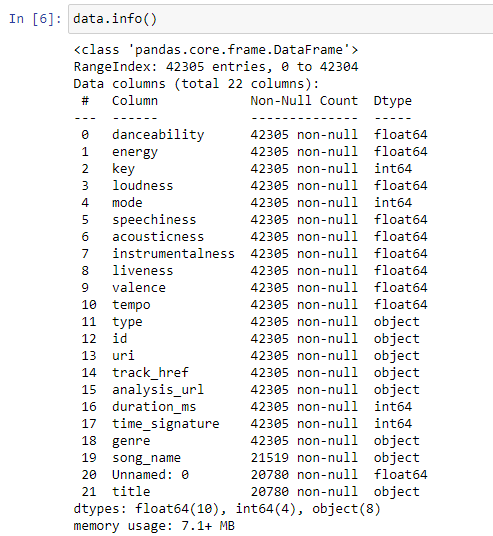
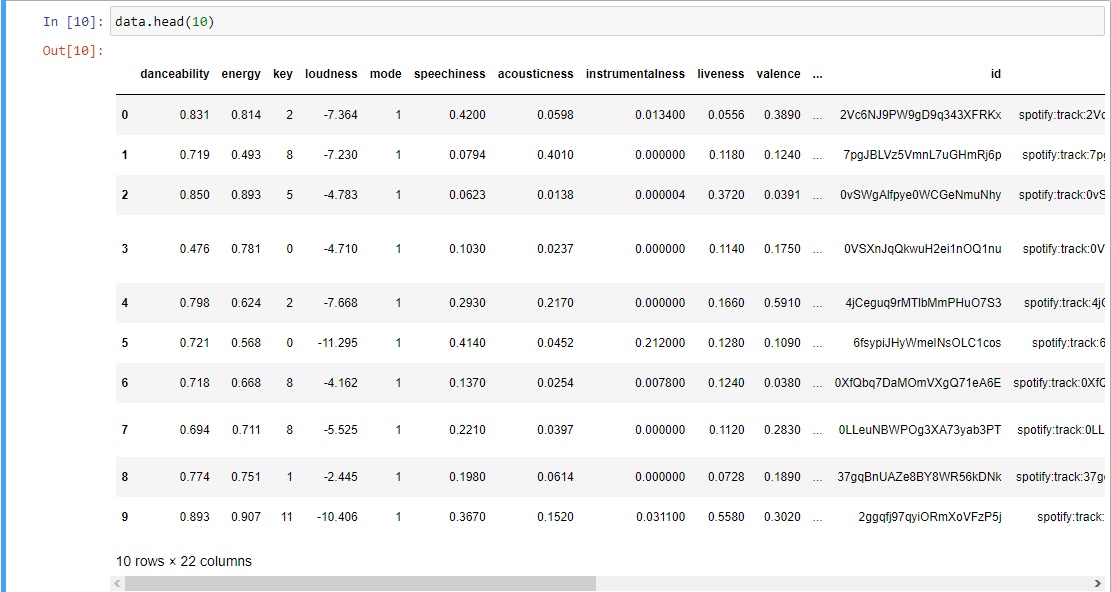


Рис. 1.2. Загальгна інформація про дані

* 'danceability' – Танцювальність описує, наскільки трек придатний для танцю на основі поєднання музичних елементів, включаючи темп, стабільність ритму, силу ритму та загальну регулярність. Значення 0,0 є найменш танцювальним, а 1,0 – найбільш танцювальним;
* 'energy' – Енергія є мірою від 0,0 до 1,0 і являє собою міру сприйняття інтенсивності та активності. Як правило, енергійні треки здаються швидкими, гучними та шумними. Наприклад, дез-метал має високу енергію, тоді як прелюдія Баха має низькі бали за шкалою. Характеристики сприйняття, що впливають на цей атрибут, включають динамічний діапазон, сприйману гучність, тембр, швидкість початку та загальна ентропія;
* 'key' – Тональність треку. Цілі числа відображаються на висоту звуку за допомогою стандартної нотації Pitch Class. наприклад 0 = C, 1 = C♯/D♭, 2 = D тощо. Якщо ключ не виявлено, значення -1;
* 'loudness' – Загальна гучність треку в децибелах (дБ). Значення гучності усереднені по всій доріжці та корисні для порівняння відносної гучності доріжок. Гучність - це якість звуку, яка є основним психологічним корелятом фізичної сили (амплітуди). Значення зазвичай коливаються від -60 до 0 дБ;
* 'mode' – Режим вказує на модальність (мажорну чи мінорну) доріжки, тип звукоряду, з якого походить її мелодійний зміст. Мажор позначається 1, а мінор – 0;
* 'speechiness' – розмовність визначає наявність вимовлених слів у доріжці. Чим більш ексклюзивно мовний запис (наприклад, ток-шоу, аудіокнига, поезія), тим ближче до 1,0 значення атрибута. Значення вище 0,66 описують треки, які, ймовірно, повністю складаються з вимовлених слів. Значення від 0,33 до 0,66 описують доріжки, які можуть містити як музику, так і мову, у розділах або на шарах, включаючи такі випадки, як реп-музика. Значення нижче 0,33, швидше за все, представляють музику та інші композиції, не схожі на мову;
* 'acousticness' – Міра достовірності від 0,0 до 1,0 того, чи трек є акустичним. 1.0 свідчить про високу впевненість, що композиція є акустичною;
* 'instrumentalness' – Прогнозує, чи доріжка не містить вокал. Звуки «Ох» і «Аа» розглядаються як інструментальні в цьому контексті. Реп або розмовні композиції явно «вокальні». Чим ближче значення інструментальності до 1,0, тим більша ймовірність того, що трек не містить вокального вмісту. Значення понад 0,5 призначені для представлення інструментальних треків, але впевненість вища, коли значення наближається до 1,0;
* 'liveness' – Визначає присутність аудиторії в записі. Більш високі значення живості означають підвищену ймовірність того, що композиція була виконана наживо. Значення вище 0,8 забезпечує високу ймовірність того, що трек активний;
* 'valence' – Міра від 0,0 до 1,0, що описує музичну позитивність, яку передає трек. Композиції з високою валентністю звучать більш позитивно (наприклад, весело, весело, ейфорично), тоді як треки з низькою валентністю звучать більш негативно (наприклад, сумно, пригнічено, сердито).
* 'tempo' – Загальний очікуваний темп треку в ударах на хвилину (BPM). У музичній термінології темп - це швидкість або темп певної частини, яка безпосередньо випливає з середньої тривалості такту;
* 'type' – Тип об'єкта;
* 'id' – Ідентифікатор Spotify для треку;
* 'uri' – URI Spotify для треку;
* 'track\_href' – Посилання на кінцеву точку Web API з повною інформацією про трек;
* 'analysis\_url' – URL-адреса для доступу до повного аудіоаналізу цієї доріжки. Для доступу до цих даних потрібен маркер доступу;
* 'duration\_ms' – Тривалість треку в мілісекундах;
* 'time\_signature' – Приблизний тактовий розмір. Тактовий розмір (метр) — це умовне позначення, яке визначає кількість ударів у кожному такті (або такті). Тактовий розмір коливається від 3 до 7, вказуючи на тактовий розмір від «3/4» до «7/4»;
* 'genre' – Музичний жанр, до якого належить аудіотрек.
* 'song\_name' – Назва треку;
* 'Unnamed: 0' – Ідентифікатор у випадку відсутності назви;
* 'title' – Альтернативна назва, що містить жанр;



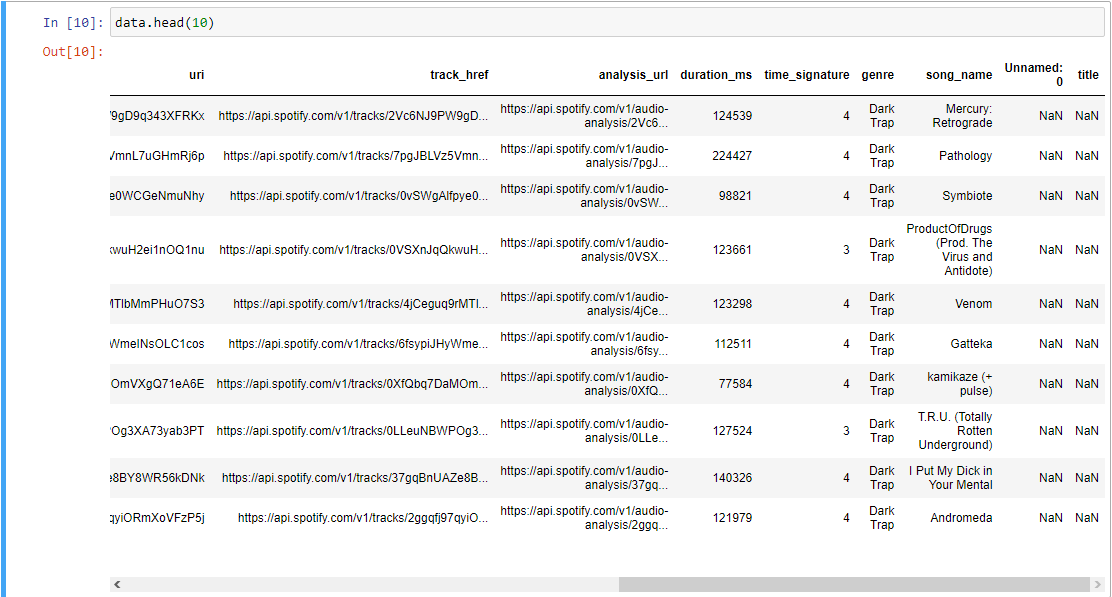
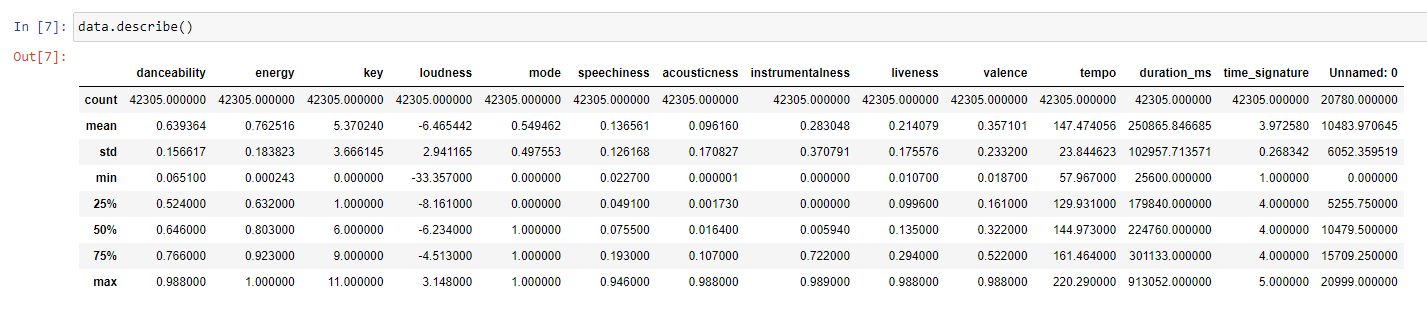


Рис. 1.3. Загальний вигляд датасету

Описова статистика (рис.1.4.) включає статистику, яка підсумовує центральну тенденцію, дисперсію та форму розподілу набору даних, за винятком значень NaN. Аналізує числові ряди та ряди об’єктів, а також набори стовпців DataFrame змішаних типів даних. Результат буде різним залежно від того, що надається.

Рис. 1.4. Описова статистика датасету

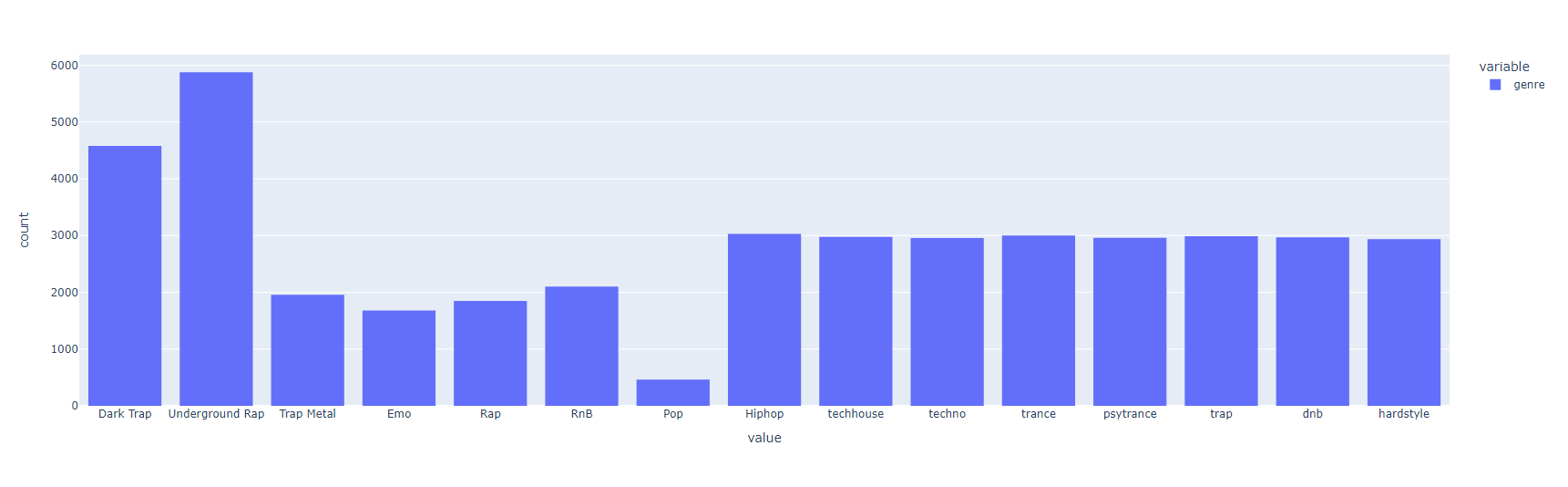
Аналіз розподілу даних є важливою частиною аналізу даних, оскільки він допомагає краще зрозуміти набір даних і його характеристики[3]. Аналізуючи розподіл значень між різними змінними в наборі даних, ми можемо краще ідентифікувати викиди або неочікувані результати, а також виявити значущі тенденції та закономірності. Завдяки розумінню статистичних даних і візуальних представлень, створених на основі цього аналізу, ми можемо зробити значущі висновки щодо наших даних, які можуть обґрунтувати рішення та допомогти визначити області для покращення.

Рис. 1.5. Розподіл даних в датасеті відносно жанру

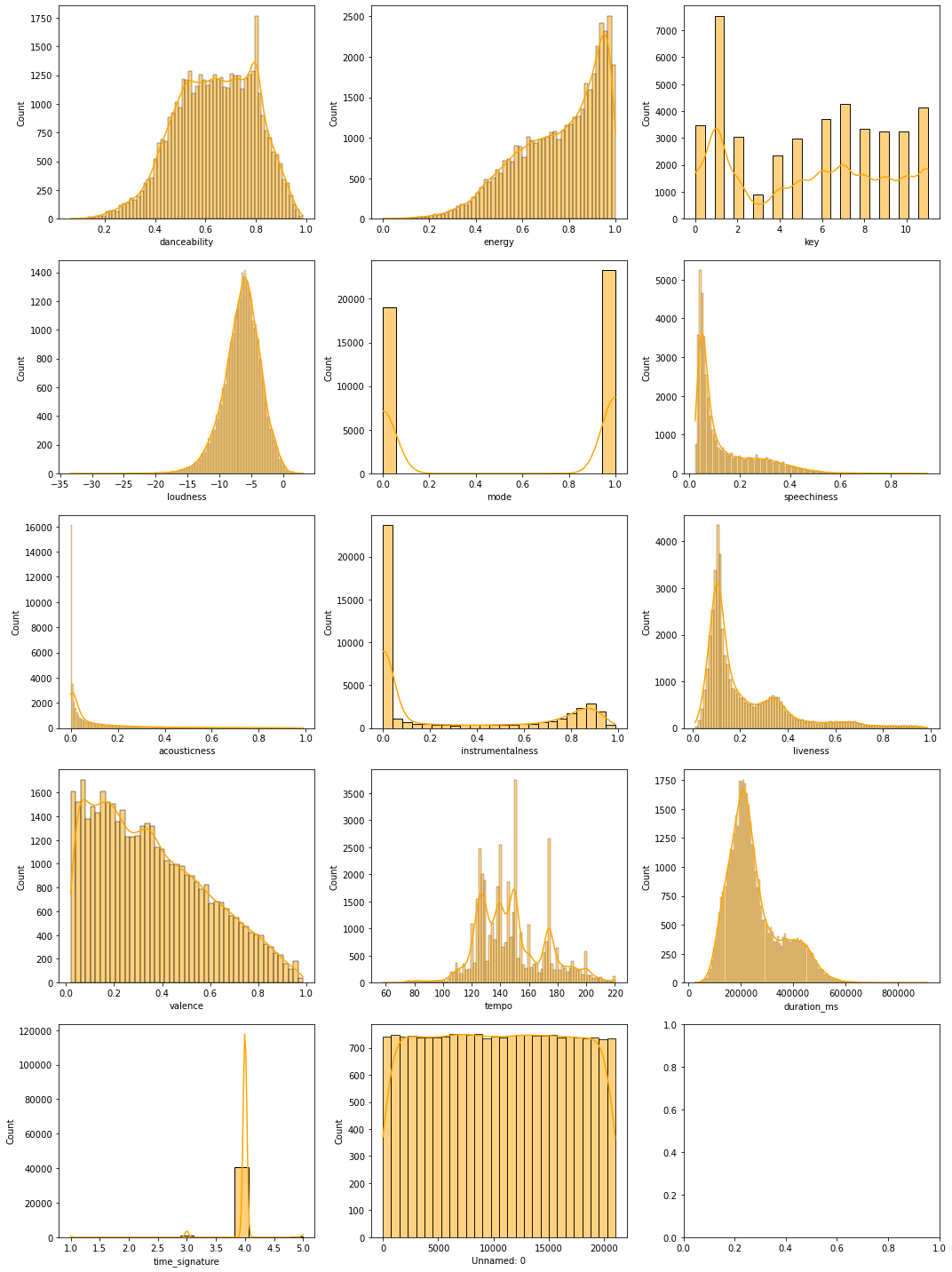


Рис. 1.6. Розподіл даних в датасеті по кожній ознаці

Досліджуючи рис.1.6. можна побачити, що танцювальність має приблизно нормальний розподіл. Енергетична функція, якщо негативно зміщена. Функція гучності зазвичай розподіляється з кількома викидами. Кількість екземплярів із модальністю 1 більша, ніж з модальністю 0. Характеристика «мовленнєвості» позитивно зміщена, тобто середнє значення функції «мовленнєвості» більше, ніж медіана. Характеристики «акустичності», «інструментальності», «жвавості» та валентності – усі вони позитивно скошені. Функції "temp" і "duration\_ms", мають нормальний розподіл.

## **Процедури попереднього опрацювання даних**

Препроцесинг[4] у машинному навчанні – це дуже важливий крок, який допомагає підвищити якість даних, отримуючи тільки значущу інформацію з них. Попередня обробка даних у машинному навчанні відноситься до техніки підготовки (очищення та упорядкування) вихідних даних, щоб зробити їх придатними для побудови та навчання різноманітних моделей машинного навчання. Отже, попередня обробка даних у машинному навчанні - це метод інтелектуального аналізу даних, який перетворює неопрацьовані дані у зрозумілий і читабельний формат.

Що стосується створення моделі машинного навчання, то попередня обробка даних є першим кроком, що позначає початок процесу. На жаль, вхідні дані часто є неповними, непослідовними, неточними (містять помилки або викиди) і не мають конкретних значень/тенденцій атрибутів. Саме в таких випадках необхідна попередня обробка даних, оскільки вона допомагає очистити, відформатувати та організувати вхідні дані, тим самим зробивши їх готовими до використання у тренуванні моделей машинного навчання.

До етапів попередньої обробки даних у машинному навчанні слід віднести:

*1. Видалення викидів (outliers)*

Являється важливим етапом в попередній обробці даних. Викид - це певний об’єкт генеральної сукупності, який дуже сильно відрізняється від генеральної сукупності. Тобто він має характеристики, що відрізняються від більшості інших об’єктів у наборі даних. Наявність викидів у наборі даних може негативно вплинути на якість натренованої моделі машинного навчання. Саме тому, основним етапом препроцесингу є боротьба з викидами.

Крім того, важливо розрізняти шум та викиди. Викиди можуть бути законними об'єктами даних або значеннями, тому, на відміну від шуму, викиди іноді можуть викликати інтерес для подальшого дослідження.

Існує декілька способів боротьби з викидами:

1. Standard Deviation Method або метод середньоквадратичного відхилення. Якщо відомо, що набір даних відповідає нормальному розподілу або наближений до нього, то можна використати метод середньоквадратичного відхилення для видалення викидів. Нормальний розподіл містить 2 важливих параметри - mean та standard deviation. В нормальному розподілі ці 2 параметри можуть бути використані для того, щоб визначити у вибірці нетипові дані. У межах одного стандартного відхилення від середнього значення - 68% даних. У межах двох стандартних відхилень - 95%, а у межах трьох - 99.7%. Дані, що лежать поза 3 стандартними відхиленнями є також частиною генеральної сукупності, але це неправильні, нетипові або ж дуже рідкісні випадки даних. Отож, зазвичай, усі дані, що лежать поза 3 стандартними відхиленням вважаються викидами та підлягають видаленню з генеральної сукупності. Ці значення можуть мінятися в залежності від розміру датасету: якщо розмір даних є дуже великим, то можуть викидатися дані, що лежать поза 4 стандартними відхиленнями, або ж навпаки - якщо розмір даних є малий, то викидаються дані, які лежать поза 2 стандартними відхиленнями.
2. Interquartile Range Method або метод міжквартильного розмаху. Добрим способом для узагальнення гаусівського розподілу даних є міжквартильний діапазон. Міжквартильний діапазон обчислюється як різниця між 75 і 25 процентилями даних. Методи виявлення викидів на основі статистики передбачають, що вони виникають в областях з низькою ймовірністю стохастичної моделі, а нормальні дані відповідно в областях з високою ймовірністю. Тому метод міжквартильного розмаху може бути використаний для ідентифікації викидів, визначаючи межі значень вибірки, так званим коефіцієнтом k IQR, що лежить нижче 25 -го процентиля або вище 75 -го процентиля. Загальновживаним значенням коефіцієнта k є 1.5.

*2. Виявлення та обробка відсутніх значень*

Під час попередньої обробки даних важливо визначити та правильно обробити відсутні значення, якщо цього не зробити, то можна отримати неточні та хибні результати та відповідно висновки щодо даних. Існує два способи обробки відсутніх даних:

1. Deleting a particular row ( Видалення певного рядка). У цьому методі видаляється певний рядок, який має нульове значення для об’єкта або певного стовпця. Переваги такого методу - отримання надійної моделі. Недоліки - втрата великої кількості інформації, і рекомендується використовувати його лише тоді, коли в наборі даних є достатня кількість даних з непустими значеннями.
2. Calculating the mean or median ( Розрахунок середнього значення або медіани) - Цей метод корисний для об’єктів, що мають числові дані, такі як вік, заробітна плата, рік тощо. Тут можна обчислити середнє значення, медіану чи режим певної функції чи стовпця чи рядка, що містить відсутнє значення, і замінити результат для відсутнього значення. Переваги - не відбувається втрата даних, добре працює з малими наборами даних та легкий в імплементації. Недоліки - працює тільки з числовими даними, не враховується коваріантність між атрибутами.

*3. Кодування категоріальних даних*

Категоріальні дані відносяться до інформації, яка має певні категорії в наборі даних. Моделі машинного навчання насамперед базуються на математичних рівняннях. Таким чином, можна інтуїтивно зрозуміти, що збереження категоріальних даних у рівнянні спричинить певні проблеми, оскільки потрібні лише числа у рівняннях.

Розглянемо два найпоширеніші методи: One-Hot-Encoding та Label-Encoder.

Обидва ці кодери є частиною бібліотеки SciKit-learn (однієї з найбільш широко використовуваних бібліотек Python). Її використовуються для перетворення текстових або категорійних даних у числові дані, які модель очікує і з якими краще працює.

* Label-Encoder. Цей підхід передбачає перетворення кожного значення у стовпці в число. Даний алгоритм кодування має великий мінус. Проблема використання числа полягає в тому, що вони вводять відношення/порівняння між категоріями. Таким чином, модель ML може припустити, що між цими змінними існує деяка кореляція, тим самим створюючи несправний вихід. Але слід зауважити, якщо у стовпці потрібно дотримуватися порядку/пріоритетності, наприклад створення рівня безпеки (high, medium, low ), то цей метод кодування є цілком підходящим.
* One-Hot-Encoding. У цій стратегії кожне значення категорії перетворюється в новий стовпець, якому призначається значення 1 або 0 (позначення істинного/хибного). Хоча цей підхід усуває проблеми з ієрархією/порядком, але має і зворотну сторону - додавання додаткових стовпців до набору даних. Це може спричинити значне збільшення кількості стовпців, якщо у стовпці категорії є багато унікальних значень.

*4. Розбиття набору даних*

Кожен набір даних, перед тим як потрапити на тренування моделі машинного навчання розділяється на навчальний та тестувальний набори даних.

Тренувальний набір даних використовується для підбору відповідних параметрів моделі машинного навчання, які якнайкраще б задовільнили тренувальний набір даних. При цьому модель підбирає відповідні параметри за певним оптимізаційним алгоритмом, найчастіше це градієнтний спуск.

Тестувальний набір даних використовується для забезпечення неупередженої оцінки отриманої після тренування моделі машинного навчання. Оцінка буде неупередженою, оскільки тестувальні дані модель ще не зустрічала.

Хорошим тоном вважається розділення набору даних також і на так званий затверджувальний датасет. Затверджувальний датасет використовується при тренування моделі машинного навчання. Він також забезпечує неупереджену оцінку на етапі тренування моделі та може бути використаний для регуляризації шляхом ранньої зупинки, щоб уникнути перенавчання та погіршення результату вихідної моделі.

Набір даних можна розділяти у різних співвідношеннях. Це залежить від розміру даних та їх розкиду. Дуже часто використовуються такі співвідношення (навчальний-затверджувальний- тестовий): 60 - 20 - 20 / 70 - 15 - 15 / 80 - 10 - 10.

*5. Масштабування атрибутів (Feature scaling)*

Масштабування атрибутів- це приведення всіх об’єктів у наборі даних до однакового масштабу, це необхідно під час навчання моделі машинного навчання, оскільки в деяких випадках певні атрибути стають настільки домінуючими, що інші звичайні атрибути можуть не враховуватися моделлю.

Існують 2 способи привести атрибути до одного масштабу:

1. Стандартизація (Standard scaling) -

Стандартизація приводить усі атрибути об’єктів у межі значень від -1 до 1. Тобто дані перетворюються таким чином, що значення атрибуту буде розподілом із середнім значенням 0 та стандартним відхиленням 1. Такий спосіб ще називають центруванням.

1. Нормалізація

При нормалізації набір даних набирає значення всіх об’єктів між 0 і 1, так що всі стовпці знаходяться в одному діапазоні, і, отже, немає домінуючої функції.

Масштабування атрибутів позначає кінець попередньої обробки даних у машинному навчанні. Це метод приведення незалежних змінних набору даних у певний загальний діапазон значень. Іншими словами, масштабування атрибутів обмежує діапазон змінних, щоб модель могла порівнювати їх з усіма іншими незалежно від масштабу значення.

# **РОЗДІЛ 2. ДОСЛІДНИЦЬКИЙ РОЗДІЛ**

Класифікація музики є важливою задачею в обробці аудіосигналів[5]–[7], яка використовується для ідентифікації різних музичних інструментів або жанрів музики. Це можна зробити як вручну, так і за допомогою моделей машинного навчання. Було розроблено багато алгоритмів машинного навчання, які використовують керовані та неконтрольовані методи для точної класифікації музики відповідно до її жанру, інструменту, виконавця чи інших характеристик[8]. У цьому розділі обговорюватимуться різні моделі машинного навчання, які використовуються для проблем класифікації музики, і те, як вони працюють для точного визначення типу музики. Також тут буде описано деякі стратегії, які можна застосувати для підвищення точності цих моделей та описано їх гіперпараметри[9].

## **Опис та обгрунтування обраних моделей машинного навчання**

Першою перевагою використання логістичної регресії є її інтерпретація. Логістична регресія часто використовується як метод для розуміння впливу кожної функції на змінну відповіді та для виявлення закономірностей у даних, які представляють різні зв’язки між функціями та цікавим результатом. Це може дати розуміння того, як різні елементи взаємодіють, полегшуючи дослідникам прийняття кращих рішень щодо свого проекту.

Ще однією перевагою логістичної регресії є простота її використання. Моделі логістичної регресії можна швидко адаптувати до існуючих типів даних завдяки їх спрощеному математичному формулюванню та простішому процесу підгонки, ніж складніші алгоритми. Крім того, оскільки це лінійна модель, її можна легко інтерпретувати після встановлення та не потребує жодних процесів налаштування чи оптимізації, що робить логістичну регресію ефективним інструментом для швидкого прогнозування з надійною точністю. Ці властивості роблять його ідеальним вибором для тих, хто хоче створювати прогнозні моделі, не витрачаючи надто багато часу на дослідження даних або налаштування алгоритмів.

Однією з переваг використання класифікатора Gaussian Naive Bayes для класифікації музики є його швидкість. Порівняно з іншими алгоритмами машинного навчання, Gaussian Naive Bayes може навчати та класифікувати музичні дані набагато швидше, що робить його ідеальним інструментом для додатків у реальному часі. Крім того, його хороша точність також є бажаною рисою. Це може бути особливо корисним у додатках, де точність на ряду зі швидкістю обробки вихідних даних моделі має першочергове значення, наприклад, передбачення жанру нового музичного твору або автоматичний вибір відповідних пісень із великої бібліотеки.

Ще одна перевага алгоритму Байєса Гауса полягає в тому, що він вимагає відносно мало вхідних даних; оскільки йому потрібні лише базові музичні дані (темп, ритм, інструменти тощо), його можна швидко й успішно навчити, використовуючи мінімальну кількість навчальних даних. Крім того, завдяки своїй простоті та зрозумілій реалізації Gaussian Naive Bayes не потребує значного налаштування чи налаштування параметрів, перш ніж його можна буде розгорнути у виробничих середовищах. Підсумовуючи, ці переваги роблять Gaussian Naive Bayes хорошим варіантом для ефективних і точних завдань класифікації музики

Першою перевагою використання k-найближчих сусідів (KNN) для класифікації музики є його простота. KNN простий для розуміння, швидкий для навчання та високоефективний за належного впровадження. Основний принцип KNN полягає в тому, що точки даних, розташовані близько одна до одної в багатовимірному просторі (наприклад, музичному просторі), ймовірно, належатимуть до однієї групи. Таким чином, йому потрібно лише кілька позначених прикладів на етапі навчання, і він може швидко передбачити клас нових екземплярів, дивлячись на своїх найближчих сусідів.

Також, KNN повністю непараметричний, тобто він не робить жодних припущень щодо базової структури даних або будь-яких попередніх знань про класи. Це усуває будь-яке можливе упередження, що випливає з припущень, накладених на дані. Крім того, KNN також може добре справляється з шумом і викидами порівняно з іншими алгоритмами, оскільки він покладається на локальну топологію, а не на глобальні статистичні властивості набору даних. Оскільки музичні зразки часто містять різноманітні типи шумів або варіації в функціях, ця перевага значно покращить їх застосування для завдань класифікації музики.

Однією з головних переваг використання дерев рішень для класифікації музики є їх точність і ефективність. Дерева рішень забезпечують ефективний засіб аналізу складних необроблених даних, таких як музичний тембр і частота, що призводить до точно класифікованих аудіофрагментів. Крім того, дерева рішень є дуже надійними щодо зашумлених аудіоданих або відсутніх значень даних завдяки їхній здатності легко справлятися з розбіжностями в наборах даних. Крім того, дерева рішень не вимагають багато навчальних даних, оскільки вони автоматично навчаються з характеристик вхідних шаблонів. Це прискорює завдання навчання моделі для передбачення правильної мітки класу для кожного шаблону введення.

Ще одна перевага використання моделей дерева рішень для класифікації музики полягає в тому, що вони можуть класифікувати декілька класів одночасно. Традиційні алгоритми, такі як Наївний Байєс, зазвичай вимагають одного класифікатора на завдання та багато контрольованого навчання для налаштування його параметрів; однак, порівняно з цими методами, моделі дерева рішень більш ефективні за часом. Крім того, оскільки для цього не потрібні спеціальні методи розробки функцій, моделі дерева рішень виявилися ефективнішими та ефективнішими порівняно з більшістю інструментів машинного навчання при використанні на наборах даних, пов’язаних із музикою. Здатність цих алгоритмів визначати закономірності навіть у великих наборах даних робить їх дуже придатними для завдань класифікації музики, таких як завдання ідентифікації жанру чи аудіоособливостей.

Однією з головних переваг використання Random Forest для класифікації музики є його здатність обробляти багатовимірні дані. Один Random Forest може брати до уваги тисячі характеристик, таких як висота, гучність, тривалість тощо, щоб точно класифікувати пісню у відповідній категорії, що робить його одним із найпотужніших методів для завдань класифікації музики.

Ще одна перевага полягає в тому, що випадкові ліси дуже стійкі до перенавчання, оскільки вони покладаються на початкову вибірку, яка бере повторні вибірки з набору даних із заміною. Ця техніка ще більше підвищує точність моделі, уникаючи перенавчання. Використання begging (початкове агрегування) є основоположним у випадкових лісах, оскільки воно зменшує стандартну помилку, пов’язану з будь-яким окремим деревом, і, отже, призводить до більш точних прогнозів. Крім того, міра важливості, пов’язана з кожною функцією, допомагає визначити, які характеристики мають вищу передбачувану силу для конкретного завдання, що дозволяє нам краще зрозуміти продуктивність нашої моделі та внести відповідні коригування.

Перша велика перевага використання методу опорних векторів (SVM) для класифікації музики полягає в тому, що вона надзвичайно ефективна при класифікації складних аудіосигналів. SVM розроблено для створення моделей із максимальним узагальненням для класифікації складних даних, завдяки чому вони добре підходять для відокремлення різних класів від різноманітних вхідних даних. Для класифікації музики це означає, що SVM може розрізняти різні жанри або звуки в межах одного вхідного сигналу. Крім того, використання ядер дозволяє моделі SVM створювати межу шляхом відображення точок даних у просторі більших розмірів, що дає змогу точно класифікувати кілька тонів, навіть якщо вони накладаються в межах набору даних.

Ще однією перевагою використання SVM для музичної класифікації є його надійність і масштабованість. На відміну від інших алгоритмів класифікації, таких як дерева рішень і нейронні мережі, SVM потребують менше навчання та налаштування, що робить їх ідеальними для швидкого створення базових музичних каталогів без потреби в значних налаштуваннях параметрів. Крім того, завдяки їхній здатності легко масштабуватися порівняно з іншими популярними методами, SVM не тільки ідеально підходять для великих наборів даних, але й вимагає низьких накладних витрат пам’яті, що робить його особливо корисними під час роботи з недостатнім обладнанням або ресурсами. Це полегшує розробникам і дослідникам швидке впровадження ефективного рішення для каталогізації музики будь-якого рівня складності.

## **Процедури підбору оптимальних параметрів роботи моделей**

Оптимальні робочі параметри моделі[9] можна визначити шляхом оцінки продуктивності моделі до та після зміни певних параметрів, таких як швидкість навчання, розмір партії, функція активації або оптимізатор. Зазвичай це робиться методом проб і помилок, щоб знайти найкращу комбінацію параметрів, яка дає найточніші результати. Крім того, дослідники можуть використовувати встановлені евристики та метрики для визначення оптимальних значень для кожного параметра. Нарешті, здатність моделі узагальнювати невидимі дані можна оцінити за допомогою незалежних наборів даних перевірки.

**Гіперпараметри логістичної регресії**

1. Штраф: цей гіперпараметр визначає, який тип регулярізації застосовувати (наприклад, L1, L2).
2. C (Інтенсивність, обернена регуляризації): контролює ступінь регуляризації, регулюючи ступінь штрафу для коефіцієнтів (ваг) моделі. Менше значення дасть сильнішу регуляризовану модель, і навпаки.
3. Вирішувач: алгоритм, який використовується для навчання моделі (наприклад, liblinear, lbfgs).
4. Мультиклас: стратегія обробки кількох класів (наприклад, ovr: один проти решти або багаточлен).
5. Максимальна кількість ітерацій: максимальна кількість ітерацій розв’язувачем перед його завершенням.
6. Масштабування: чи слід стандартизувати вхідні змінні перед операцією підгонки (наприклад, True, False).
7. Критерій допуску для зупинки: Порогове значення для зупинки процесу оптимізації (наприклад, 0,0001).

**Гіперпараметри наївного Баєса**

* 1. varsmoothing: цей параметр дозволяє нам контролювати дисперсію кожного атрибута, який можна використовувати для згладжування небажаних шумів у наших даних. Низьке значення цього параметра передбачає більшу регуляризацію, а вище значення означає меншу регуляризацію.
  2. priors: це встановлює пріоритетні ймовірності кожного класу. Ми можемо призначати різну вагу класам, змінюючи цей параметр, який використовується під час прогнозування.
  3. fitprior: цей параметр визначає, чи слід оцінювати попередні ймовірності на основі даних навчання (True) чи ні (False). Якщо False, використовуються однакові попередні.
  4. algorithm: тут визначається алгоритм, який використовується для обчислення апостеріорних ймовірностей для кожного класу. Деякі з доступних алгоритмів – «map», «mle» і «bayes», що означає «максимальний апостериорний», «оцінка максимальної правдоподібності» та підхід Байєса відповідно.

**Гіперпараметри дерева рішень**

* 1. Максимальна глибина: максимальна глибина дерева. Якщо немає, то вузли розгортаються, доки всі листи не стануть чистими або поки всі листи не міститимуть менше вибірок minsamplesssplit.
  2. Мінімальна кількість вибірок: мінімальна кількість вибірок, необхідних для розділення внутрішнього вузла.
  3. Мінімальна кількість зразків листків: мінімальна кількість зразків, необхідна для листкового вузла.
  4. Максимальна кількість функцій: максимальна кількість функцій, які слід враховувати під час пошуку найкращого розділення.
  5. Максимальна кількість вузлів листя: виростіть дерево з максимальними вузлами листя за принципом найкраще. Найкращі вузли визначаються як відносне зменшення домішок.
  6. Критерій: функція для вимірювання якості розбиття; Підтримуваними критеріями є «джині» для домішки Джіні та «ентропія» для вимірювання приросту інформації або ентропійної домішки відповідно.

**Гіперпараметри KNN**

* 1. nneighbors: це ціле число, яке визначає кількість найближчих сусідів, які слід враховувати під час формування прогнозу. Збільшуючи цей параметр, алгоритм враховуватиме більше даних у своїх прогнозах.
  2. weights: це може бути «uniform» або «distance». Якщо встановлено значення «uniform», усі сусіди будуть однаково зважені в прогнозі; якщо встановлено значення «distance», точки, розташовані ближче, отримають вищу вагу в процесі прогнозування.
  3. metric: різні метрики відстані, такі як Манхеттенська, Евклідова, Мінковського тощо, можна використовувати для обчислення відстані між точками даних під час пошуку найближчих сусідів для прогнозів.
  4. algorithm: це вказує на тип алгоритму, що використовується для ідентифікації найближчих сусідів, на основі яких робляться прогнози; значення можуть включати «balltree» і «kdtree» від scikit learn, а також «brute».
  5. leafsize: це важливий параметр алгоритмів balltree і kdtree, який представляє максимальну кількість точок у листовому вузлі в структурі дерева, що використовується для представлення точок даних, перш ніж наступний підрозділ відбудеться на менші гілки в межах кожного вузла та ідентифікує найближчих сусідів. Збільшення цього значення зменшує час обчислення, але також може знизити точність результатів через меншу кількість уточнень, які відбуваються під час фільтрації відповідних найближчих сусідів на основі критеріїв метрики відстані, застосованих під час фази навчання/тестування

**Гіперпараметри Random Forest**

* 1. nestimators: цей параметр встановлює кількість дерев рішень для побудови в моделі випадкового лісу. Як правило, більше дерев означає кращу точність, але також вищі обчислювальні витрати. Збільшення кількості дерев не покращить результати, якщо дані є низької якості або переповнені.
  2. maxfeatures: цей параметр визначає кількість функцій, які використовуються під час побудови кожного дерева рішень. Більше значення підвищить точність передбачення, але також може збільшити час обчислення та використання пам’яті.
  3. minsamplessplit: мінімальна кількість зразків, необхідних для розбиття внутрішнього вузла в дереві рішень, встановлюється цим параметром, який допомагає контролювати надмірне або недостатнє налаштування. Якщо встановити занадто низько, це може призвести до великої дисперсії та перенавчання; Якщо встановити занадто високо, це може призвести до того, що модель стане занадто жорсткою та призведе до недостатнього припасування.
  4. maxdepth: Максимальна глибина кожного дерева рішень визначається цим параметром. Збільшення цього значення може підвищити точність, оскільки це дозволяє кожному дереву ставати складнішим, проектуючи більше рівнів у його гілках, але це також може сповільнити його роботу, а також потенційно призвести до перенавчання, якщо не збалансовано з іншими параметрами, такими як minsamplessplit і maxfeautres/ minsamplesleaf.
  5. minsamplesleaf: Цей параметр встановлює мінімальну кількість зразків, які повинні бути на листовому вузлі, перш ніж його можна буде розділити на дві гілки далі по структурі дерева. Збільшуючи це значення, ми зменшуємо кожен листовий вузол, що призводить до менш складної моделі. але все ще містить цінну інформацію про нашу цільову змінну(и). Як і у випадку з усіма гіперпараметрами, балансування з іншими параметрами, такими як max-features або min-sample-split, є ключовим для запобігання поведінці недостатнього/перенавчання нашого навчального набору даних моделі, а також наборів даних тестування

**Гіперпараметри SVM**

* 1. Ядро: тип ядра, що використовується для навчання та прогнозування даних. Загальні параметри включають лінійну, поліноміальну та RBF (радіальну базисну функцію).
  2. C: Параметр регуляризації, який впливає на компроміс між точністю та простотою моделі. Менше значення C призводить до простішої моделі з меншою дисперсією, тоді як більше значення C призводить до більш складної моделі з більшою дисперсією.
  3. Гама: це коефіцієнт, який використовується при виборі ядра RBF. Нижчі значення гами, як правило, призводять до більш простих моделей, тоді як вищі значення гами дозволяють моделі краще фіксувати нелінійні зв’язки.
  4. Ступінь: використовується лише при використанні поліноміального ядра; цей гіперпараметр контролює ступінь полінома, який використовується в межах рішень алгоритмів SVM.
  5. Coef0: цей термін використовується в поліноміальних ядрах для контролю того, яка частина функції враховується під час оцінювання меж у класифікаторах

## **Показники оцінювання ефективності роботи моделей**

Confusion matrix[10] надає детальну інформацію про точність моделі. Це дозволяє нам визначити, наскільки добре модель правильно ідентифікує кожен клас, а також як часто вона неправильно позначає їх. Це корисно для розуміння того, які класи є особливо складними для моделі, і дозволяє нам відповідним чином коригувати, поки ми не знайдемо оптимальне рішення.

Confusion matrix також пропонує візуальний зворотний зв’язок щодо продуктивності вашої моделі, що дозволяє легко швидко зрозуміти, де модель працює добре чи погано. Вивчаючи цю візуалізацію, ми можемо помітити тенденції неправильної класифікації та виявити потенційні проблеми швидше, ніж якби нам довелося дивитися лише на таблицю чисел.

Крім того, результати confusion matrix можна використовувати в поєднанні з іншими показниками, такими як точність і запам’ятовування, під час оцінювання точності системи машинного навчання для завдань класифікації музики[11]. Об’єднання цих показників дає нам повну картину продуктивності нашої системи, дозволяючи нам приймати більш обґрунтовані рішення щодо налаштування наших моделей для отримання кращих результатів.

* Precision – вимірює який відсоток прогнозів коректний. Точність в ідеалі повинна дорівнювати 1 для хорошого класифікатора. Точність стає 1 тільки тоді, коли чисельник і знаменник рівні, тобто TP = TP +FP, це також означає, що FP дорівнює нулю. Зі збільшенням FP значення знаменника стає більшим за чисельник, а значення точності зменшується (чого ми не хочемо).

(2.1.)

* Recall – вимірює відсоток знаходження всіх позитивних зразків. Recall також відоме як чутливість або частота істинно позитивних результатів і визначається наступним чином:

(2.2.)

В ідеалі, для хорошого класифікатора коефіцієнт Recall повинен дорівнювати 1. Recall стає 1 тільки тоді, коли чисельник і знаменник рівні, тобто TP = TP + FN, це також означає, що FN дорівнює нулю. Зі збільшенням FN значення знаменника стає більшим за чисельник, а значення пригадування зменшується (чого ми не хочемо).

* Accuracy –показує кількість правильно класифікованих екземплярів даних відзагальної кількості екземплярів даних. Accuracy не є добрим показником, коли набір даних незбалансований.

(2.3.)

Показник F1-Score це показник, який враховує як Precision, так і Recall, і визначається наступним чином:

(2.4.)

F1-Score дорівнює 1 лише тоді, коли Precision та Recall дорівнюють 1. F1-Score стає високим лише тоді, коли Precision та Recall є високими. F1-Score є середнім гармонійним значенням Precision та Recall і є кращим показником, ніж Accuracy.

# **РОЗДІЛ 3. МОДЕЛЮВАННЯ**

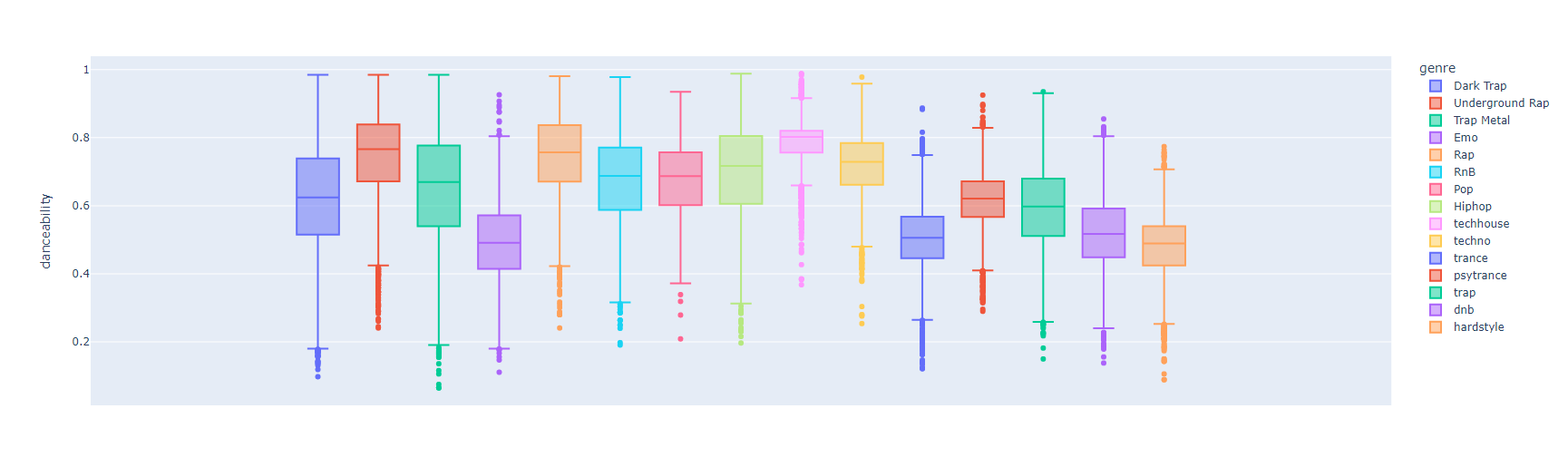
У цьому розділі буде досліджено ознаки, які можна використовувати для класифікації музики. Щоб ефективно ідентифікувати та класифікувати музику, важливо розуміти характеристики музики та її потенційний внесок у точність класифікації. Буде досліджено різні якості, які можна отримати з аудіозаписів, наприклад структуру ритму, тембр, гучність і висоту, а також те, як ці аудіохарактеристики відрізняються в різних жанрах музики. Розуміння цих якостей дозволить нам краще зрозуміти, як їх можна використовувати для точної класифікації певного музичного твору в один або кілька попередньо визначених музичних жанрів. Детальне дослідження кожної функції допоможе зрозуміти, які аспекти найкраще підходять для розрізнення музичних стилів і оптимізації продуктивності нашої моделі машинного навчання.

## **Результати аналізу та попереднього опрацювання даних**

Spotify[1] – велика музична платформа. У ньому є пісні різних жанрів. Дані, в цьому проекті, містять лише 15 жанрів. Мета полягає в тому, щоб проаналізувати всі жанри з різними характеристиками та перевірити, як вони пов’язані з жанрами.

Seaborn[12] pairplot є чудовим інструментом для аналізу набору даних, оскільки він забезпечує зручний спосіб швидко візуалізувати зв’язки між кількома змінними в даних. Його також легко налаштовувати та змінювати за допомогою вбудованих параметрів Seaborn, що дозволяє користувачам краще виділяти будь-які тенденції чи шаблони, які вони можуть знайти. Крім того, такі параметри, як відтінок і змінні, можна використовувати для швидкого розбиття даних на підмножини, що забезпечує ще більш детальну перевірку набору даних. Рисунок наявний в додатку Б.

Далі необхідно дослідити зв’язок кожної ознаки з жанрами:



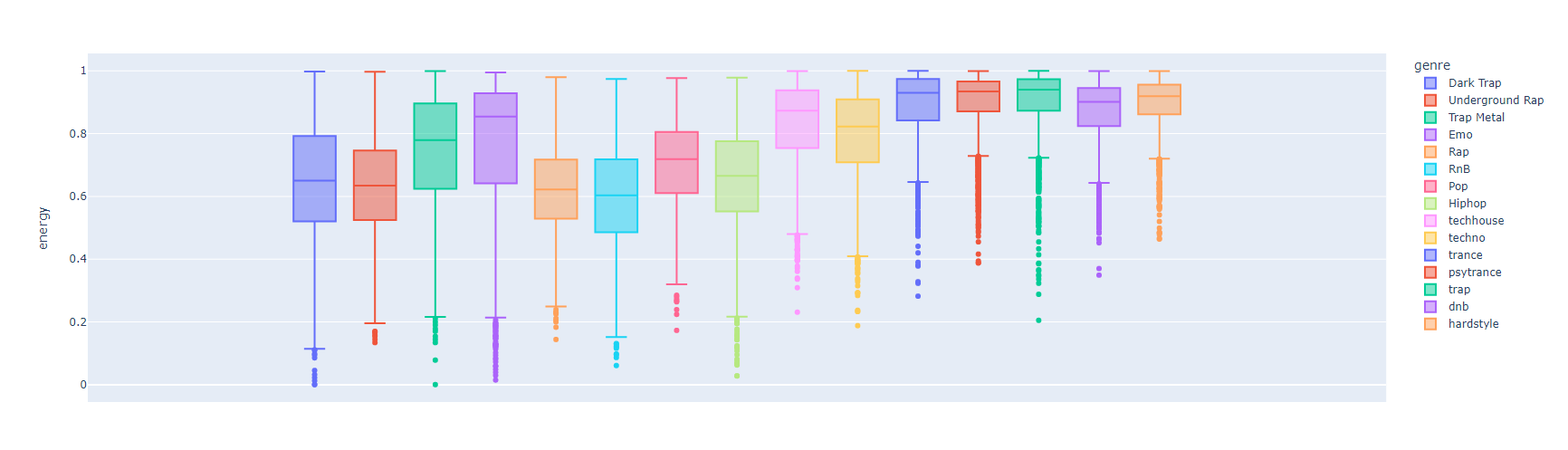
а)

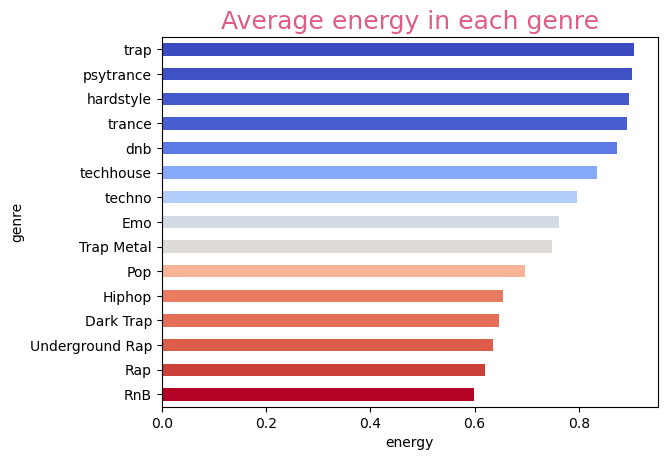


б)

Рис. 3.1. Залежність кожного жанру від 'danceability' а); б)

Найбільшу середню танцювальність мають пісні жанру «techhouse», за ними йде андеграундний реп. А хардстайлові пісні мають мінімальну танцювальність.

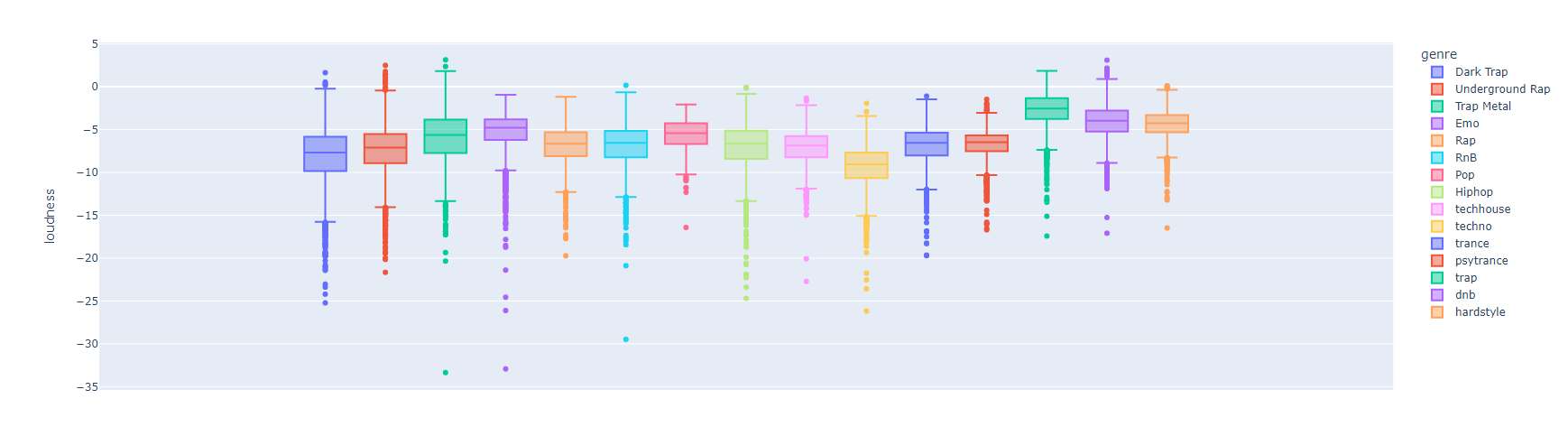
а)

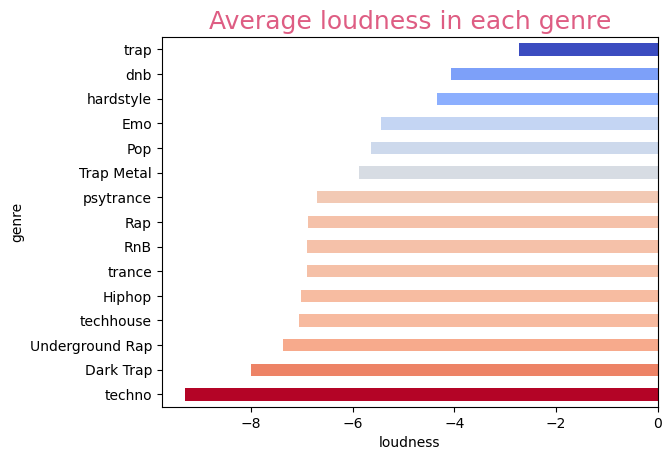


б)

Рис. 3.2. Залежність кожного жанру від 'energy' а); б)

Найвищу енергетику мають пісні жанрів треп, псі-транс, «hardstyle», транс або dnb, а найменшу RnB, Rap або Underground Rap.

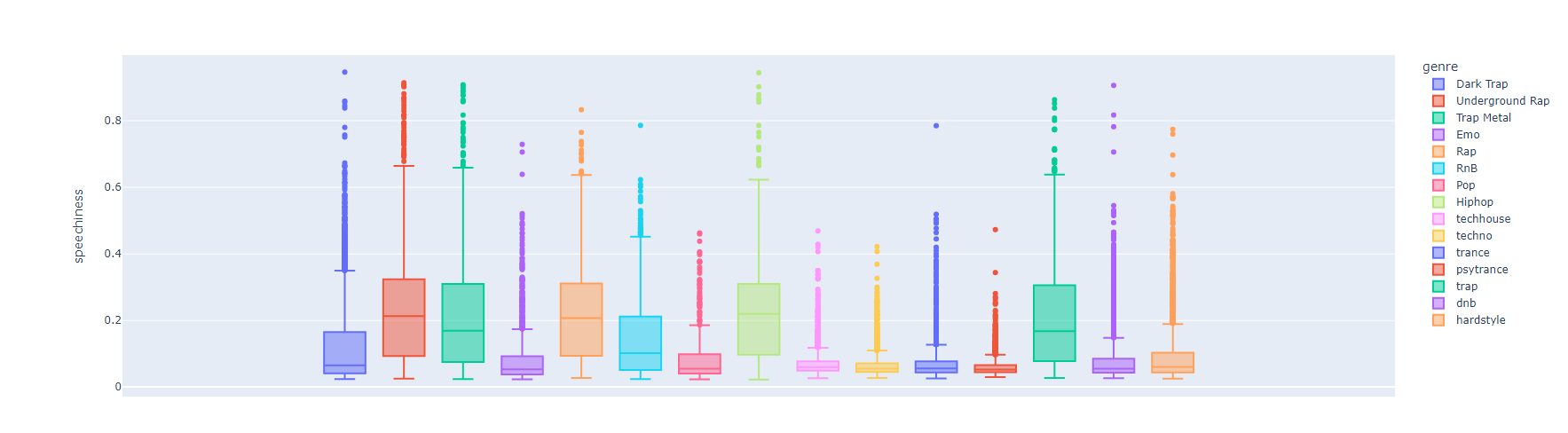
а)

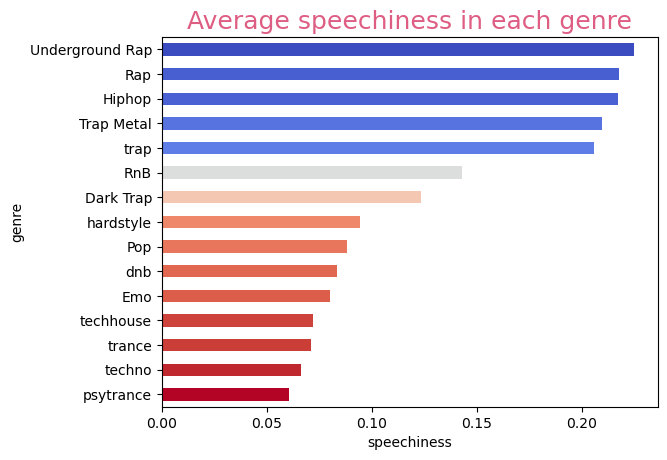


б)

Рис. 3.3. Залежність кожного жанру від 'loudness' а); б)

Пісні всіх жанрів мають негативні оцінки гучності. Найбільшу «гучність» має жанр треп, а найменшу — жанр техно.

а)

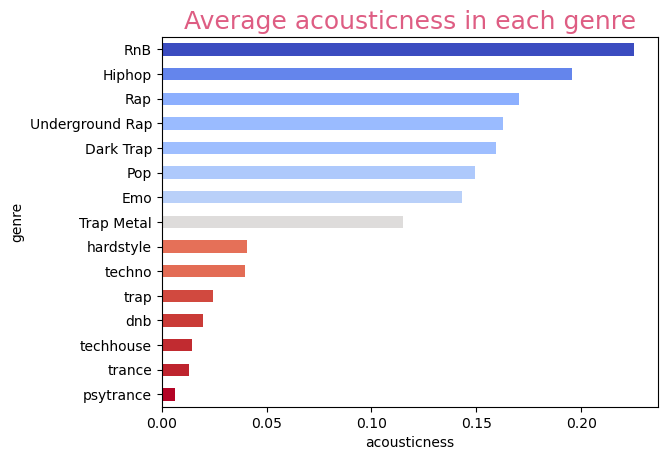


б)

Рис. 3.4. Залежність кожного жанру від 'speechiness' а); б)

Андерграунд-реп, реп і хіп-хоп займають 3 найкращі жанри за «розмовністю», а транс, техно та псі-транс – найнижчими.

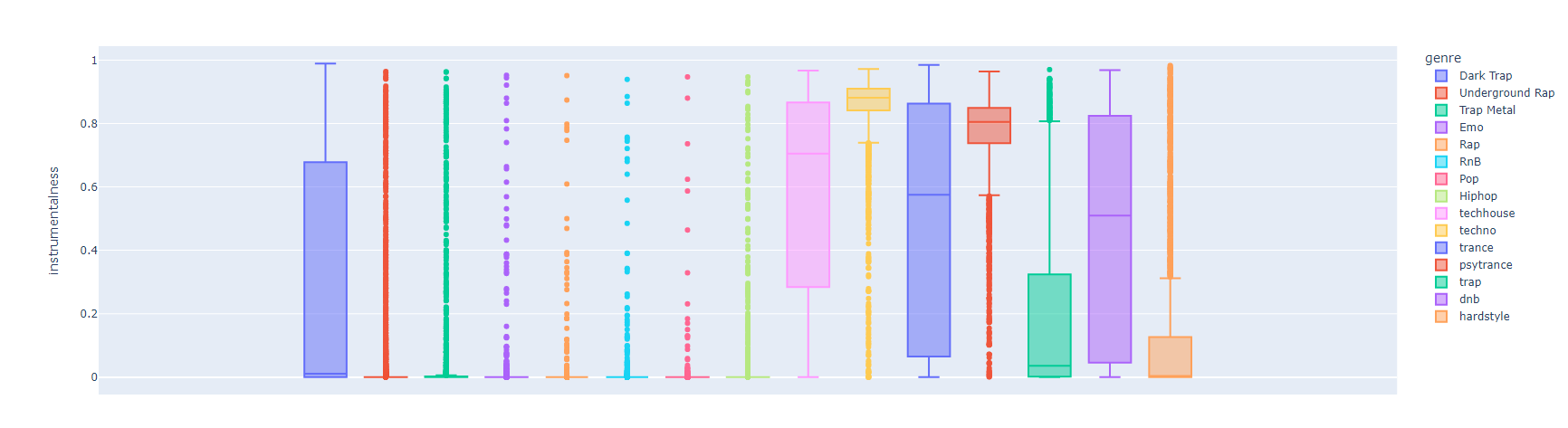
а)

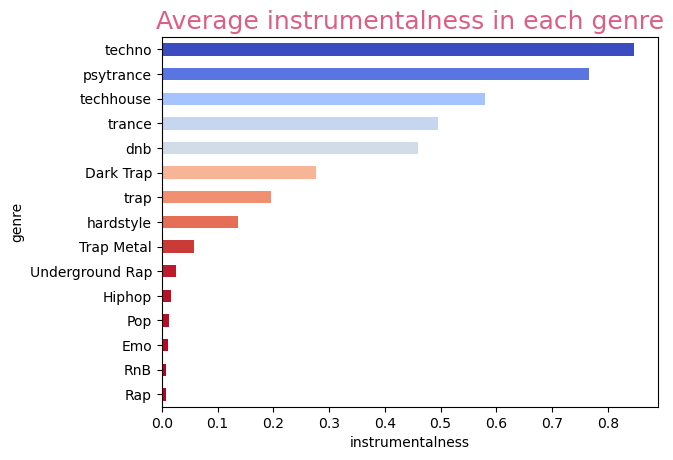


б)

Рис. 3.5. Залежність кожного жанру від 'acousticness' а); б)

Максимальний бал за акустичність мають пісні в жанрі RnB, за ними йдуть хіп-хоп і реп.

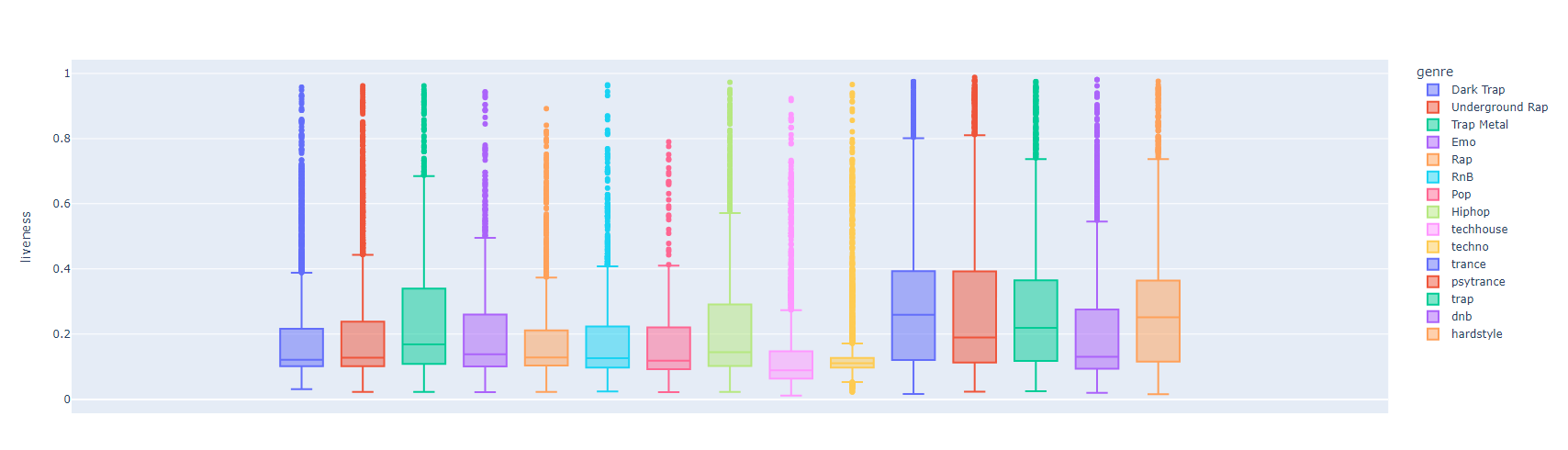
а)

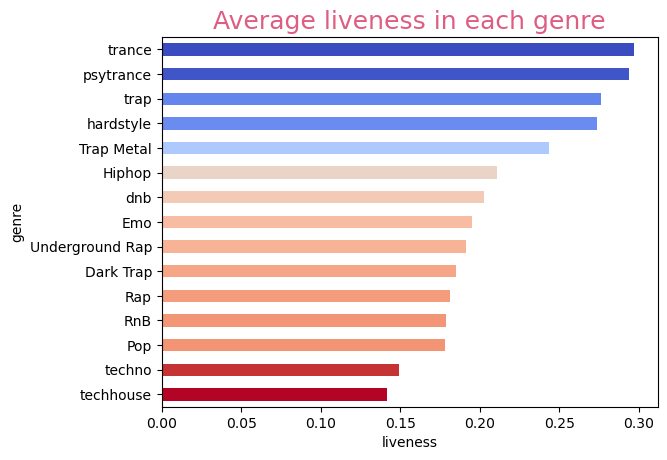


б)

Рис. 3.6. Залежність кожного жанру від 'instrumentalness' а); б)

Найбільший показник інструментальності мають пісні жанру техно, за ними йде жанр псі-транс.

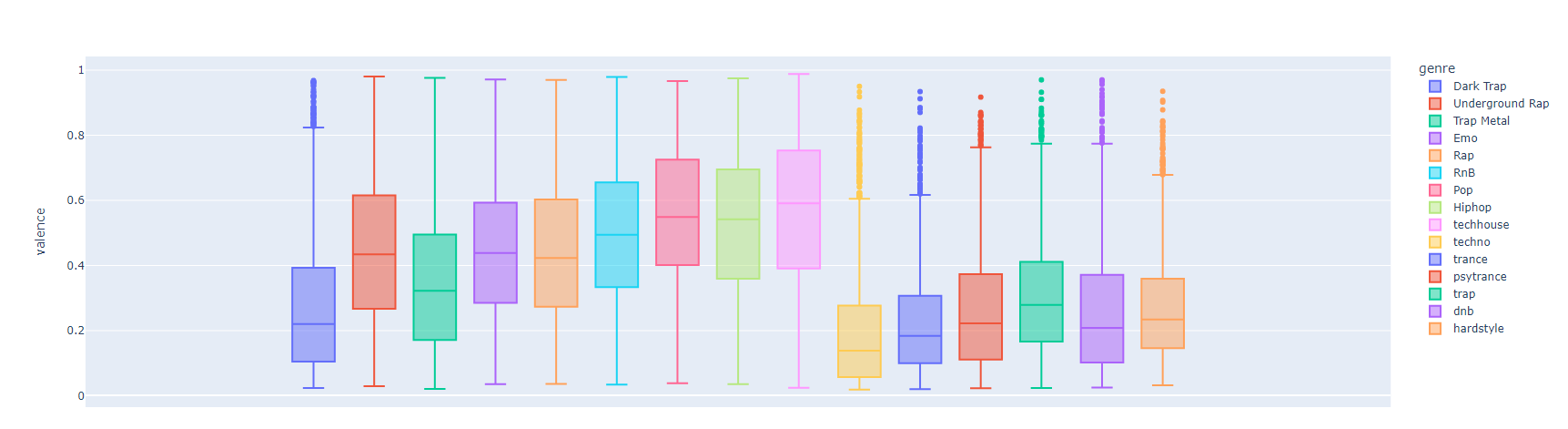
а)

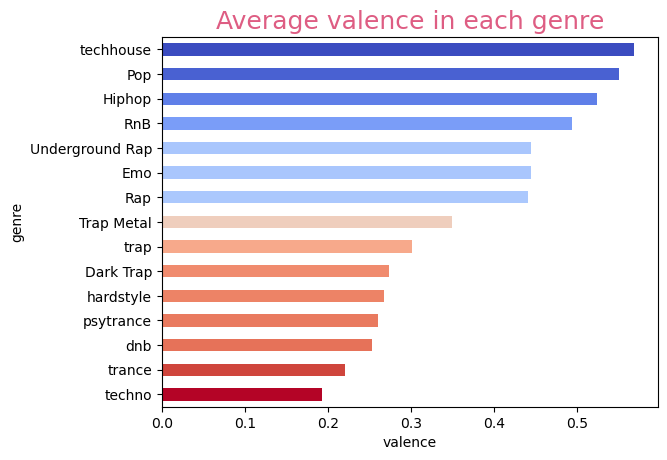


б)

Рис. 3.7. Залежність кожного жанру від 'liveness' а); б)

Транс і псітранс є найкращими жанрами з оцінками живості.

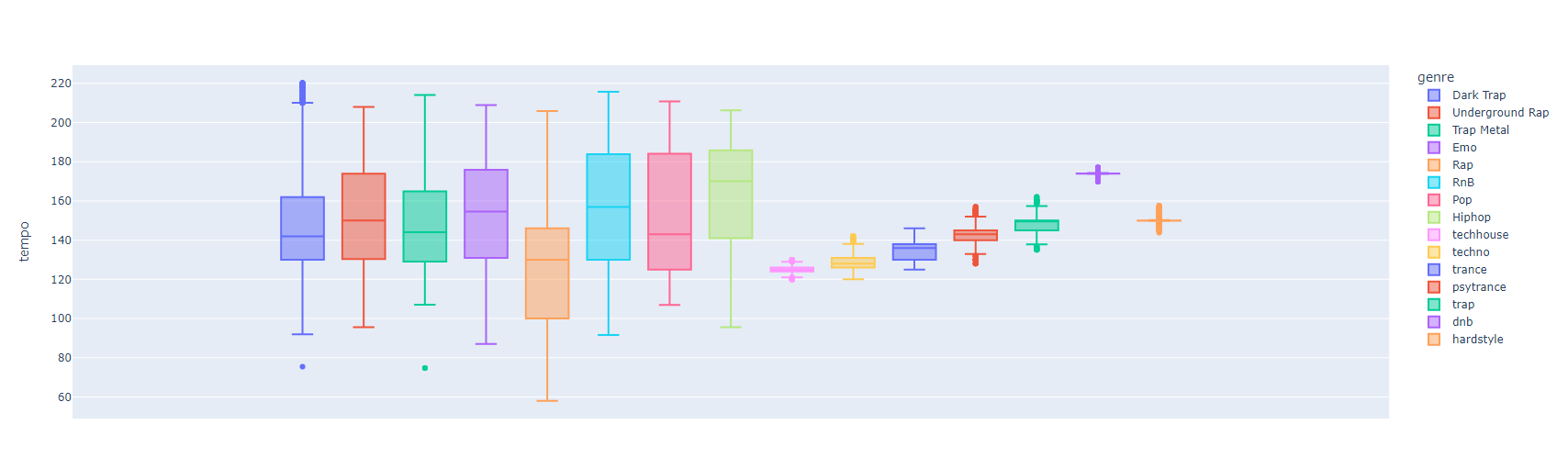
а)

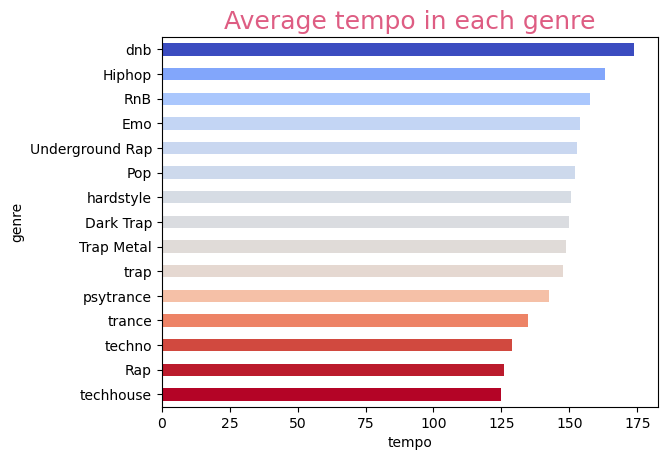


б)

Рис. 3.8. Залежність кожного жанру від 'valence' а); б)

Валентність є максимальною для жанру «techhouse», за якою йде поп, і мінімальною для техно та трансу.

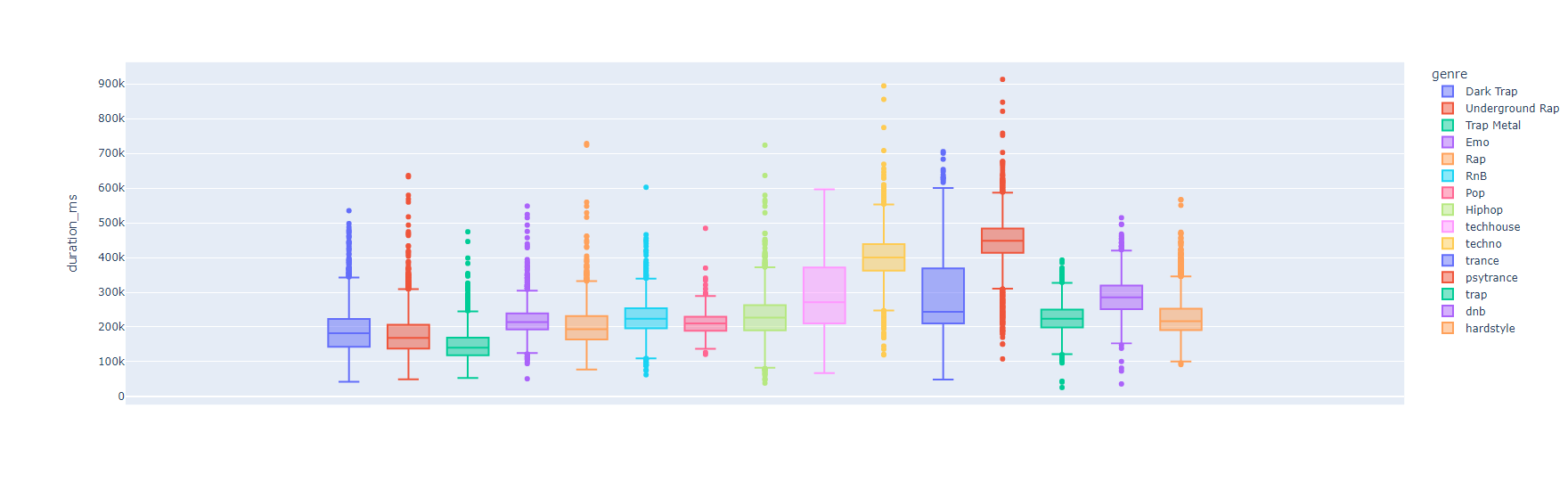
а)

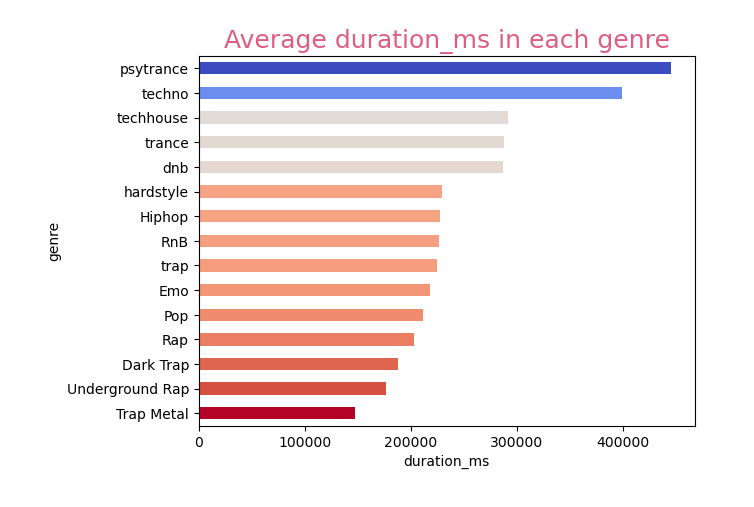


б)

Рис. 3.9. Залежність кожного жанру від 'tempo' а); б)

Темп має найвищий бал для жанру «dnb», за яким йде жанр хіп-хоп. А «techhouse», має найнижчий показник.

а)



б)

Рис. 3.10. Залежність кожного жанру від 'tempo' а); б)

Пісні в жанрах псі-транс і техно є найдовшими за всі інші жанри.

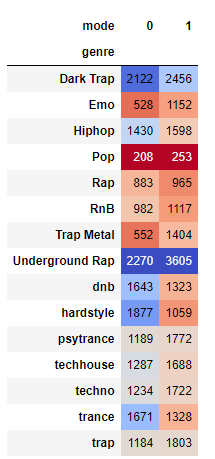


Рис. 3.11. Залежність кожного жанру від 'mode'

В обох режимах Underground Rap має максимальну кількість екземплярів.



Рис. 3.12. Залежність кожного жанру від 'time\_signature'

За винятком «time-signature» 4, усі мають максимальну кількість екземплярів у жанрі «Dark Trap». Underground Rap має максимальну кількість екземплярів у «time-signature» 4.

Після виконання дослідницького аналізу даних можна перейти до препроцесингу. Для початку переведемо колонку тривалості з мілісекунд в хвилини. Це робиться для уникнення ненормованих градієнтів, що призведе до збільшення помилки моделі.

Також пропустимо дані через StandardScaler. StandardScaler є ідеальним інструментом попередньої обробки наборів музичних даних, оскільки він дозволяє масштабувати всі значення даних до єдиного масштабу, полегшуючи їх порівняння та аналіз. Це допомагає зменшити шум і зміщення, а також спростити виявлення шаблонів або аномалій у нашому наборі даних. StandardScaler також допомагає гарантувати, що викиди не впливають на якість прогнозних результатів наших моделей, видаляючи їх з аналізу.

Колонку ж жанрів закодуємо зі значенням від 0 до n\_classes-1 з використанням LabelEncoder.

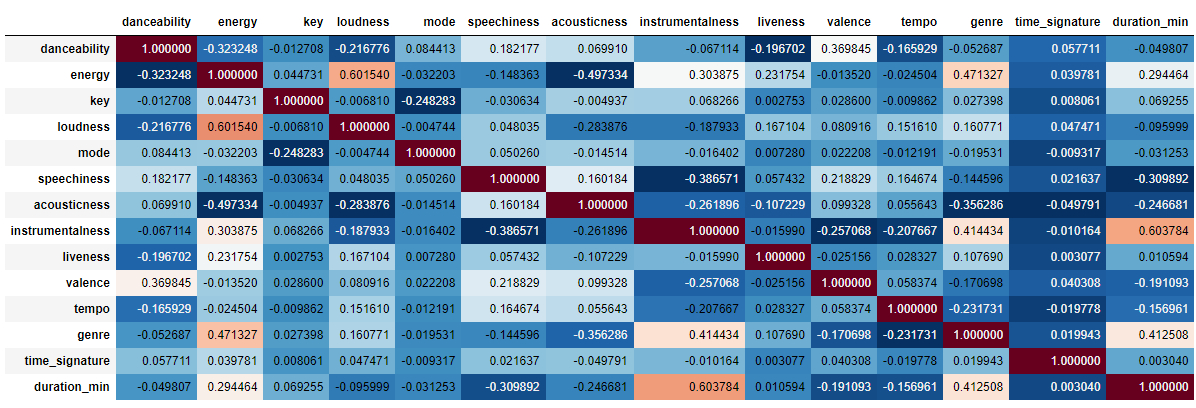
Далі проведемо кореляційний аналіз.

Рис. 3.13. Матриця кореляції ознак

З аналізу матриці кореляції можна побачити, що «танцювальність» позитивно корелює з ознакою «валентність». «Енергія» має позитивну кореляцію з «гучністю», «інструментальністю», «жвавістю» та «тривалістю». «Інструментальність» позитивно корелює з «тривалістю часу».

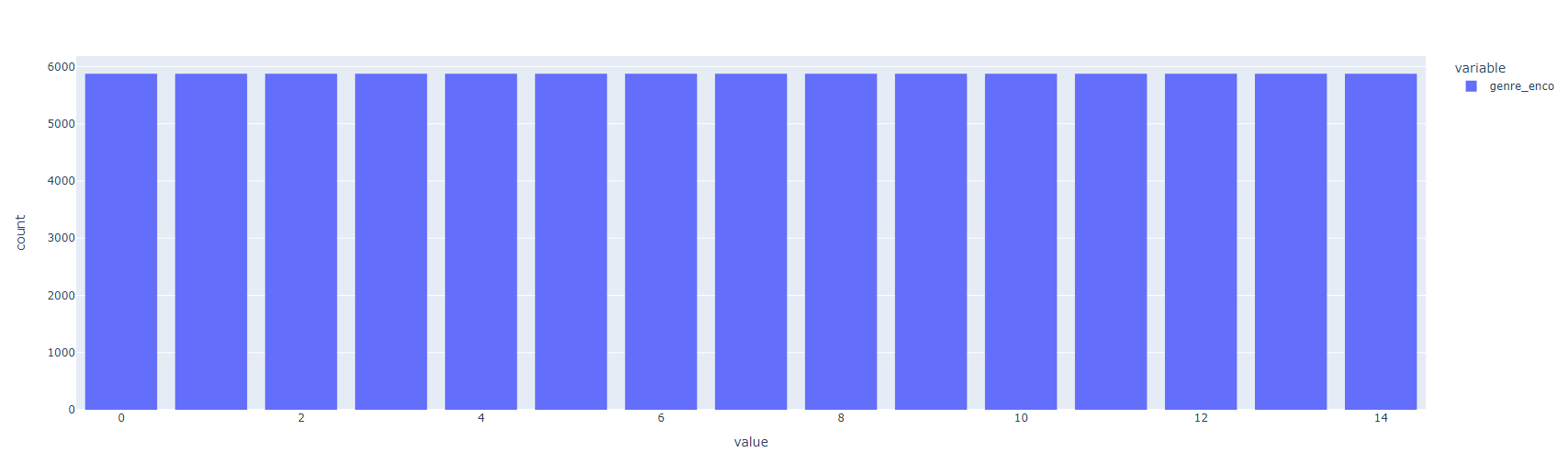
З рис. 1.5. можна побачити, що датасет не є збалансованим. Це може сильно зашкодити задачі класифікації, тому було використано SMOTE (Synthetic Minority Over-sampling Technique). SMOTE – ефективний метод попередньої обробки наборів музичних даних, оскільки він може синтезувати додаткові точки даних у наборі даних, що допомагає створити збалансований набір даних, який краще представляє всі категорії чи класи музики. Це полегшує створення та оцінку точних моделей, які можуть виявляти тонкі відмінності між музичними жанрами чи стилями. Крім того, SMOTE також може зменшити перекіс даних, викликаний рідкісними класами в наборі даних, тим самим покращуючи точність моделі. Дані розділимо на тренувальні та тестові у співвідношенні 80-20

Рис. 3.14. Розподіл даних в датасеті відносно жанру після застосування SMOTE

## **Оцінювання ефективності роботи обраних моделей та їх порівняння**

Оцінка продуктивності вибраних моделей включає оцінку того, наскільки добре дана модель працює для даної проблеми. Зазвичай це передбачає обчислення їх точності та порівняння з базовим показником. Точність в залежності від типу задачі[13]–[15] зазвичай вимірюється за допомогою таких показників, як середня абсолютна похибка (MAE), середньоквадратична похибка (RMSE), Accuracy, Precision, Recall, F1-socre та їх порівняння. Це також може включати аналіз частоти помилкових та позитивних результатів моделі (FPR). Порівняння продуктивності кількох моделей вимагає надання загальної оцінки цих показників у різних для визначення найкращого варіанту для вирішення певної проблеми.



Рис. 3.15. Confusion matrix для результатів логістичної регресії

Таблиця 3.1. Результати роботи логістичної регресії

|  |  |  |
| --- | --- | --- |
| Оптимальні параметри роботи | Показник ефективності | Значення показника ефективності логістичної регресії |
| Кількість ітерацій, для збіжності max\_iter = 1000. | Accuracy | 0.58 |
| Precision(avg) | 0.57 |
| Recall(avg) | 0.58 |
| F1-score(avg) | 0.57 |



Рис. 3.16. Confusion matrix для результатів наївного Байєса

Таблиця 3.2. Результати роботи наївного Байєса

|  |  |  |
| --- | --- | --- |
| Оптимальні параметри роботи | Показник ефективності | Значення показника ефективності наївного Байєса |
| - | Accuracy | 0.59 |
| Precision(avg) | 0.59 |
| Recall(avg) | 0.59 |
| F1-score(avg) | 0.57 |

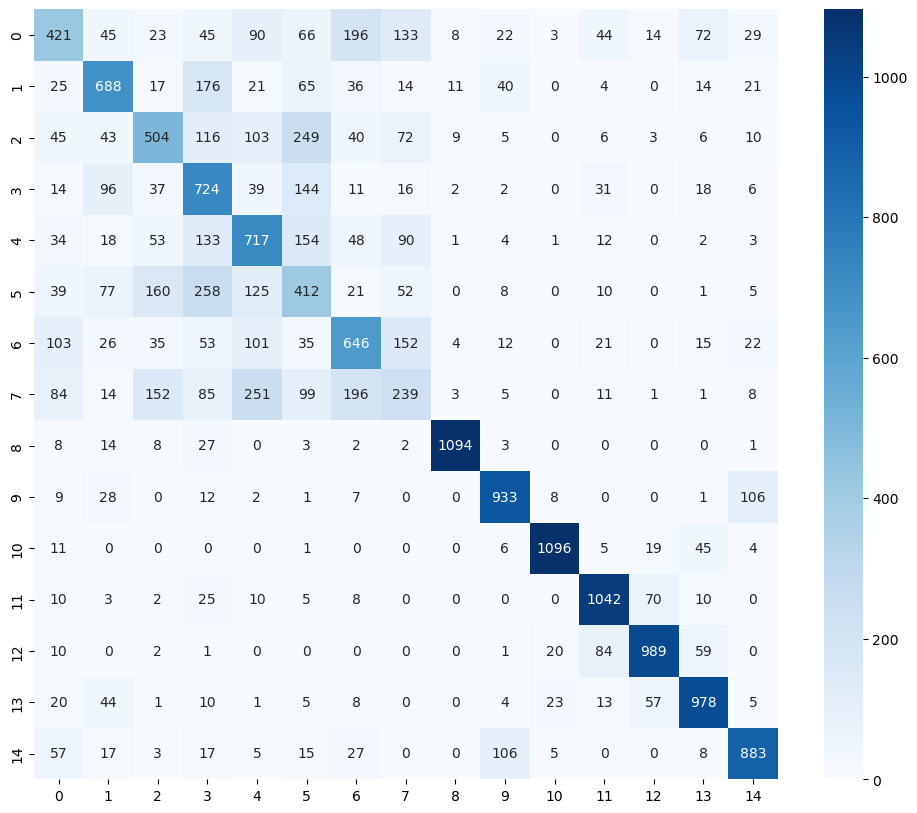


Рис. 3.17. Confusion matrix для результатів дерева рішень

Таблиця 3.3. Результати роботи дерева рішень

|  |  |  |
| --- | --- | --- |
| Оптимальні параметри роботи | Показник ефективності | Значення показника ефективності дерева рішень |
| Глибина дерева max\_depth = 4,  кількість зразків, для розбиття вузла  min\_samples\_split = 10 | Accuracy | 0.64 |
| Precision(avg) | 0.64 |
| Recall(avg) | 0.64 |
| F1-score(avg) | 0.64 |

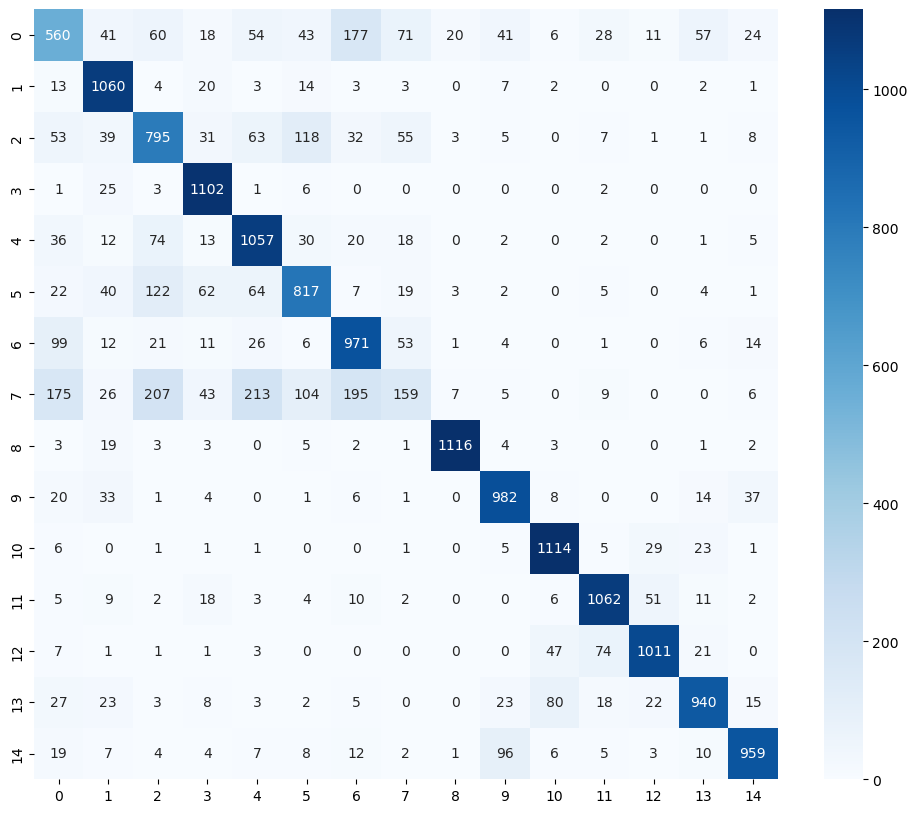


Рис. 3.18. Confusion matrix для результатів KNN

Таблиця 3.4. Результати роботи KNN

|  |  |  |
| --- | --- | --- |
| Оптимальні параметри роботи | Показник ефективності | Значення показника ефективності KNN |
| Кількість сусідів n\_neighbors = 4 | Accuracy | 0.78 |
| Precision(avg) | 0.76 |
| Recall(avg) | 0.78 |
| F1-score(avg) | 0.76 |

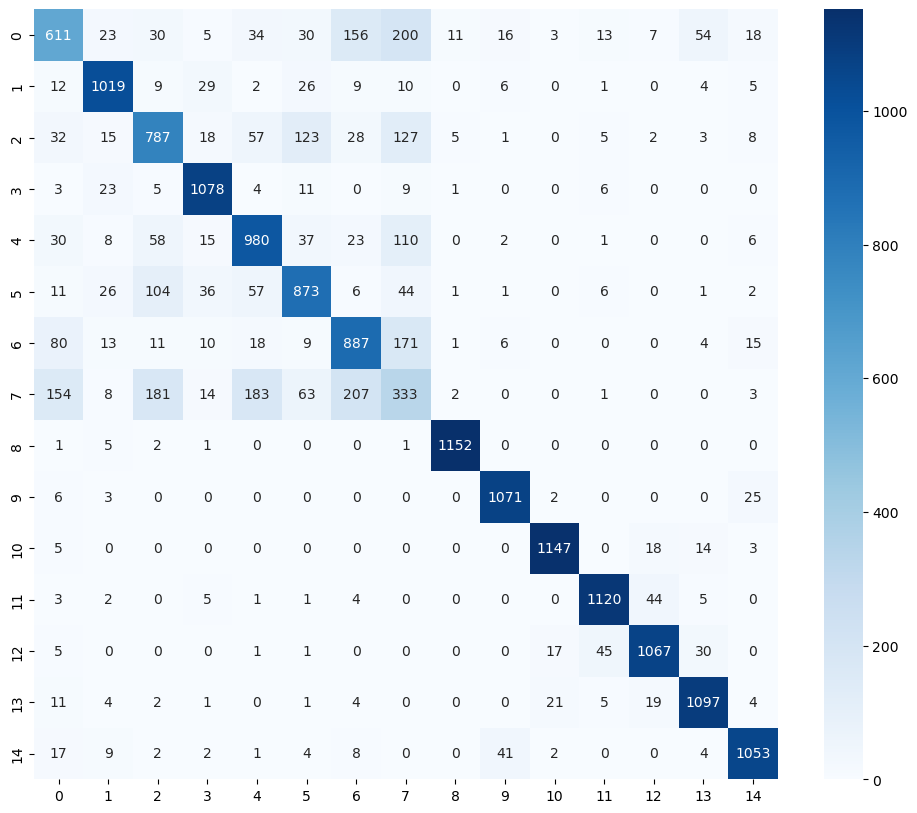


Рис. 3.19. Confusion matrix для результатів Random Forest

Таблиця 3.5. Результати роботи Random Forest

|  |  |  |
| --- | --- | --- |
| Оптимальні параметри роботи | Показник ефективності | Значення показника ефективності Random Forest |
| Кількість дерев n\_estimators = 200, максимальна глибина = 30 | Accuracy | 0.80 |
| Precision(avg) | 0.81 |
| Recall(avg) | 0.81 |
| F1-score(avg) | 0.81 |



Рис. 3.20. Confusion matrix для результатів SVM

Таблиця 3.6. Результати роботи машини опорних векторів

|  |  |  |
| --- | --- | --- |
| Оптимальні параметри роботи | Показник ефективності | Значення показника ефективності SVM |
| Ядро  kernel= ‘rbf’,  Параметр регулярізації C=1000,  Коефіцієнт ядра Gamma = scale,  Критерій допуску зупинки  tol = 0.001 | Accuracy | 0.82 |
| Precision(avg) | 0.81 |
| Recall(avg) | 0.82 |
| F1-score(avg) | 0.81 |

## **Інтерпретація результатів роботи найкращої моделі**

Отож, згідно з результатів можна зробити висновок, що найкраще в задачі класифікації музики себе показав метод SVM. Для цього методу результати F1-score та Precision становили 81%, а Accuracy – 82%. Це являється достатньо хорошим показником. Даний мето класифікації показав себе найгірше з Underground Rap(7) з показником точності в 33%, Dark Trap(0) з точність в 57% та Hiphop(2) з точністю в 65%.

В подальшому дана модель класифікації музики може бути використана для автоматичного впорядкування великих музичних колекцій відповідно до жанру, настрою та інших атрибутів. Це може спростити користувачам пошук певних типів музики або відкриття нових виконавців. Модель класифікації також може бути використана як частина механізму рекомендацій потокового сервісу або системи створення персоналізованого списку відтворення.

# **ВИСНОВКИ**

У цій роботі було подано огляд важливості музичних характеристик у сприянні прогнозу музичного жанру та обговорено, як можна розробити модель класифікації для точного виконання такого завдання. Дослідження виявило, що більшість музичних жанрів містили деякі узгоджені основні характеристики, навіть якщо в кожному жанрі були подібні класи.

Було проведення детального аналізу музичних даних, отриманих з сервісу Spotify, досліджено залежності між особливостями і жанрами та побудовано кореляційну матрицю. Також було розглянуто розподіл даних, який вказував на незбалансованість датасету. В якості препроцесингу було проведено балансування даних, за допомогою алгоритму SMOTE[16], та масштабування за допомогою стандартизації.

Для передбачення було взято 6 різних моделей класифікації, серед яких: Найкраще себе показав класифікатор SVM, який було розроблено з використанням аудіофункцій із набору даних. Він продемонстрував непогані результати з точністю в 82%. Щоб ще більше підвищити свою точність, модель може включати більше даних з інших жанрів і ширший спектр аудіофункцій для кращої результативності або розглянути можливість включення інших алгоритмів машинного навчання для досягнення ще кращих результатів.

Загалом це дослідження показує, що збір аудіо-функцій у поєднанні з методами машинного навчання створює потужний інструмент для розробки ефективних моделей класифікації музики, які відображають точні прогнози щодо жанрових міток. Крім того, ці моделі потенційно можуть допомогти у створенні більш складних автоматизованих систем рекомендацій щодо музики, коли мільйони людей покладаються на технології як частину свого повсякденного життя.

# **СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ**

[1] “Spotify — About Spotify,” *Spotify*. https://newsroom.spotify.com/company-info/ (accessed Feb. 10, 2023).

[2] “Web API Reference | Spotify for Developers.” https://developer.spotify.com/documentation/web-api/reference/ (accessed Feb. 10, 2023).

[3] T. H. PV Ramana, “Data Distributions,” *Six Sigma Study Guide*, Mar. 28, 2014. https://sixsigmastudyguide.com/data-distributions/ (accessed Feb. 10, 2023).

[4] C. Fan, M. Chen, X. Wang, J. Wang, and B. Huang, “A Review on Data Preprocessing Techniques Toward Efficient and Reliable Knowledge Discovery From Building Operational Data,” *Front. Energy Res.*, vol. 9, p. 652801, Mar. 2021, doi: 10.3389/fenrg.2021.652801.

[5] S. W. Smith, “Audio Processing,” in *Digital Signal Processing*, Elsevier, 2003, pp. 351–372. doi: 10.1016/B978-0-7506-7444-7/50059-5.

[6] M. Darji, “Audio Signal Processing: A Review of Audio Signal Classification Features,” *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, vol. 2, pp. 227–230, May 2017.

[7] T. Giannakopoulos and A. Pikrakis, “Getting Familiar with Audio Signals,” in *Introduction to Audio Analysis*, Elsevier, 2014, pp. 9–31. doi: 10.1016/B978-0-08-099388-1.00002-9.

[8] “Overview of Classification Methods in Python with Scikit-Learn.” https://stackabuse.com/overview-of-classification-methods-in-python-with-scikit-learn/ (accessed Feb. 10, 2023).

[9] “1. Supervised learning,” *scikit-learn*. https://scikit-learn/stable/supervised\_learning.html (accessed Feb. 10, 2023).

[10] Z. Karimi, “Confusion Matrix,” Oct. 2021.

[11] X. Hu, J. Downie, K. West, and A. Ehmann, “Mining Music Reviews: Promising Preliminary Results.,” Jan. 2005, pp. 536–539.

[12] M. Waskom, “seaborn: statistical data visualization,” *JOSS*, vol. 6, no. 60, p. 3021, Apr. 2021, doi: 10.21105/joss.03021.

[13] A. Mishra, “Metrics to Evaluate your Machine Learning Algorithm,” *Medium*, May 28, 2020. https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234 (accessed Feb. 10, 2023).

[14] M. Hossin and S. M.N, “A Review on Evaluation Metrics for Data Classification Evaluations,” *International Journal of Data Mining & Knowledge Management Process*, vol. 5, pp. 01–11, Mar. 2015, doi: 10.5121/ijdkp.2015.5201.

[15] A. Jierula, S. Wang, T.-M. Oh, and P. Wang, “Study on Accuracy Metrics for Evaluating the Predictions of Damage Locations in Deep Piles Using Artificial Neural Networks with Acoustic Emission Data,” *Applied Sciences*, vol. 11, no. 5, p. 2314, Mar. 2021, doi: 10.3390/app11052314.

[16] J. Wang, M. Xu, H. Wang, and J. Zhang, “Classification of Imbalanced Data by Using the SMOTE Algorithm and Locally Linear Embedding,” in *2006 8th international Conference on Signal Processing*, Guilin, China, 2006, p. 4129201. doi: 10.1109/ICOSP.2006.345752.

**ДОДАТОК А  
Програмна реалізація**

Посилання на репозиторій з кодом – <https://github.com/Edwin1349/ML-BD_project/blob/master/Untitled.ipynb>

**ДОДАТОК Б  
Комбінований графік відношень ознак розділений по жанрах**

