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»

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](#_Toc126802346)

[**РОЗДІЛ 1. ВИБІР ПРЕДМЕТНОЇ ОБЛАСТІ ТА ОПИС НАБОРУ ДАНИХ** 4](#_Toc126802347)

[**1.1.** **Постановка задачі** 5](#_Toc126802348)

[**1.2.** **Опис набору даних** 6](#_Toc126802349)

[**1.3.** **Процедури попереднього опрацювання даних** 13](#_Toc126802350)

[**РОЗДІЛ 2. ДОСЛІДНИЦЬКИЙ РОЗДІЛ** 18](#_Toc126802351)

[**2.1.** **Опис та обгрунтування обраних моделей машинного навчання** 18](#_Toc126802352)

[**2.2.** **Процедури підбору оптимальних параметрів роботи моделей** 21](#_Toc126802353)

[**2.3.** **Показники оцінювання ефективності роботи моделей** 21](#_Toc126802354)

[**РОЗДІЛ 3. МОДЕЛЮВАННЯ** 23](#_Toc126802355)

[**3.1.** **Результати аналізу та попереднього опрацювання даних** 23](#_Toc126802356)

[**3.2.** **Оцінювання ефективності роботи обраних моделей та їх порівняння** **Error! Bookmark not defined.**](#_Toc126802357)

[**3.3.** **Інтерпретація результатів роботи найкращої моделі** 39](#_Toc126802358)

[**ВИСНОВКИ** 40](#_Toc126802359)

[**СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ** 41](#_Toc126802360)

[**ДОДАТОК А Програмна реалізація** 42](#_Toc126802361)

[**ДОДАТОК Б Програмна реалізація** 43](#_Toc126802362)

# **ВСТУП**

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

Аналіз необроблених музичних даних відіграє важливу роль у завантаженні треків у цифрові бібліотеки та потокові сервіси. Аналізуючи такі точки даних, 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 безцінним надбанням для багатьох аспектів створення та зберігання музики. Завдяки складним програмним забезпеченням, які фіксують складні параметри зі звуків, відтворених вручну, у поєднанні з алгоритмами машинного навчання для розпізнавання шаблонів у кількох зразках інженери можуть точніше ідентифікувати інструменти та створювати легкозасвоювані бібліотечні системи. Покращуючи доступність і організацію музичного вмісту, аудіоінженери надають слухачам безпрецедентний доступ до різноманітних наборів звукових пейзажів без шкоди для високоякісного звуку чи можливостей точного пошуку.

Завантаження музичних даних передбачає доступ до онлайнової бази даних потокового сервісу, наприклад Spotify. Дані зазвичай складаються з треків, виконавців і альбомів разом із пов’язаними з ними характеристиками, такими як жанр, популярність виконавця та дата випуску. 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. Деякі з них включають визначення популярних тенденцій з часом як у жанрах, так і в окремих піснях; визначення кластерів або «сцен», де певні стилі є особливо впливовими.

## **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.

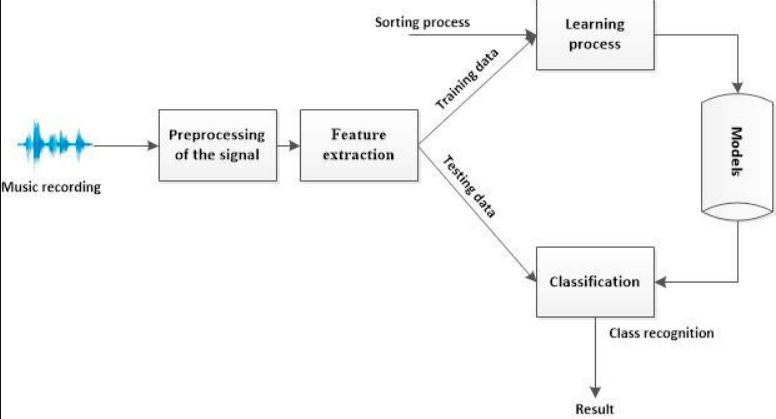


Рис. 1.1. Приклад оформлення рисунку

## **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:

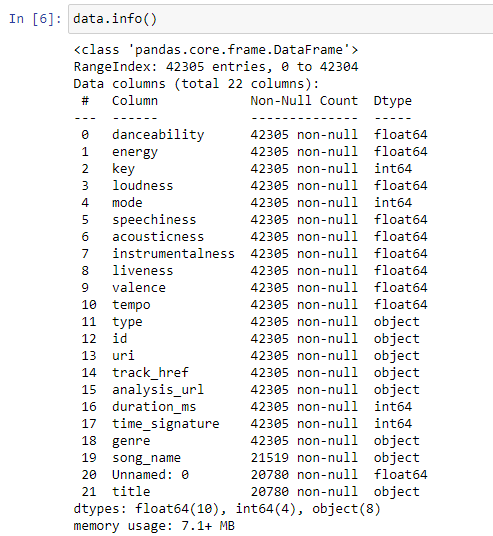
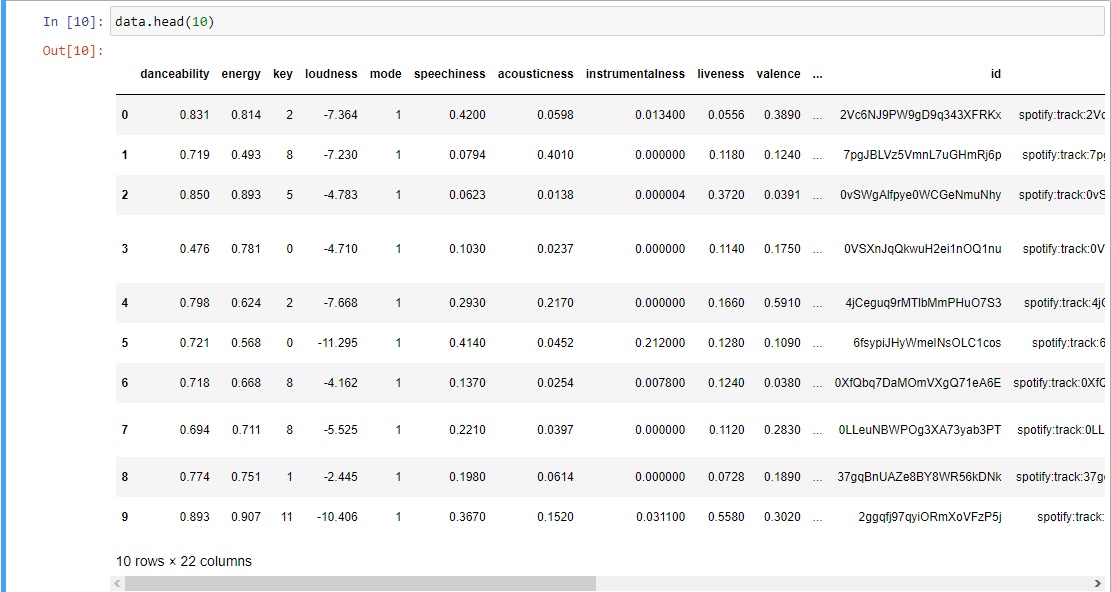


Рис. 1.2. Загальгна інформація про дані

* '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;



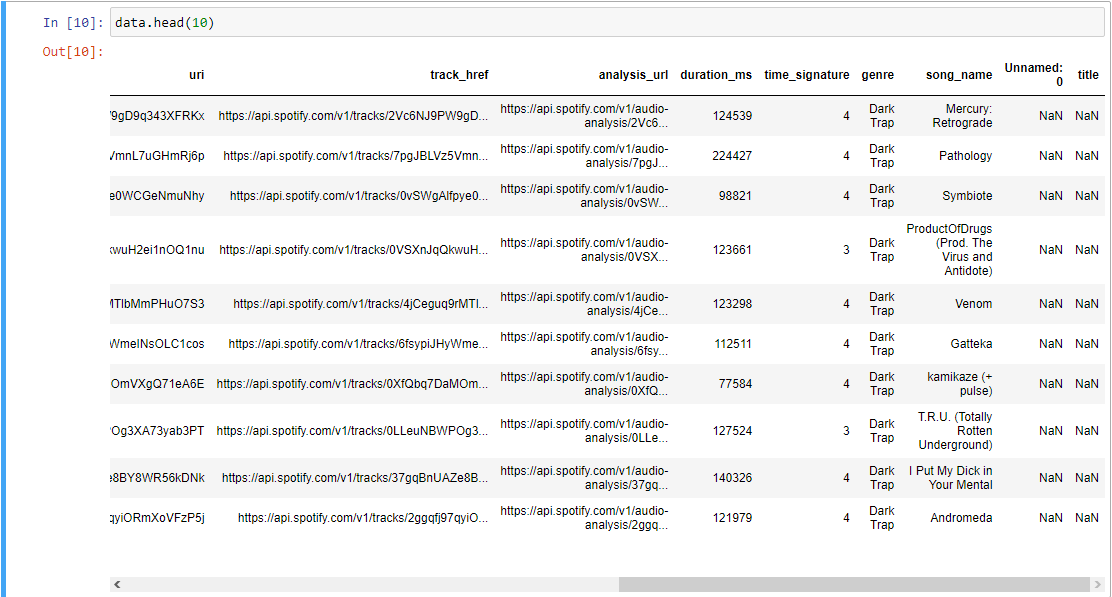
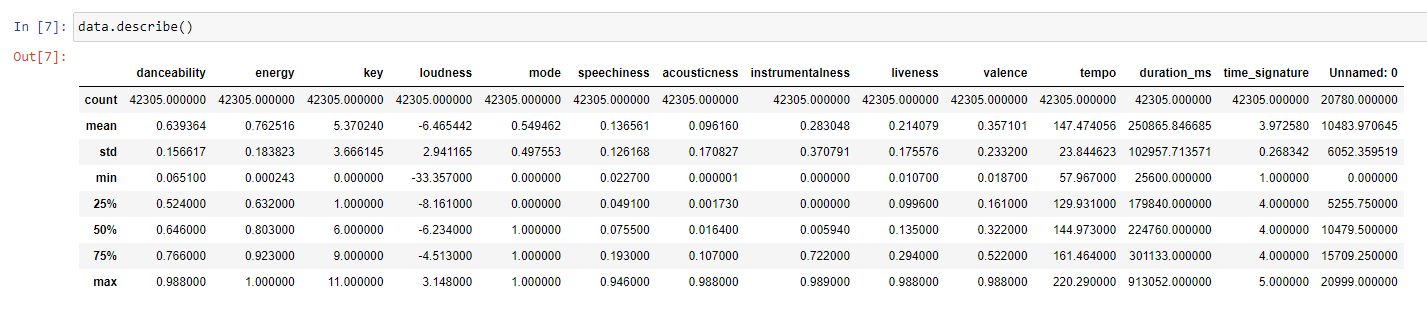


Рис. 1.3. Загальний вигляд датасету

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

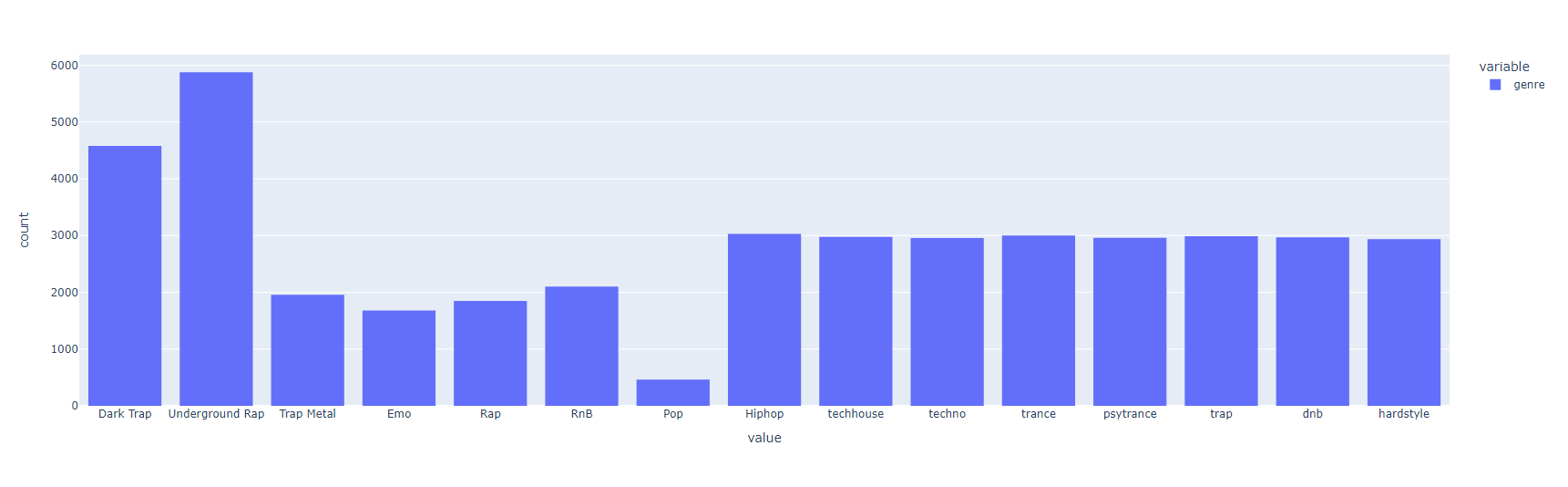
Аналіз розподілу даних є важливою частиною аналізу даних, оскільки він допомагає краще зрозуміти набір даних і його характеристики. Аналізуючи розподіл значень між різними змінними в наборі даних, ми можемо краще ідентифікувати викиди або неочікувані результати, а також виявити значущі тенденції та закономірності. 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

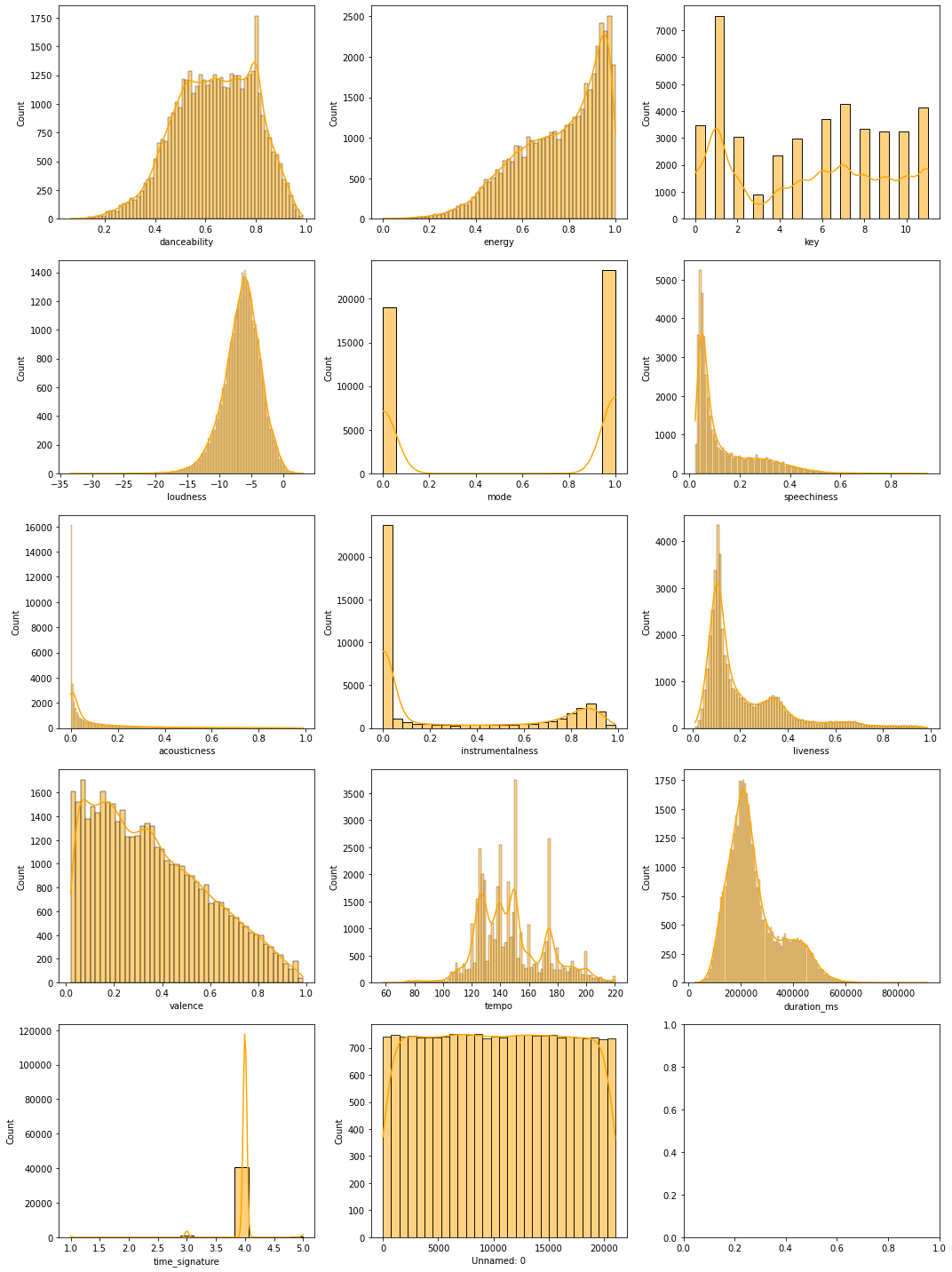


Рис. 1.6. Розподіл даних в датасеті по кожній ознаці

## **Процедури попереднього опрацювання даних**

Препроцесинг 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**

Текст текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст.

## **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.

Першою перевагою використання k-найближчого сусіда (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 випадкових лісів для класифікації музики є його здатність обробляти багатовимірні дані. Один випадковий ліс 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 переобладнання, оскільки вони покладаються на початкову вибірку, яка бере повторні вибірки з вашого набору даних із заміною. Ця техніка ще більше підвищує точність моделі, уникаючи переобладнання. Використання мішків (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 опорної векторної машини (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, такими як мережі perceptron, SVM не тільки ідеально підходять для великих наборів даних, але й вимагають низьких накладних витрат пам’яті, що робить їх 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 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.Текст текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст.

## **Показники оцінювання ефективності роботи моделей**

Confusion matrix 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. 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.

(1.1.)

* Recall – вимірює відсоток знаходження всіх позитивних зразків.

(1.2.)

* Accuracy – вимірює відсоток знаходження всіх позитивних зразків.

(1.3.)

# **РОЗДІЛ 3. МОДЕЛЮВАННЯ**

Текст текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст.

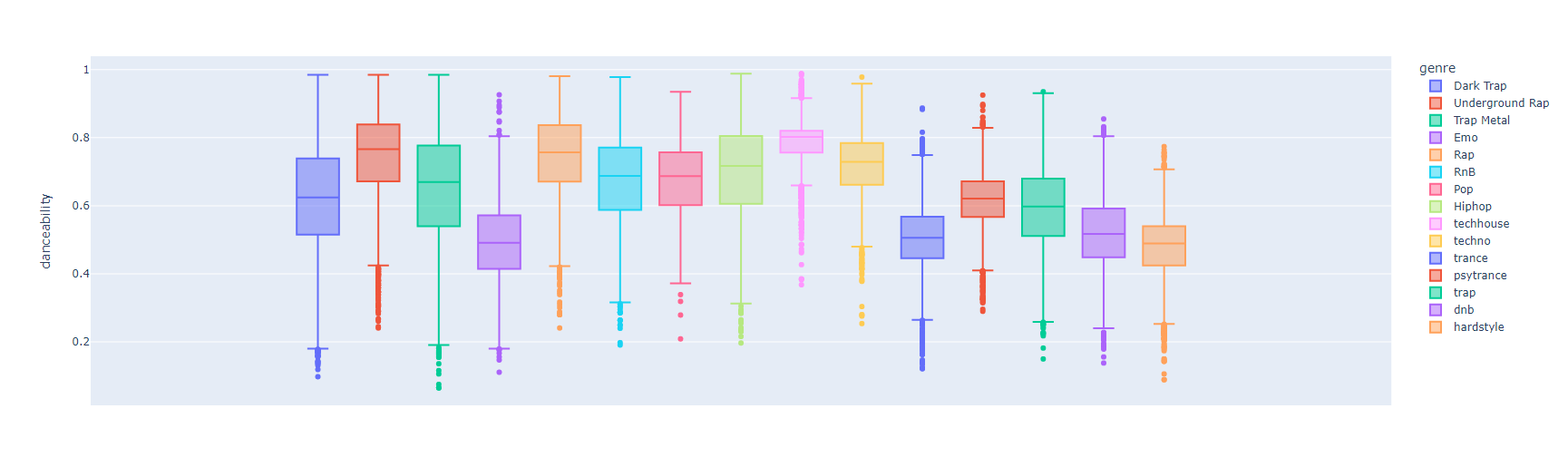
## **Результати аналізу та попереднього опрацювання даних**

Текст текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст.

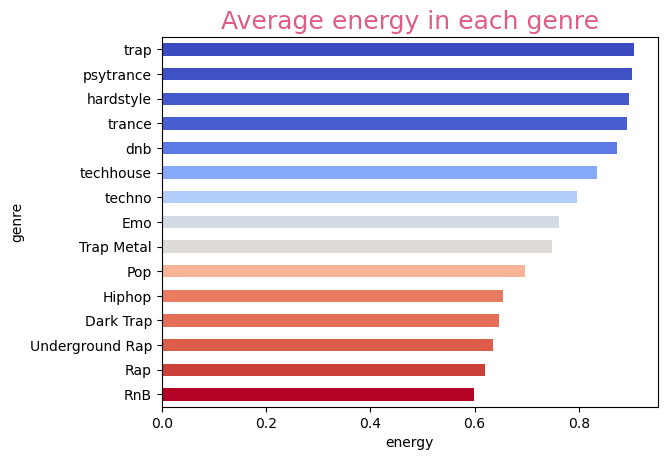
Seaborn pairplot є чудовим інструментом для аналізу набору даних, оскільки він забезпечує зручний спосіб швидко візуалізувати зв’язки між кількома змінними в даних. Його також легко налаштовувати та змінювати за допомогою вбудованих параметрів Seaborn, що дозволяє користувачам краще виділяти будь-які 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.

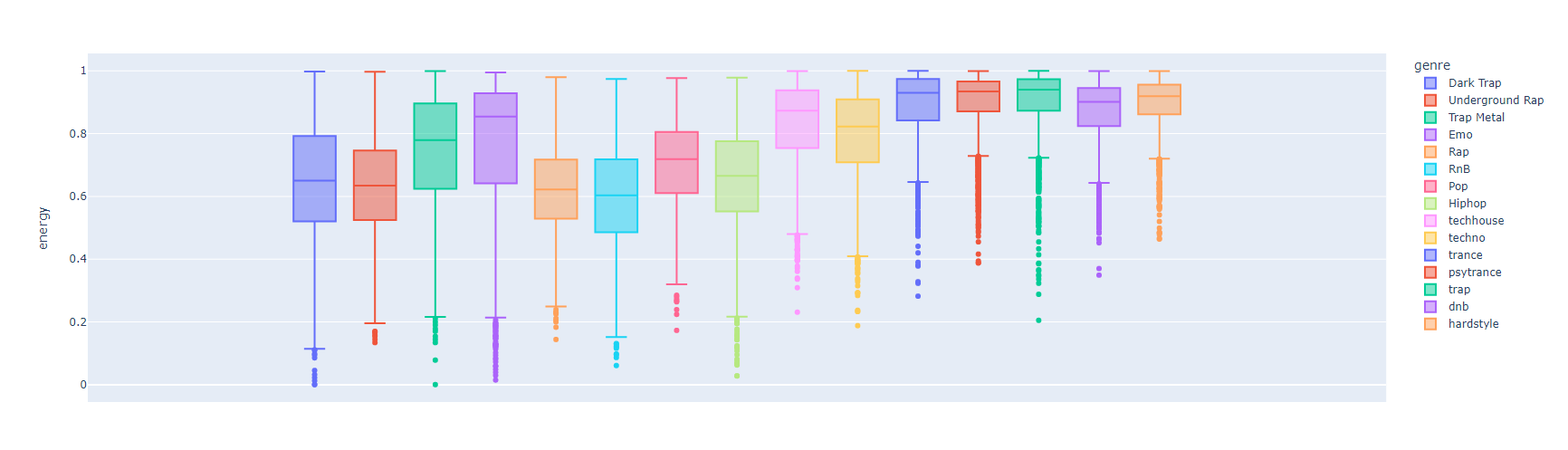




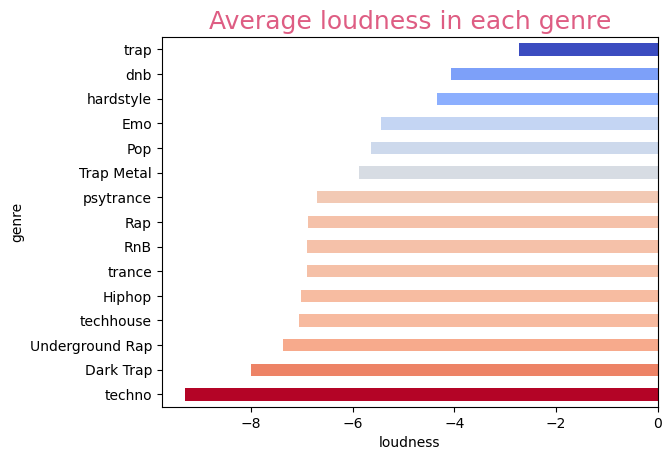


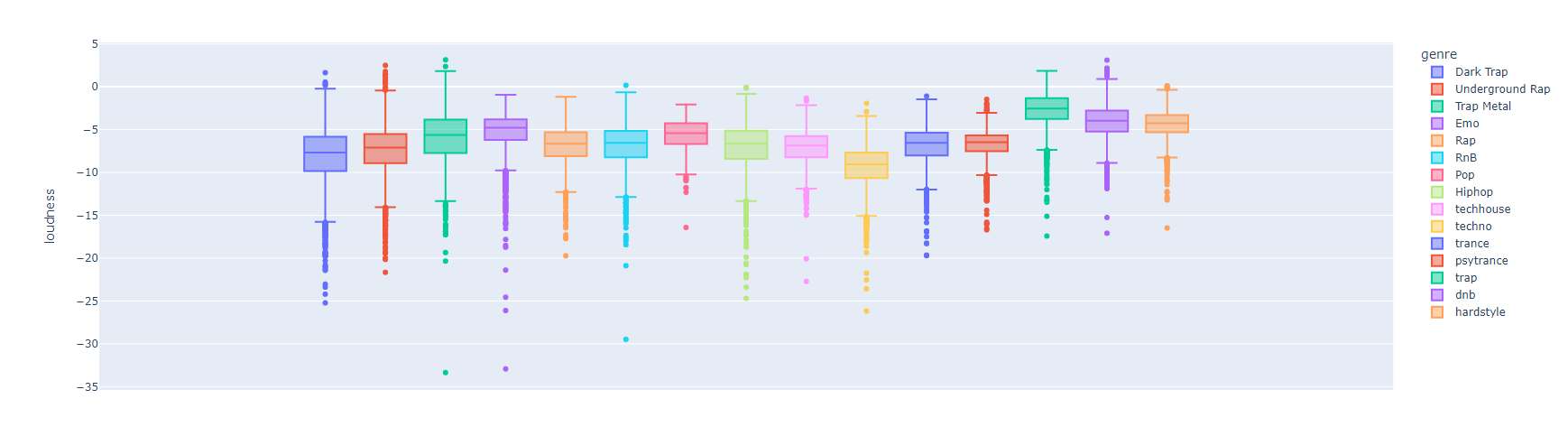
Найбільшу середню танцювальність мають пісні жанру «techhouse», за ними йде андеграундний реп. А хардстайлові пісні мають мінімальну танцювальність.



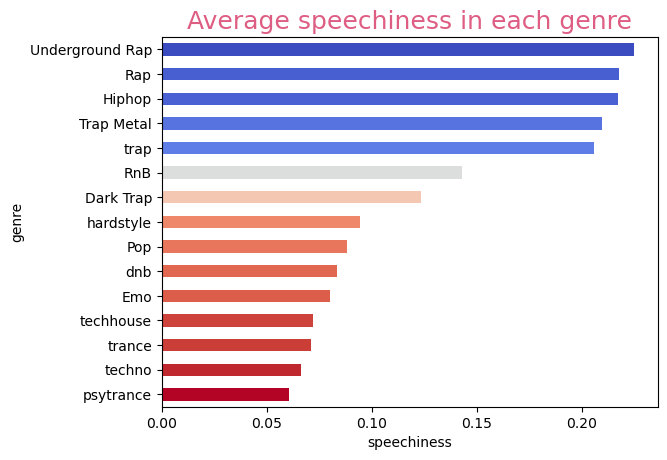


Найвищу енергетику мають пісні жанрів треп, псі-транс, «hardstyle», транс або dnb, а найменшу RnB, Rap або Underground Rap.

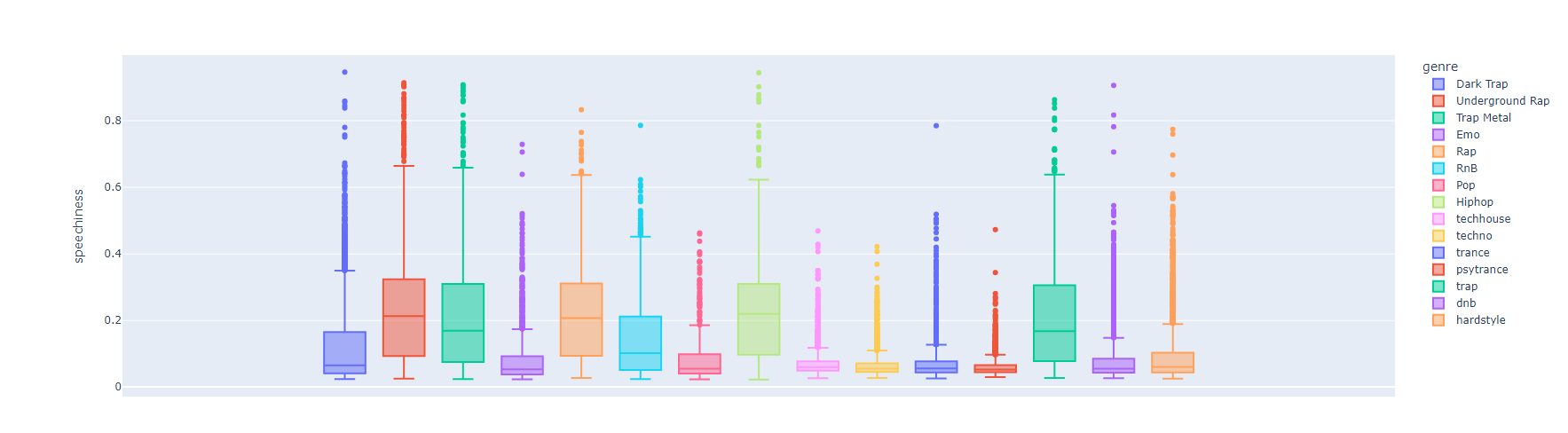


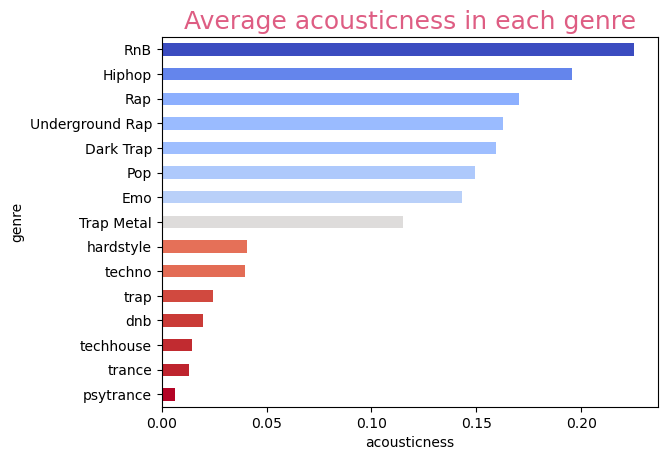


Пісні всіх жанрів мають негативні оцінки гучності. Найбільшу «гучність» має жанр треп, а найменшу — жанр техно.



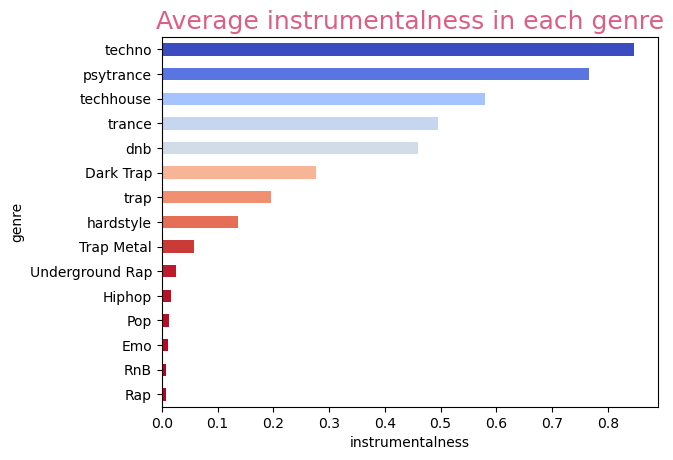
Андерграунд-реп, реп і хіп-хоп займають 3 найкращі жанри за «розмовністю», а транс, техно та псі-транс – найнижчими.

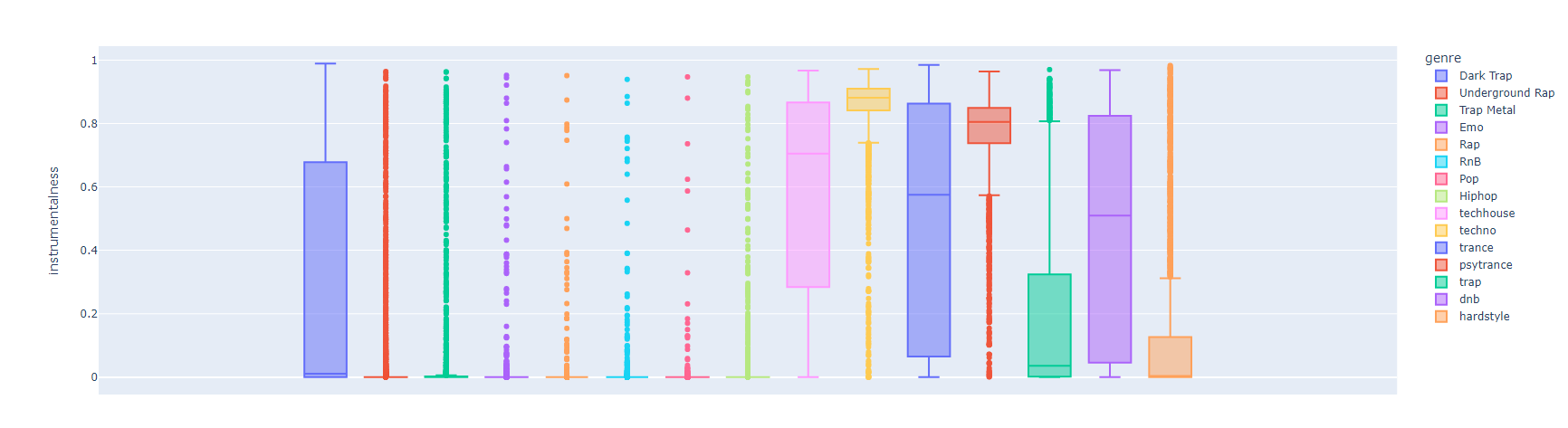




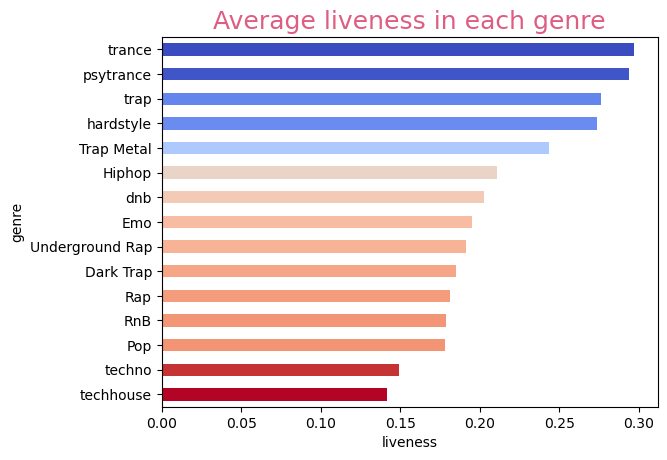


Максимальний бал за акустичність мають пісні в жанрі RnB, за ними йдуть хіп-хоп і реп.

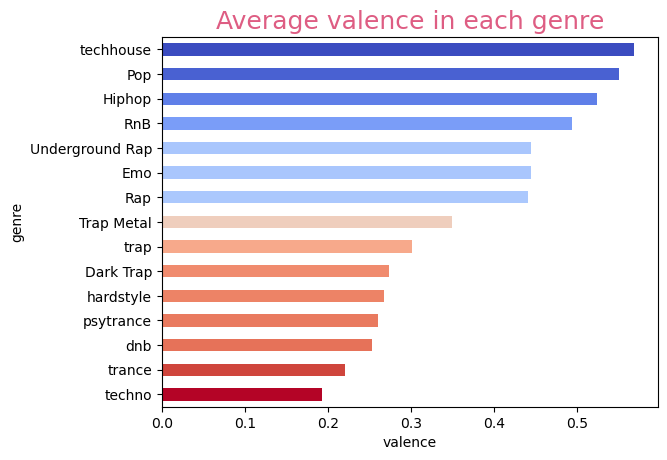
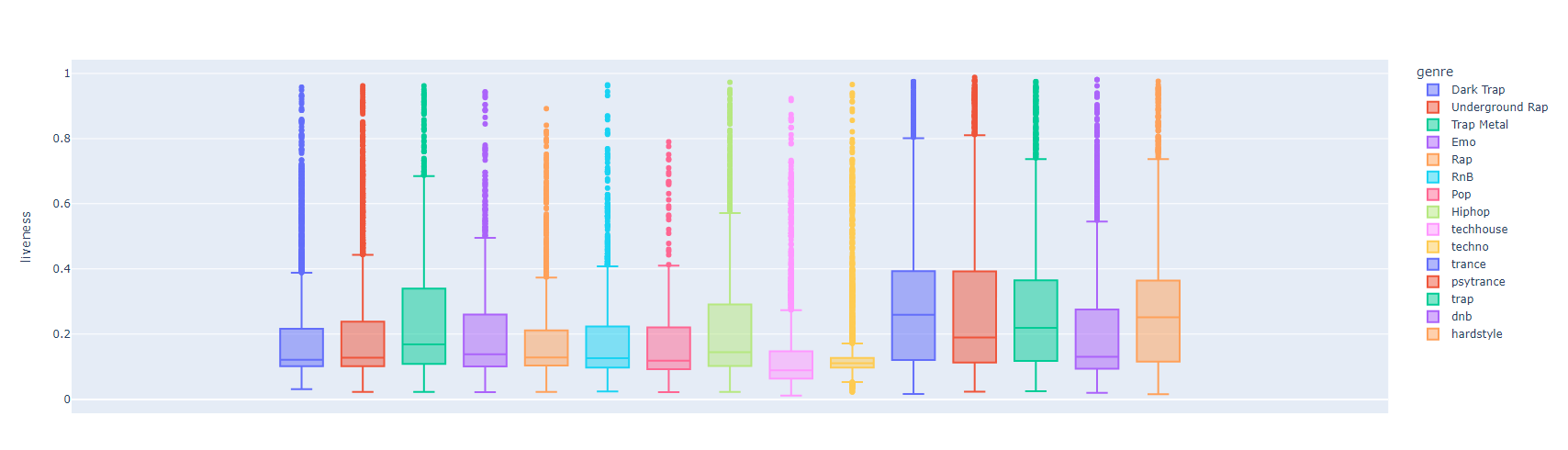


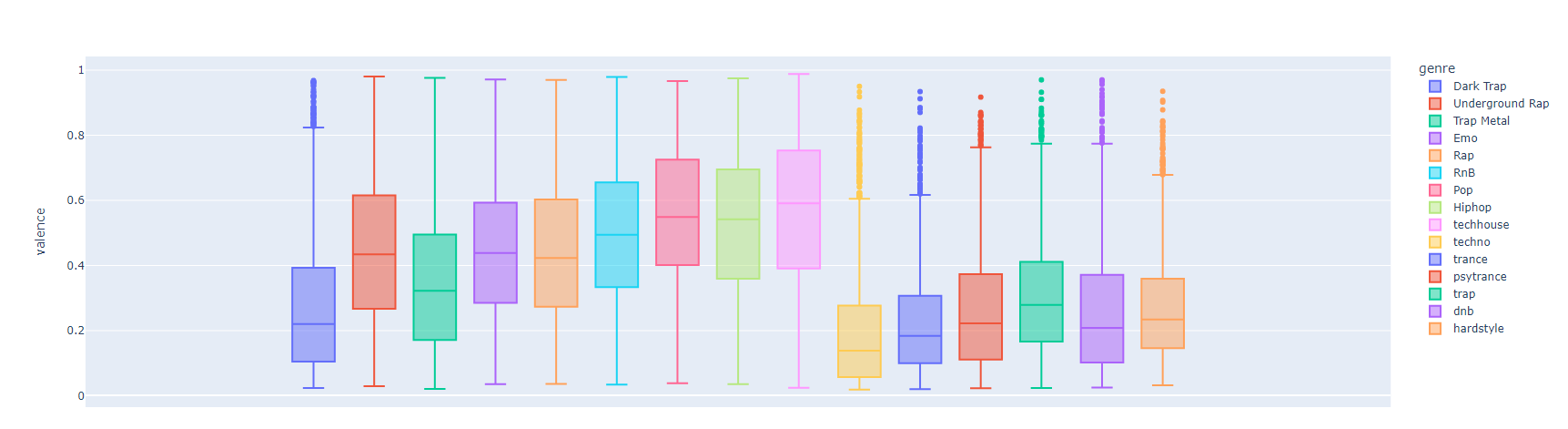


Найбільший показник інструментальності мають пісні жанру техно, за ними йде жанр псі-транс.

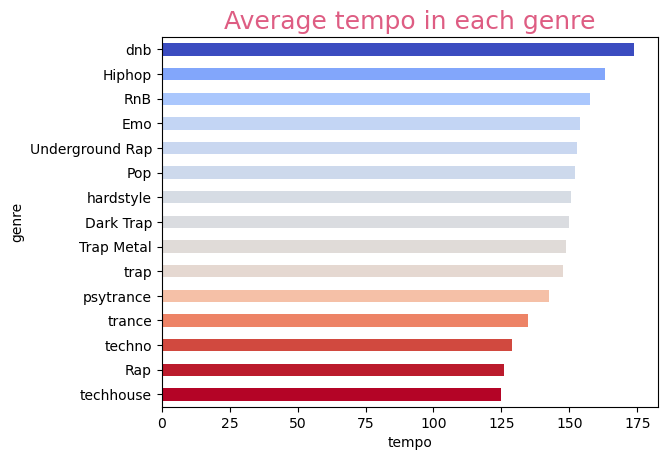


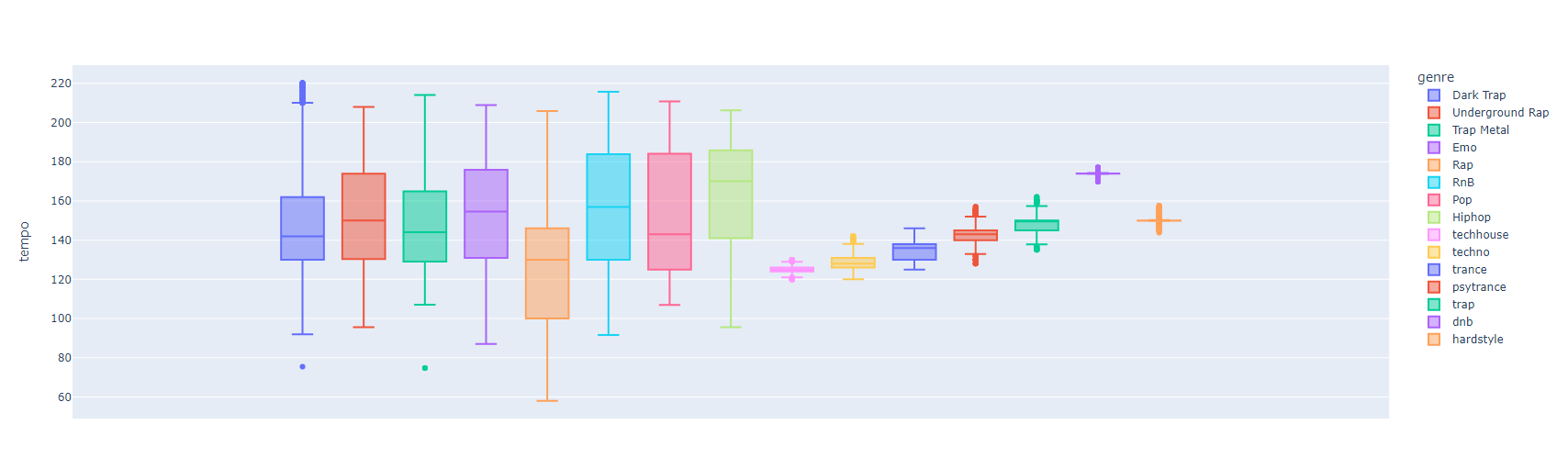
Транс і псітранс є найкращими жанрами з оцінками живості.



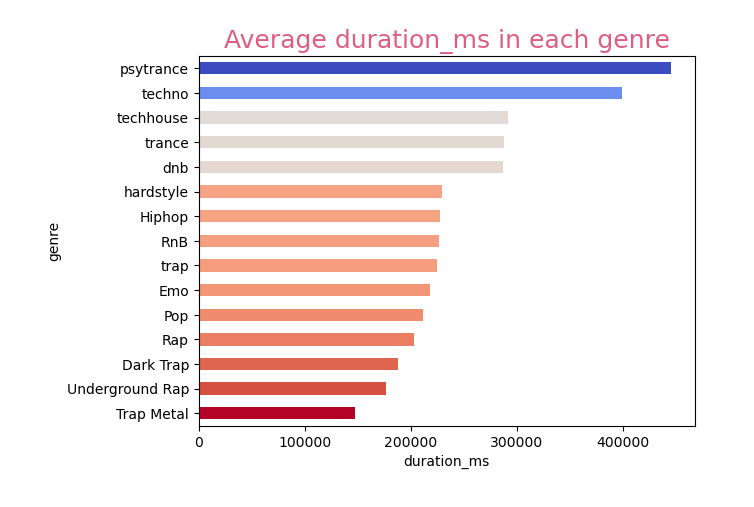


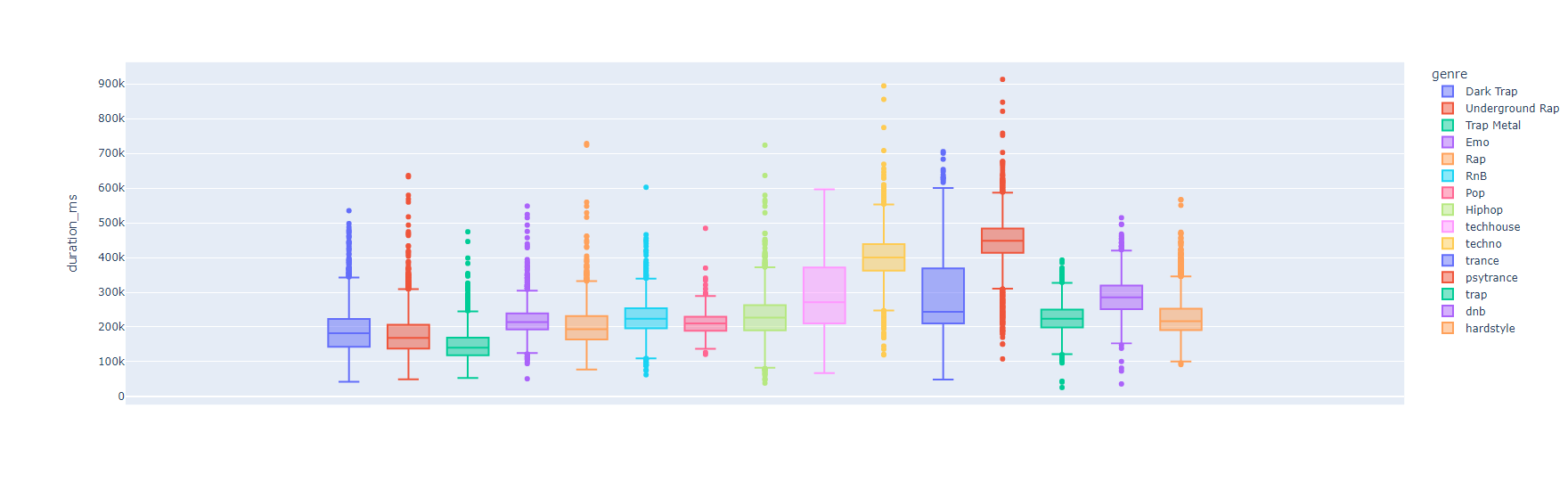
Валентність є максимальною для жанру «techhouse», за якою йде поп, і мінімальною для техно та трансу.

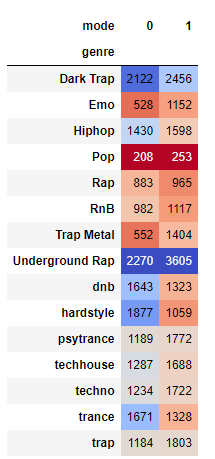




Темп має найвищий бал для жанру «dnb», за яким йде жанр хіп-хоп. А «techhouse», має найнижчий показник.



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



В обох режимах Underground Rap має максимальну кількість екземплярів.



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 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.

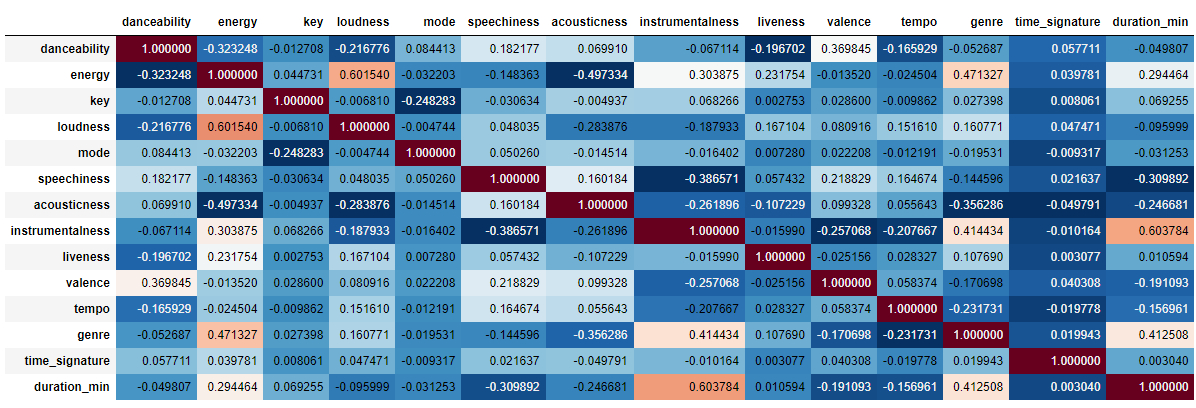
Далі проведемо кореляційний аналіз.

Рис. 3.10. 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 перекос даних, викликаний рідкісними класами в наборі даних, тим самим покращуючи точність моделі.

Рис. 3.10. Розподіл даних в датасеті відносно жанру після застосування SMOTE

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

Текст текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст. 

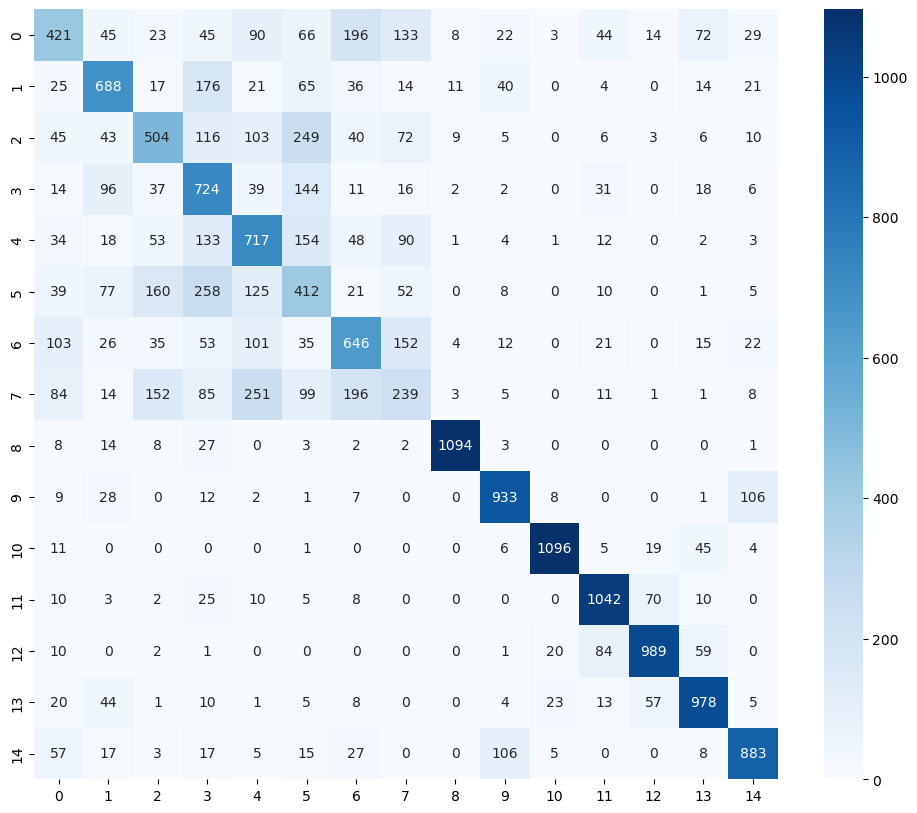
Таблиця 3.1. Результати роботи логістичної регресії

|  |  |  |
| --- | --- | --- |
| 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 |



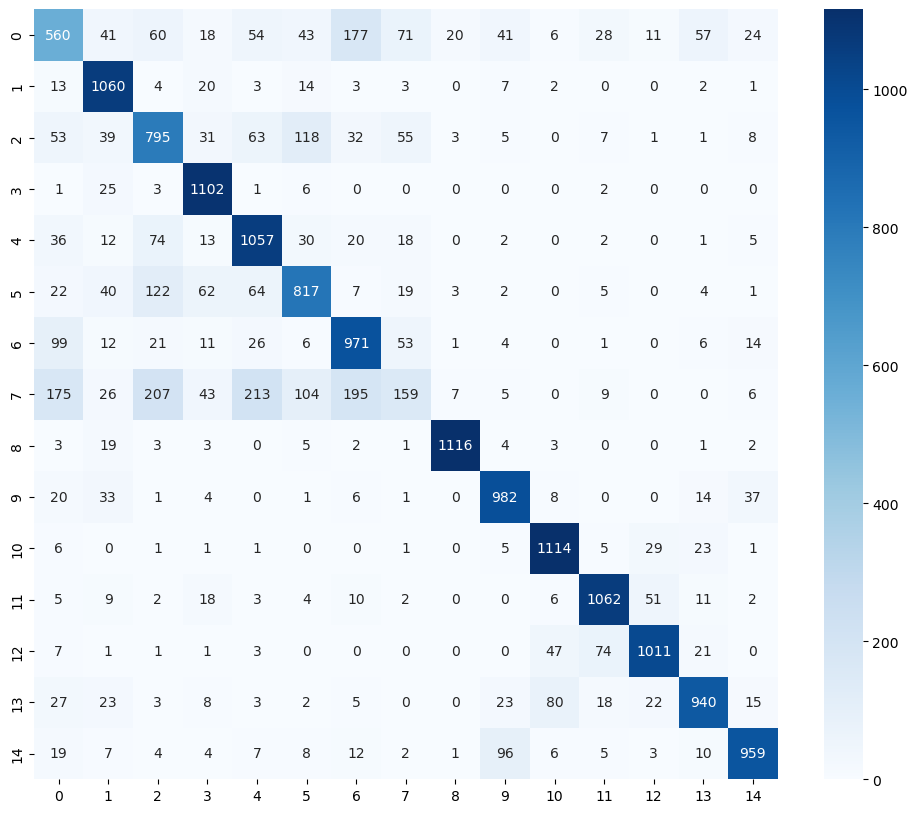
Таблиця 3.2. Результати роботи наївного Байєса

|  |  |  |
| --- | --- | --- |
| 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 |



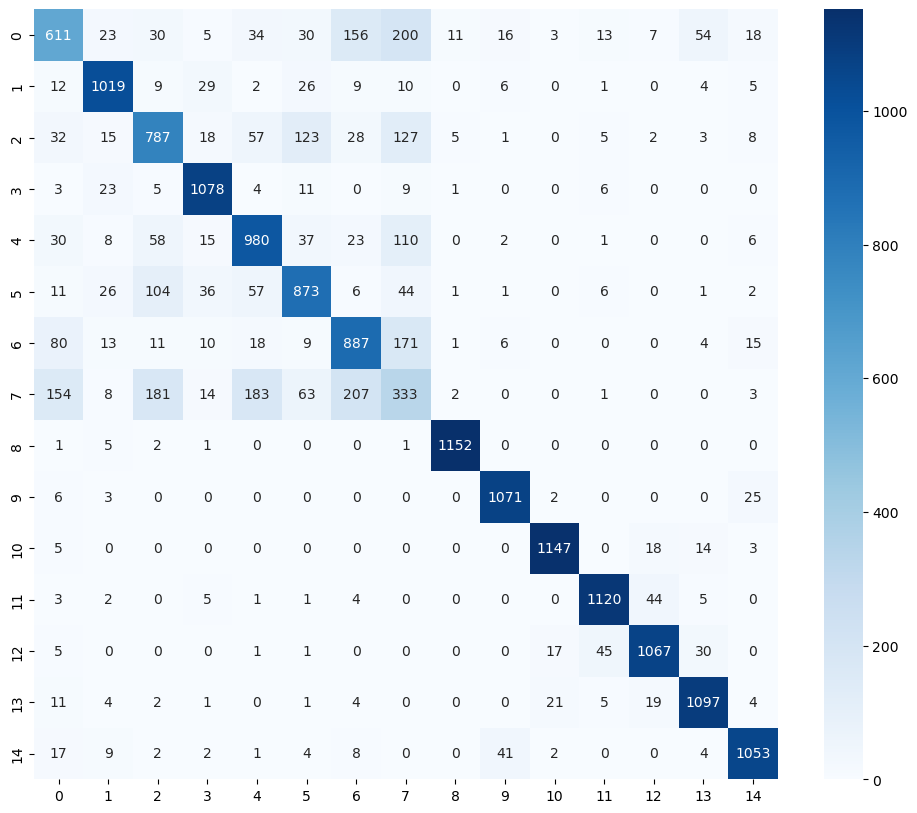
Таблиця 3.3. Результати роботи дерева рішень

|  |  |  |
| --- | --- | --- |
| 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 |



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 |



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 |

Таблиця 3.6. Результати роботи машини опорних векторів

|  |  |  |
| --- | --- | --- |
| Оптимальні параметри роботи | Показник ефективності | Значення показника ефективності оптимізованої SVM |
| Ядро  kernel= ‘rbf’,  Параметр регулярізації C=1000,  Коефіцієнт ядра Gamma = scale,  Критерій допуску зупинки  tol = 0.001 | Accuracy | 9772.76 |
| Precision(avg) | 986.6 |
| Recall(avg) | 1974460.75 |
| F1-score(avg) | 1405.16 |

## **Інтерпретація результатів роботи найкращої моделі**

Текст текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст.

# **ВИСНОВКИ**

Текст текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст.

Текст текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст.

Текст текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст.

Текст текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст.

# **СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ**

1. Liuzzi P. Predicting outcome of patients with prolonged disorders of consciousness using machine learning models based on medical complexity / P. Liuzzi, A. Magliacano, [et al.] // Scientific Reports. — 2022. — Vol. 12, No. 1. — P. 13471.
2. Wang Y. Cascade regression-based face frontalization for dynamic facial expression analysis / Y. Wang, X. Dong, G. Li, [et al.] // Cognitive Computation. — 2022. — Vol. 14, No. 5. — P. 1571–1584.
3. Larestani A. Compositional modeling of the oil formation volume factor of crude oil systems: application of intelligent models and equations of state / A. Larestani, A. Hemmati-Sarapardeh, Z. Samari, M. Ostadhassan // ACS Omega. — 2022. — Vol. 7, No. 28. — P. 24256–24273.

**ДОДАТОК А  
Програмна реалізація**

**ДОДАТОК Б  
Програмна реалізація**

