

Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Факультет _____ комп'ютерних наук _____
(повна назва)

Кафедра _____ програмної інженерії _____
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти _____ другий (магістерський) _____

Дослідження ансамблевих методів машинного навчання
для класифікації музичних жанрів у інформаційних системах
_____ (тема)

Виконав:
здобувач _____ 2 _____ року навчання
групи _____ ПЗМ-23-2 _____
_____ Андрій МОВЧАН _____
(Власне ім'я, ПРІЗВИЩЕ)

Спеціальність _____ 121 – Інженерія програмного
забезпечення. _____
(код і повна назва спеціальності)

Тип програми _____ освітньо-наукова _____

Керівник _____ проф. Сергій СМЕЛЯКОВ _____
(посада, Власне ім'я, ПРІЗВИЩЕ)

Допускається до захисту
Зав. кафедри _____

_____ Кирило СМЕЛЯКОВ _____
(підпис) (Власне ім'я, ПРІЗВИЩЕ)

2025 р.

Харківський національний університет радіоелектроніки

Факультет _____ комп'ютерних наук _____
 Кафедра _____ програмної інженерії _____
 Рівень вищої освіти _____ другий (магістерський) _____
 Спеціальність _____ 121 – Інженерія програмного забезпечення _____
 Тип програми _____ освітньо-наукова _____
 Освітня програма _____ Інженерія програмного забезпечення _____
 (шифр і назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

«___» _____ 2025 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

студенту _____ Мовчану Андрію Сергійовичу _____
 (Прізвище, ім'я, по батькові)

1. Тема роботи: «Дослідження ансамблевих методів машинного навчання для класифікації музичних жанрів у інформаційних системах»

Затверджена наказом по університету №290 Ст від 15.04.2025 _____

2. Термін подання студентом роботи до екзаменаційної комісії 19.06.2025

3. Вихідні дані до роботи: Python, ансамблі моделей, класифікація, мета-навчання, музичні треки, FMA, задача багатокритеріального вибору.

4. Перелік питань, які потрібно опрацювати в роботі:

Визначення оптимальних ансамблевих моделей, аналіз математичного апарату обраних моделей, оцінка впливу гіперпараметрів на продуктивність, практичне дослідження роботи найкращих моделей на обраному наборі даних, розробка комбінованого ансамблю.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Отримання завдання	19.05.2025	<i>виконано</i>
2	Аналіз предметної галузі і постановка задачі	20.05.2025	<i>виконано</i>
3	Огляд й аналіз літературних, наукових джерел	21.05.2025	<i>виконано</i>
4	Огляд математичного апарату моделей	24.05.2025	<i>виконано</i>
5	Планування експериментів	25.05.2025	<i>виконано</i>
6	Розробка програмної частини та проведення експериментів	28.05.2025	<i>виконано</i>
7	Підготовка до апробації результатів дослідження. Публікація матеріалів	30.05.2025	<i>виконано</i>
8	Підготовка пояснювальної записки	02.06.2025	<i>виконано</i>
9	Підготовка презентації та доповіді	04.06.2025	<i>виконано</i>
10	Перевірка на плагіат	08.06.2025	<i>виконано</i>
11	Нормоконтроль	09.06.2025	<i>виконано</i>
12	Рецензування	11.06.2025	<i>виконано</i>
13	Попередній захист	16.06.2025	<i>виконано</i>
14	Занесення диплома в електронний архів	16.06.2025	<i>виконано</i>
15	Допуск до захисту у зав. кафедри	16.06.2025	<i>виконано</i>

Дата видачі завдання 19.05.2025р.

Студент

_____ (підпис)

Андрій МОВЧАН

Керівник роботи

_____ (підпис)

проф. Сергій СМЕЛЯКОВ

(посада, Власне ім'я, прізвище)

РЕФЕРАТ / ABSTRACT

Пояснювальна записка містить: 66 с., 4 табл., 13 джерел, 6 рисунків.

АНСАМБЛІ МОДЕЛЕЙ, МАШИННЕ НАВЧАННЯ, МОДЕЛІ КЛАСИФІКАЦІЇ, МУЗИЧНИЙ ТРЕК, PYTHON, FMA.

Об'єкт дослідження – процес класифікації музичних треків за жанрами, що потребує створення ефективних методів для досягнення високої точності.

Мета роботи – підвищення точності та продуктивності класифікації музичних треків за жанрами шляхом обґрунтування, вибору та програмної реалізації найбільш ефективних ансамблевих моделей.

Методами розробки та аналізу є дослідження проблемної галузі, емпірична оцінка різних алгоритмів машинного навчання, вирішення багатокритеріальної задачі прийняття рішень для вибору оптимальних ансамблів.

У результаті дослідження визначено, обґрунтовано та програмно реалізовано найбільш перспективні ансамблеві методи класифікації жанрів музики, що можуть бути застосовані на наборі даних FMA із використанням засобів Python.

MODEL ENSEMBLE, MACHINE LEARNING, CLASSIFICATION MODELS, MUSIC TRACK, PYTHON, FMA.

Object of the research – the process of classifying music tracks by genre, which requires developing effective methods to achieve high accuracy.

The purpose of the work is to increase the accuracy and performance of music genre classification by substantiating, selecting, and programmatically implementing the most effective ensemble models.

The development and analysis methods include investigating the problem domain, empirically evaluating different machine learning algorithms, and solving a multi-criteria decision-making problem to choose optimal ensembles.

As a result of the study, the most promising ensemble methods for classifying music genres have been identified, substantiated and implemented in software. These methods can be applied to the FMA dataset using Python tools.

Завідувачу кафедри

ПІ

(скорочена назва кафедри)

проф. Кирилу СМЕЛЯКОВУ

(вчене звання, власне ім'я, прізвище)

ЗАЯВА

щодо самостійності виконання кваліфікаційної роботи та можливості її публікації
(та/або публікації анотації кваліфікаційної роботи) в електронному архіві
відкритого доступу EIAr KhNURE

Я, Мовчан Андрій Сергійович
(прізвище, ім'я, по батькові)

здобувач вищої освіти на другому (магістерському) рівні вищої освіти
академічної групи ІІЗМ-23-2
кафедри програмної інженерії
(повна назва кафедри)

заявляю: моя кваліфікаційна робота на тему

Дослідження ансамблевих методів машинного навчання для класифікації
музичних жанрів у інформаційних системах,
(назва роботи)

що буде представлена в екзаменаційну комісію для публічного захисту, виконана
самостійно, в ній не містяться елементи плагіату і вона може бути опублікована в
репозиторії "EIArKhNURE". Погоджуюся з авторським договором, відповідно до
Положення про репозиторій ХНУРЕ "EIArKhNURE". Всі запозичення з
друкованих та електронних джерел мають відповідні посилання.

Я ознайомлений з вимогами академічної доброчесності, згідно з якими
виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до
захисту та застосування дисциплінарних заходів.

Дата

Підпис

ЗМІСТ

Вступ.....	9
1 Аналіз предметної галузі і постановка задачі	11
1.1 Аналіз предметної галузі.....	11
1.2 Постановка задачі багатокритеріального вибору	12
2 Огляд й аналіз літературних, наукових джерел	19
2.1 Огляд основних джерел.....	19
2.2 Аналіз літератури.....	21
2.3 Оцінка актуальності та новизни	22
2.4 Висновки з огляду літератури.....	22
3 Постановка задачі.....	24
4 Теоретичне дослідження	27
4.1 Вступ до ансамблевих методів класифікації музичних треків.....	27
4.2 Bootstrapping Ensemble (Bagging).....	27
4.2.1 Основи методу	27
4.2.2 Математичний апарат Bagging	28
4.3 AdaBoost (Adaptive Boosting).....	29
4.3.1 Основи методу	29
4.3.2 Математичний апарат AdaBoost.....	29
4.4 Random Forests.....	31
4.4.1 Основи методу	31
4.4.2 Математичний апарат Random Forests.....	31
4.4.3 Вибір критерію розщеплення вузлів	32
4.5 Аналіз проблеми класифікації за багатьма жанрами	33
4.6 Теоретичні аспекти налаштування і підвищення ефективності моделей	34
4.7 Теоретичні аспекти стійкості до перенавчання та адаптації до нових даних	35
4.8 Підготовка до експериментальних досліджень	35
4.9 Побудова комбінованого ансамблю.....	37
5 Програмна реалізація	38
5.1 Загальна архітектура програмного забезпечення	38

	8
5.2 Підготовка та відбір ознак	39
5.3 Використання itertools для первинного навчання.....	40
5.4 Оптимізація гіперпараметрів за допомогою Optuna.....	41
5.5 Реалізація ансамблевого навчання	42
5.6 Результати експериментів	44
Висновки	48
Перелік джерел посилання	50
Перелік джерел посилання за науковими напрямками керівника та науковців кафедри програмної інженерії	52
Додаток А Слайди презентації.....	53
Додаток Б Апробація результатів роботи. Тези.....	62
Додаток В Звіт результатів перевірки на унікальність тексту в базі ХНУРЕ.....	65
Додаток Г Експертний висновок результатів перевірки кваліфікаційної роботи на відповідність вимогам ДСТУ 3008: 2015.....	66

ВСТУП

У сучасному світі цифрових технологій музика стала невід'ємною частиною повсякденного життя мільйонів людей. Зі зростанням обсягу цифрових музичних бібліотек виникає необхідність ефективної систематизації та пошуку музичних творів. Класифікація музичних треків за жанрами стала критично важливою для поліпшення рекомендаційних систем, персоналізації користувацького досвіду та полегшення навігації в масштабних музичних бібліотеках.

Проте класифікація музичних треків залишається складною задачею через високу варіативність музичних стилів, змішування жанрів та суб'єктивність сприйняття музики. Традиційні методи машинного навчання часто стикаються з проблемами низької точності та перенавчання, особливо при роботі з великими та неоднорідними наборами даних.

У зв'язку з цим виникає потреба у дослідженні нових підходів до класифікації, зокрема використання мета-навчання та ансамблевих методів. Мета-навчання дозволяє моделі адаптуватися до нових завдань на основі попереднього досвіду, а ансамблеві методи підвищують загальну продуктивність шляхом комбінування декількох моделей. Ці підходи можуть значно покращити точність та стабільність класифікації музичних жанрів.

Варто зазначити, що вибір та коректне застосування дослідницької методології є одним із ключових аспектів проведення якісних наукових досліджень у галузі інформаційних систем, оскільки вона визначає інструментарій для досягнення поставлених цілей і адекватного аналізу отриманих результатів [1].

У даному дослідженні планується застосувати мета-навчання та побудову ансамблів моделей для класифікації музичних треків за жанрами. Для цього буде використано набір даних FMA (Free Music Archive), зокрема його підмножину `fma_small`, яка містить різноманітні музичні треки з анотаціями жанрів. Цей набір даних є відкритим та широко використовується для проведення експериментів.

Актуальність теми полягає у зростаючій потребі в ефективних методах класифікації музики для покращення користувацького досвіду та оптимізації

роботи музичних сервісів. Використання сучасних методів машинного навчання, зокрема мета-навчання та ансамблів, відкриває нові можливості для вирішення цієї задачі.

Метою роботи є розробка моделей класифікації музичних треків за жанрами, які базуються на мета-навчанні та ансамблевих методах, з ціллю досягнення високої точності та продуктивності. У ході дослідження планується описати математичний апарат, що лежить в основі обраних моделей, та провести їх теоретичний аналіз.

Об'єктом дослідження є класифікація музичних треків за жанрами за допомогою методів машинного навчання. Предметом дослідження є моделі класифікації музичних жанрів, засновані на методах машинного навчання.

Для досягнення мети поставлено такі завдання:

- проаналізувати теоретичні аспекти роботи методів обраних в результаті вирішення задачі багатокритеріального вибору;
- визначити оптимальні стратегії налаштування гіперпараметрів для обраних моделей;
- дослідити і обрати спосіб вирішення проблеми класифікації серед багатьох жанрів одночасно (one-vs-one, one-vs-rest);
- сформулювати загальний план виконання експериментів з моделями;
- програмно реалізувати обрані моделі.

Структура роботи включає аналіз предметної галузі та постановку задачі, огляд літературних та наукових джерел, детальну постановку задачі, теоретичне дослідження обраних моделей, програмну реалізацію та висновки. Очікується, що результати дослідження сприятимуть подальшому розвитку методів класифікації музичних треків та їх практичному застосуванню в індустрії.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ І ПОСТАНОВКА ЗАДАЧІ

1.1 Аналіз предметної галузі

У сучасних умовах цифрової епохи музика стала невід'ємною частиною нашого повсякденного життя, доступною практично без обмежень. Зі зростанням обсягів музичних архівів, потокових сервісів та онлайн-бібліотек виникає потреба в автоматизованих системах, здатних ефективно класифікувати та структурувати музичні дані. Одним із важливих аспектів такої систематизації є визначення жанру музичного треку, що має вирішальне значення для пошуку, рекомендаційних систем, створення каталогів, аналізу музичних трендів та формування музичних добірок.

Існує велика кількість музичних жанрів, які можуть бути досить близькими за стилістичними рисами, інструментальним складом, темпами чи мелодичними патернами. Подібність між жанрами часто ускладнює завдання класифікації, оскільки алгоритмам потрібно розрізняти тонкі особливості в акустичному сигналі. До того ж, реальні музичні треки можуть містити елементи кількох жанрів одночасно, утворюючи гібридні стилі та піджанри. Це вимагає побудови моделей з високою здатністю до узагальнення та адаптації.

Ключовим фактором у розв'язанні цієї проблеми є вибір відповідних ознак та методів обробки аудіоданих. Музичний сигнал можна перетворювати в різні репрезентації: частотні (спектрограми, мел-спектрограми, коефіцієнти MFCC), часові чи навіть комбіновані. Вибір таких ознак впливає на якість класифікації, адже різні жанри можуть відрізнятися за темпами, ритмічними малюнками, гармонічними зв'язками чи інструментальним складом.

Окрім класифікації музики за жанрами, область обробки музичної інформаційної охоплює ще й завдання рекомендацій, сегментації, прогнозування популярності треків та аналізу музичних трендів. Усі ці напрямки вимагають ефективних алгоритмів, здатних працювати з великими даними, адаптуватися до появи нових стилів та масштабуватися для онлайн-сервісів. Завдання класифікації жанрів є важливим елементом цієї екосистеми, оскільки саме жанр є однією з ключових категорій, за якими можна структурувати музичні колекції.

Таким чином, предметна галузь класифікації музичних треків за жанрами – це міждисциплінарне поле, що поєднує методи обробки сигналів, машинне та глибинне навчання, а також музикознавство. Сучасні підходи фокусуються на використанні глибинних архітектур, застосуванні ансамблевих методів, використанні мета-навчання та оптимізації гіперпараметрів. Все це спрямоване на досягнення максимальної точності класифікації, стійкості до шуму та здатності обробляти різноманітні стилі й напрями музики.

1.2 Постановка задачі багатокритеріального вибору

Для вибору найкращих моделей класифікації необхідно аналітично оцінити їх за різними критеріями, які не потребують запуску моделей і безпосереднього проведення експериментів. Отже, постає задача багатокритеріального вибору найкращих моделей із запропонованого набору альтернатив, що потребує формалізації та застосування відповідного методичного апарату для прийняття рішень у складних умовах [2].

При розгляді проблеми класифікації музичних треків за жанрами, існуючі алгоритми машинного навчання можна поділити на дві основні категорії:

- прості моделі;
- ансамблі моделей.

Прості моделі, такі як Support Vector Classifier (SVC), часто застосовуються для задач з лінійно роздільними класами та мають меншу складність реалізації. Ансамблеві методи, такі як Random Forests, Gradient Boosting, Bootstrapping Ensemble та AdaBoost, мають певні особливості які є важливими під час тренування, що дозволяє досягти покращення точності та стабільності, але слід зауважити, що ці ансамблі можуть вимагати більше ресурсів та часу для навчання.

Сформуємо наступний набір альтернатив для багатокритеріальної задачі прийняття рішень:

a) Support Vector Classifier (SVC):

- 1) проста модель класифікації, що використовує гіперплощини для

поділу даних;

2) ефективна для задач з високорозмірними даними;

б) Random Forests:

1) ансамблевий метод, що об'єднує множину дерев рішень;

2) стійкий до викидів та працює з різними типами даних;

в) Gradient Boosting:

1) ансамблевий метод, який послідовно додає моделі для мінімізації похибки;

2) забезпечує досить високу точність, але може бути складним у налаштуванні;

г) Bootstrapping Ensemble (Bagging):

1) метод, що використовує ресемплінг з заміщенням для створення ансамблю моделей;

2) підвищує стабільність окремих моделей та зменшує дисперсію оцінок;

д) AdaBoost (Adaptive Boosting):

1) ансамблевий метод, що комбінує слабкі класифікатори;

2) ефективний для підвищення продуктивності моделей, що слабо навчаються.

З метою порівняння обраних моделей машинного навчання, визначимо декілька основних критеріїв, на яких буде засновуватися наш вибір. Набір критеріїв виглядає наступним чином:

- інтерпретованість моделі;
- використання пам'яті;
- асимптотична обчислювальна складність;
- складність підбору параметрів;
- можливість паралелізації.

Обґрунтуємо вибір саме таких критеріїв, а також визначимо й опишемо шкали для них.

Інтерпретованість моделі буде показувати, наскільки легко нам зрозуміти, як саме модель приймає рішення й доходить до результатів класифікації.

Інтерпретованість визначається наявністю бібліотек (SHAP, ELI5) для опису процесу отримання результатів класифікації та наявністю взаємодії між моделями (що значно ускладнює інтерпретацію). Вища інтерпретованість є кращою, тому моделі з більшою інтерпретованістю отримують вищі оцінки:

- висока інтерпретованість – 3 бали;
- середня інтерпретованість – 2 бали;
- низька інтерпретованість – 0 балів.

Оцінка критерію використання пам'яті базується на обсязі оперативної пам'яті, необхідного для роботи алгоритму. Цей критерій є важливим при роботі з великими наборами даних та обмежених ресурсах. Менше використання пам'яті є кращим, тому моделі з меншим використанням пам'яті отримують вищі оцінки:

- низьке використання пам'яті ($O(n)$ або менше) – 3 бали;
- середнє використання пам'яті ($O(n \log n)$) – 2 бали;
- високе використання пам'яті ($O(n^2)$ або більше) – 0 балів.

Асимптотична обчислювальна складність надає теоретичну оцінку часу виконання алгоритму залежно від розміру даних. Цей критерій також є важливим, адже вхідний розмір даних напряму впливає на швидкість навчання моделі. Нижча складність є кращою, тому моделі з меншою складністю отримують вищі оцінки:

- низька складність ($O(n \log n)$ або менше) – 3 бали;
- середня складність ($O(n^2)$) – 2 бали;
- висока складність ($O(n^3)$ або більше) – 0 балів.

Що стосується складності підбору параметрів, тут оцінюється наскільки важко налаштувати параметри моделі для досягнення оптимальної продуктивності. Відомими для цього методами є Grid Search та Баясова оптимізація. Це може мати значний вплив на час навчання моделі. Менша складність є кращою, тому моделі з простішим налаштуванням отримують вищі оцінки:

- проста складність налаштування (до 30 комбінацій) – 3 бали;
- середня складність налаштування (до 100 комбінацій) – 2 бали;

– висока складність налаштування (понад 100 комбінацій) – 0 балів.

Розглянемо також останній критерій. Можливість паралелізації визначає здатність алгоритму бути розпаралеленим для прискорення обчислень. Це значно впливає на час навчання моделі, особливо на великих наборах даних. Вища можливість паралелізації є кращою, тому моделі, які можуть ефективно використовувати паралельні обчислення, отримують вищі оцінки:

- висока можливість паралелізації – 3 бали;
- обмежена можливість паралелізації – 2 бали;
- відсутність можливості паралелізації – 0 балів.

Приведемо значення для кожного із критеріїв наших альтернатив (див. таб. 1.1).

Таблиця 1.1 – Значення критеріїв альтернатив (таблиця виконана самостійно)

	Можливість інтерпретації моделі	Використання пам'яті	Асимптотична обчислювальна складність	Складність підбору параметрів	Можливість паралелізації
SVC	середня	$O(n^2)$	$O(n^3)$	понад 100	відсутня
Random Forests	середня	$O(m*n)$	$O(m*n*\log n)$	понад 100	висока
Gradient Boosting	середня	$O(m*n)$	$O(m*n*\log n)$	понад 100	відсутня
Bootstrapping Ensemble	середня	$O(m*n)$	$O(m*n*\log n)$	~50	висока
Adaptive Boosting	середня	$O(m*n)$	$O(m*n*\log n)$	~27	відсутня

Таблицю заповнено базуючись на відомих фактах про розглядувані алгоритми. Також враховувалась інформація щодо існуючих бібліотек для інтерпретації результатів моделей.

Тепер, коли ми маємо значення відповідних критеріїв, переведемо їх у кількісний формат, щоб застосувати відповідний метод теорії корисності (див.

таб. 1.2)

Таблиця 1.2 – Числові значення критеріїв альтернатив

	Можливість інтерпретації моделі	Використання пам'яті	Асимптотична обчислювальна складність	Складність підбору параметрів	Можливість паралелізації
SVC	2	0	0	0	0
Random Forests	2	2	2	0	3
Gradient Boosting	2	2	2	0	0
Bootstrapping Ensemble	2	2	2	2	3
Adaptive Boosting	2	2	2	3	0

Скористаємося принципом Парето. Завдяки ньому, ми позбавилися двох альтернатив, які є гіршими за інші, це SVC та Gradient Boosting.

Сформована множина Парето наведена нижче (див. таб. 1.3).

Таблиця 1.3 – Множина Парето

	Можливість інтерпретації моделі	Використання пам'яті	Асимптотична обчислювальна складність	Складність підбору параметрів	Можливість паралелізації
Random Forests	2	2	2	0	3
Bootstrapping Ensemble	2	2	2	2	3
Adaptive Boosting	2	2	2	3	0

Критерії не є рівнозначними для поставленої задачі вибору, тому для її вирішення було обрано лінійну адитивну згортку з ваговими коефіцієнтами.

Застосуємо нормування для Парето-оптимальних альтернатив, які ми отримали (див. таб. 1.4).

Таблиця 1.4 – Таблиця нормованих значень множини Парето

	Можливість інтерпретації моделі	Використання пам'яті	Асимптотична обчислювальна складність	Складність підбору параметрів	Можливість паралелізації
Random Forests	0.67	0.67	0.67	0.0	1.0
Bootstrapping Ensemble	0.67	0.67	0.67	0.67	1.0
Adaptive Boosting	0.67	0.67	0.67	1.0	0.0

Таблицю нормованих значень було розраховано при $x_{\min} = 0$, $x_{\max} = 3$.

При визначенні вагових коефіцієнтів було враховано специфіку задачі та практичні аспекти застосування моделей. Основна увага приділялася критеріям, які безпосередньо впливають на ефективність розробки та використання моделі в реальних умовах.

Складність підбору параметрів безпосередньо впливає на час і ресурси, необхідні для налаштування моделі до оптимальної продуктивності, адже на практиці ми повинні будемо навчити і протестувати модель стільки разів, скільки маємо різних комбінацій параметрів. Крім того, простота підбору параметрів знижує ризик перенавчання та покращує узагальнюючу здатність моделі. На другому місці буде асимптотична обчислювальна складність, адже вона визначає, скільки часу знадобиться для навчання та прогнозування з використанням цієї моделі. Третім по важливості обрано використання пам'яті, тому що обсяг пам'яті, необхідний для роботи моделі, впливає на можливість її застосування в реальних умовах, особливо при обмежених ресурсах. Четверте і п'яте місце займають можливість паралелізації та інтерпретованість,

Вищі коефіцієнти були присвоєні критеріям, що мають найбільший вплив

на ресурси та продуктивність, тоді як менш значні критерії отримали нижчі ваги:

- складність підбору параметрів: 0.3;
- асимптотична обчислювальна складність: 0.25;
- використання пам'яті: 0.2;
- можливість паралелізації: 0.15;
- інтерпретованість: 0.1.

Тепер можемо приступити до розрахунку корисності Парето-оптимальних альтернатив відповідно до формули 1.1.

$$U_i = \sum_{j=1}^m w_j x'_{ij} \quad (1.1)$$

де U_i – корисність i -тої альтернативи,

w_j – ваговий коефіцієнт для j -го критерію, що відображає його відносну важливість (сума всіх вагових коефіцієнтів дорівнює 1),

x'_{ij} – нормоване значення i -ї альтернативи за j -м критерієм, приведене до узгодженого масштабу ($[0;1]$ та до принципу “за максимумом”).

В результаті розрахунків наші альтернативи отримують наступні значення:

- Random Forests: 0,5185;
- Bootstrapping Ensemble: 0,7195;
- AdaBoost: 0,6685.

На основі цих розрахунків було прийнято рішення, що ансамблевий метод Bootstrapping Ensemble є найкращим кандидатом для класифікації музичних треків, але в межах даного дослідження, для більш глибокого розуміння і порівняння, доречним буде розглянути математичне представлення усіх трьох найкращих моделей.

2 ОГЛЯД Й АНАЛІЗ ЛІТЕРАТУРНИХ, НАУКОВИХ ДЖЕРЕЛ

2.1 Огляд основних джерел

Класифікація музичних треків за жанрами є однією з ключових задач у галузі обробки музичної інформації. Постійне зростання обсягів доступної музики, різноманітність жанрових стилів та необхідність ефективного способу каталогізації стимулюють науковців до пошуку нових підходів, архітектур та алгоритмів. У цьому розділі наведено огляд низки сучасних наукових робіт, що пропонують різноманітні рішення: від відкритого розпізнавання (OSR) до застосування глибинних згорткових та рекурентних мереж, а також мультимодальних та ансамблевих методів. Нижче наведено систематизований огляд літературних та наукових джерел, які було обрано за їх авторитетність, актуальність та цільовий характер.

У першій розглянутій роботі [3] автори пропонують метод Open Set Recognition (OSR) для класифікації музичних жанрів. Цей підхід передбачає, що поряд із відомими жанрами модель здатна виявляти нові чи невідомі (раніше не розглядувані) жанри. OSR має важливе практичне значення, оскільки в реальних умовах музична сцена швидко розвивається, і постійно з'являються нові стилі й напрямки. Дослідження доводить, що інтеграція OSR робить систему більш гнучкою та актуальною для великого спектра жанрів, включно з гібридними чи експериментальними стилями музики.

У роботі [4] автори зосередилися на згорткових нейронних мережах, які спеціально адаптовані до музичних сигналів. Такий підхід, названий «музично мотивованою» CNN, враховує ритмічні та гармонійні особливості звуку на етапі проектування архітектури мережі. Головна ідея полягає в тому, щоб фільтри та шари мережі відповідали логіці музичних патернів, а не були «універсальними», як у класичних CNN для зображень. Результати роботи демонструють, що така спеціалізація помітно підвищує точність визначення жанрів.

У дослідженні [5] запропоновано Convolutional Recurrent Neural Networks (CRNN), які поєднують переваги згорткових мереж (CNN) для вилучення локальних ознак із аудіосигналів та рекурентних мереж (RNN) для аналізу

глобальних часових залежностей. Такий гібридний підхід дає змогу краще розуміти не лише короткострокові спектральні фрагменти, а й цілісну структуру треку, враховуючи розвиток мелодики, гармонії та ритму в часі. Це особливо важливо, коли жанр визначається великою кількістю ознак, розподілених по всій тривалості музичного треку.

У роботі [6] автор описує порівняння між CNN та XGBoost, використовуючи Mel-спектрограми та коефіцієнти MFCC як основні ознаки для класифікації. Ключовий висновок полягає в тому, що за деяких умов алгоритм XGBoost може перевершувати глибинні мережі. Це підкреслює, що класичні підходи (Boosting) досі залишаються конкурентоспроможними, а вибір методу залежить від специфіки даних і завдання. Такий висновок змушує дослідників приділяти увагу належному налаштуванню гіперпараметрів і вибору релевантних ознак, оскільки саме в цьому полягає запорука успіху XGBoost.

У публікації [7] запропоновано застосувати гібридну модель, що поєднує ResNet та Bi-GRU для роботи зі спектрограмами. ResNet забезпечує глибоке вилучення ієрархічних ознак, а Bi-GRU – ефективне моделювання часових залежностей у двох напрямках. Такий підхід дозволяє захоплювати як локальні спектральні характеристики, так і глобальні динамічні патерни, що особливо корисно при відмінностях між жанрами, які можуть проявлятися як у коротких фрагментах, так і в загальній структурі треку.

Нарешті, у роботі [8] розглянуто мультимодальний підхід. Автори інтегрують аудіо- та текстові (можливо метадані, назви треків, виконавців) ознаки, використовуючи механізми крос-модальної уваги. Крім того, вони враховують кореляції між жанрами, що допомагає точніше класифікувати композиції, які поєднують елементи декількох стилів. Такий мультимодальний аналіз стає ще одним кроком до більш складних та реалістичних систем, які можуть ефективно працювати з різними джерелами інформації.

Таким чином, наукова література свідчить про значний прогрес у методах класифікації музичних жанрів, пропонуючи широкий спектр архітектур, алгоритмів та стратегій, які можна застосувати для досягнення високої точності,

стійкості до нових жанрів і максимальної узагальнюючої здатності. Подані джерела обрано за критеріями авторитетності (відомі конференції ISMIR, ICASSP), наявності конкретних експериментів із даними та акцентом на сучасних підходах (глибинне навчання, Boosting, CRNN, мультимодальні методи). Усі ці праці мають пряме відношення до завдань класифікації музичних треків за жанрами, відображають актуальний стан галузі та містять результати експериментів, які підтверджують ефективність запропонованих рішень.

2.2 Аналіз літератури

Розглянуті роботи зосереджені на розробці теоретичних і практичних основ для розв'язання задач класифікації жанрів. Метод OSR, представлений у [3], є важливим внеском у теорію обробки музичної інформації, оскільки дозволяє системам розпізнавати жанри, які не входять у початковий набір навчальних даних. Це суттєво розширює межі застосування моделей у реальних умовах.

Музично мотивовані CNN, розглянуті в [4], підкреслюють важливість адаптації архітектури під специфіку даних. Це є теоретичною базою для одного з підходів створення моделей, які враховують гармонічні та ритмічні особливості музики.

Модель CRNN із роботи [5] демонструє практичну реалізацію концепції об'єднання локального та глобального аналізу, що сприяє кращому розумінню складних музичних сигналів.

Методи, використані у розглянутих роботах, відображають різноманіття підходів до обробки аудіо-даних:

- OSR у [3] базується на розширенні можливостей традиційних CNN через додавання механізмів розпізнавання нових жанрів;
- у [4] використано спеціалізовані фільтри CNN, що відповідають музичним патернам;
- CRNN у [5] об'єднує переваги CNN для вилучення локальних ознак і RNN для врахування часових залежностей;
- у [6] порівняння між CNN і XGBoost показує, що класичні Boosting-

алгоритми можуть бути конкурентоспроможними завдяки гнучкості та можливості налаштування;

- у [7] гібридна модель ResNet+Bi-GRU демонструє переваги багаторівневого аналізу ознак, що важливо для складних ієрархічних даних;
- мультимодальний підхід у [8] інтегрує різні типи даних для досягнення більш точної класифікації.

Отже, дослідження [3-8] підкреслюють переваги глибоких архітектур, налаштованих під специфіку задач класифікації музики. Результати демонструють високу точність моделей, що враховують специфіку музичних сигналів.

2.3 Оцінка актуальності та новизни

Актуальність розглянутих робіт зумовлена зростаючою потребою в автоматизації процесів класифікації музики. Роботи, такі як [3] і [8], вирішують задачі, пов'язані з динамічним розширенням жанрової палітри та адаптацією моделей до нових стилів. Новизна підходів у [4] і [5] полягає в інтеграції музичних знань у дизайн архітектур CNN і CRNN, що є відходом від універсальних рішень.

Робота [6] демонструє, що класичні алгоритми, такі як XGBoost, залишаються актуальними завдяки своїй простоті та ефективності. Це важливо для задач, де обчислювальні ресурси обмежені або необхідна висока швидкість навчання.

У дослідженні [7] представлена гібридна модель ResNetBi-GRU, яка є новаторською завдяки інтеграції різних типів мереж для досягнення високої продуктивності. Такий підхід є перспективним для задач з великою кількістю різнорідних даних.

2.4 Висновки з огляду літератури

На основі аналізу літературних джерел можна зробити висновок, що існує широкий спектр підходів до класифікації музичних жанрів, кожен із яких має свої

переваги та обмеження. Загалом, розглянуті дослідження підтверджують актуальність використання ансамблевих методів, глибоких нейронних мереж та мультимодальних підходів.

Більшість робіт свідчать про доцільність використання комбінацій різних типів ознак, оскільки жодна окрема група дескрипторів не є універсальною для всіх жанрів. Також у літературі активно обговорюється проблема перенавчання моделей на обмежених або незбалансованих наборах даних, що особливо критично для задачі музичної класифікації через високу варіативність жанрової структури.

Значну увагу приділено також інтеграції зовнішніх джерел даних, таких як ліричний текст, метадані або інформація про виконавця, що дозволяє формувати мультимодальні архітектури для підвищення точності класифікації.

Водночас залишаються відкритими питання оптимізації гіперпараметрів, підвищення стійкості моделей до шуму та розширення їхньої здатності до узагальнення. Це підкреслює необхідність подальших досліджень у цій галузі.

Це узагальнення дозволяє сформуванню основи для побудови власної системи класифікації, яка враховує як попередні досягнення в галузі, так і сучасні виклики. Таким чином, обґрунтоване поєднання кількох підходів може забезпечити більш стійкі результати в умовах реального використання.

3 ПОСТАНОВКА ЗАДАЧІ

У контексті класифікації музичних треків за жанрами існує велика кількість алгоритмів машинного навчання, які можуть бути застосовані для вирішення цієї задачі. Після аналізу предметної галузі та багатокритеріального вибору (розділ 1) було визначено, що найбільш перспективними методами є ансамблеві алгоритми: Bootstrapping Ensemble (Bagging), AdaBoost та Random Forests. Ці алгоритми обрано на основі їх оптимальних характеристик за критеріями продуктивності, точності та ефективності, зокрема, за принципом Парето. Сутність підходу полягає в тому, що модель класифікації створюється не як єдиний предиктор, а як ансамбль декількох незалежних чи послідовних підмоделей, що дозволяє підвищувати стійкість та узагальнюючу здатність системи до різноманітних видів музичних патернів.

Метою даного дослідження є детальне вивчення зазначених ансамблевих методів класифікації музичних треків за жанрами, розуміння їх теоретичних основ, а також практичних аспектів їх застосування. Підсумкові результати повинні показати, наскільки ефективно обрані ансамблі можуть узагальнювати та класифікувати музичні композиції з різними ритмічними, тембральними та тональними ознаками.

Для досягнення цієї мети необхідно виконати кілька основних завдань. По-перше, детально проаналізувати структуру та механізм роботи кожного обраного ансамблю, зокрема:

- процес створення ансамблю моделей bootstrapping ensemble (Bagging) шляхом ресемплінгу з заміщенням, тобто розглянути специфіку бутстреп-вибірок, механізми комбінування результатів базових моделей та особливості зменшення дисперсії за допомогою багатьох незалежно навчених класифікаторів;
- механізм адаптивного підсилення AdaBoost, зокрема вплив ваг зразків на процес навчання та роль послідовного додавання слабких

класифікаторів, адже важливо з'ясувати, як саме змінюються ваги неправильно класифікованих зразків та як це підвищує точність;

- методику побудови ансамблю випадкових дерев рішень Random Forests, проаналізувати випадковий вибір ознак при розщепленні вузлів та визначити, яким чином кількість дерев може впливати на загальну продуктивність, адже необхідно розуміти, чому випадковість знижує кореляцію між деревами та зменшує ризик перенавчання.

Також треба розробити підходи до ефективної реалізації та налаштування обраних алгоритмів:

- визначити оптимальні стратегії налаштування гіперпараметрів для кожного методу з урахуванням специфіки даних обраного датасету і розробити план для пошуку найкращих значень гіперпараметрів, використовуючи методи Бассової оптимізації;
- розглянути можливості покращення продуктивності моделей за рахунок попередньої обробки даних, вибору релевантних ознак та використання методів зменшення розмірності.

Відповідно, для практичної реалізації моделей необхідно провести теоретичне дослідження математичного апарату кожного методу наступним чином:

- сформулювати основні концепції та алгоритмічні кроки в моделі, підготувати основу для опису статистичних методів та формул, що лежать в основі кожного ансамблю;
- визначити ключові аспекти, на які слід звернути увагу при теоретичному аналізі, як, наприклад, стійкість до перенавчання та інші теоретичні властивості.

Після проведення такого аналізу можна приступити до планування експериментального дослідження:

- розробити дизайн експериментів для оцінки ефективності обраних методів на наборі даних FMA (fma_small), визначити метрики оцінки

якості класифікації, які будуть використовуватися, такі як точність, повнота, F-міра та інші релевантні показники;

- запланувати проведення порівняльного аналізу результатів, щоб визначити сильні та слабкі сторони кожного методу в контексті розглядуваного датасету.

Таким чином, задача даного дослідження полягає у поглибленні розуміння обраних ансамблевих методів, розробці стратегій їх ефективного застосування та практичної реалізації в результаті теоретичного аналізу математичного апарату, що лежить в їх основі.

4 ТЕОРЕТИЧНЕ ДОСЛІДЖЕННЯ

4.1 Вступ до ансамблевих методів класифікації музичних треків

Ансамблеві методи машинного навчання базуються на ідеї комбінування декількох моделей з метою підвищення загальної продуктивності, точності та стійкості моделей до перенавчання. У контексті класифікації музичних треків за жанрами вони набувають особливого значення, адже музичні дані містять складні часово-спектральні структури, різноманітні тембральні та ритмічні характеристики, які важко повністю охопити однією моделлю. Застосування мета-навчання, ансамблювання моделей та оптимізація гіперпараметрів дозволять створювати більш гнучкі та ефективні алгоритми, здатні пристосуватися до різноманітних типів жанрових патернів.

На основі попереднього аналізу (розділ 1) та вирішення задачі багатокритеріального вибору було визначено, що найбільш перспективними методами є ансамблеві підходи Bootstrapping Ensemble (Bagging), AdaBoost та Random Forests. Вибір обумовлено їх високою точністю, ефективністю, здатністю до узагальнення та можливістю адаптуватися до складних умов навчання.

У цьому розділі буде детально розглянуто теоретичні засади кожного з цих методів, наведено математичний апарат, розібрано механізми роботи алгоритмів та представлено відповідні формули з детальним описом змінних. Крім того, буде розглянуто підходи до налаштування гіперпараметрів, попередньої обробки даних та оптимізації ознак, а також підкреслено теоретичні аспекти стійкості моделей.

4.2 Bootstrapping Ensemble (Bagging)

4.2.1 Основи методу

Bagging [9] – це один із перших ансамблевих методів, що істотно вплинув на розвиток машинного навчання. Ідея полягає у тому, щоб навчити велику кількість базових класифікаторів на різних випадкових підвибірках вихідного набору даних, а потім комбінувати їхні прогнози шляхом голосування або усереднення. Застосування бутстреп-ресемплінгу означає, що кожна підвибірка вибирається з поверненням, і деякі зразки можуть повторюватися, тоді як інші

можуть не потрапити в певну вибірку взагалі. Це підвищує різноманітність між базовими моделями і зменшує їх кореляцію, що призводить до зниження дисперсії ансамблю.

4.2.2 Математичний апарат Bagging

Нехай маємо початковий набір даних (формула 4.1):

$$D = \{(x_i y_i)\}_{i=1}^N \quad (4.1)$$

де $x_i \in R^M$ – вектор ознак i -го зразка,

$y_i \in \{1, \dots, K\}$ – мітка класу,

N – розмір набору даних,

M – кількість ознак,

K – кількість жанрів.

Створюємо B бутстреп-вибірок (формула 4.2):

$$D^{(b)} = \{(x_i^{(b)} y_i^{(b)})\}_{i=1}^N, b = 1, 2, \dots, B \quad (4.2)$$

де кожна бутстреп вибірка $D^{(b)}$ генерується шляхом випадкового вибору N зразків із D з поверненням.

Для кожної бутстреп-вибірки $D^{(b)}$ навчається базовий класифікатор (формула 4.3):

$$h^{(b)}(x) : R^M \rightarrow \{1, \dots, K\} \quad (4.3)$$

де $h^{(b)}(x)$ – базовий класифікатор, навчений на бутстреп вибірці $D^{(b)}$.

Тоді кінцевий прогноз ансамблю для нового зразка x розраховується наступним чином (формула 4.4):

$$H(x) = \text{mode}\{h^{(1)}(x), h^{(2)}(x), \dots, h^{(B)}(x)\} \quad (4.4)$$

де $H(x)$ – кінцевий ансамблевий прогноз для зразка x ,

B – кількість базових класифікаторів, створюваних у рамках Bagging,

mode – функція, що повертає клас, який найбільш часто зустрічається серед усіх прогнозів базових моделей.

4.3 AdaBoost (Adaptive Boosting)

4.3.1 Основи методу

AdaBoost [10] – адаптивний метод підсилення, метою якого є поетапне покращення точності шляхом фокусування на “важких” для класифікації зразках. На початку всім зразкам надаються рівні ваги. На кожній ітерації навчання базового класифікатора ваги зразків оновлюються: зразки, які були класифіковані неправильно, отримують вищі ваги, а правильно класифіковані – нижчі. Це змушує наступний базовий класифікатор більше уваги приділяти складним прикладам, поступово підвищуючи загальну точність.

4.3.2 Математичний апарат AdaBoost

Маємо початкові ваги зразків (формула 4.5):

$$w_i^{(1)} = \frac{1}{N}, i = 1, \dots, N \quad (4.5)$$

На кожній ітерації t від 1 до T ми повинні навчити базовий класифікатор $h^{(t)}(x)$ з урахуванням поточних ваг $w_i^{(t)}$.

Далі обчислюємо похибку класифікатора (формула 4.6)

$$\epsilon^{(t)} = \sum_{i=1}^N w_i^{(t)} I(y_i \neq h^{(t)}(x_i)) \quad (4.6)$$

де $I(\cdot)$ – індикаторна функція, що дорівнює 1, якщо умова виконується, та 0 в протилежному випадку.

Після чого ми обчислюємо ваговий коефіцієнт класифікатора (формула 4.7)

$$a^{(t)} = \frac{1}{2} \ln \left(\frac{1 - \epsilon^{(t)}}{\epsilon^{(t)}} \right) \quad (4.7)$$

де $a^{(t)}$ – коефіцієнт, який визначає вагу впливу базового класифікатора у кінцевому ансамблі.

При оновленні ваг зразків нам необхідно забезпечити наступну умову (формула 4.8)

$$\sum_{i=1}^N w_i^{(t+1)} = 1 \quad (4.8)$$

де $w_i^{(t)}$ – вага i -го зразка на ітерації t .

Для врахування описаної вище умови, розраховується нормувальна константа (формула 4.9).

$$Z^{(t)} = \sum_{j=1}^N w_j^{(t)} \exp(-a^{(t)} y_j * h^{(t)}(x_j)) \quad (4.9)$$

де x_j – це вхідні дані (особливості музичного треку), які подаються на вхід базовому класифікатору,

y_j – істинна мітка класу (жанру), яка вказує на правильну відповідь для зразка j ,

$h^{(t)}(x_j)$ – прогноз (клас) для j -го зразка, зроблений базовим класифікатором на ітерації t .

Тепер ми можемо оновити ваги зразків (формула 4.10):

$$w_i^{(t+1)} = \frac{w_i^{(t)} \exp(-a^{(t)} y_i h^{(t)}(x_i))}{Z^{(t)}} \quad (4.10)$$

Нарешті, отримаємо кінцевий прогноз ансамблю (формула 4.11):

$$H(x) = \text{sign} \left(\sum_{t=1}^T a^{(t)} h^{(t)}(x) \right) \quad (4.11)$$

де T – загальна кількість ітерацій (або базових класифікаторів), використаних в ансамблі.

4.4 Random Forests

4.4.1 Основи методу

Random Forests [11] – це ансамбль, що комбінує ідеї баггінгу та випадкового вибору ознак. В основі лежить побудова безлічі дерев рішень на бутстреп-вибірках початкових даних, де при кожному розщепленні вузла дерева випадково обирається підмножина ознак. Такий підхід знижує кореляцію між деревами, підвищує узагальнюючу здатність ансамблю і запобігає перенавчанню.

Random Forests має здібність працювати з пропущеними даними і гарно справляється з шумом та неструктурованою інформацією. Для задачі класифікації музичних треків за жанрами це означає, що метод здатний ефективно виділяти релевантні ознаки з великої кількості можливих характеристик музики, краще узагальнювати і ефективно використовувати різні спектральні та ритмічні індикатори жанру.

4.4.2 Математичний апарат Random Forests

Як і Bagging, Random Forests генерує B бутстреп-вибірок $D^{(b)}$.

На кожній вибірці навчається дерево рішень T_b . При розщепленні вузла, замість вибору ознак із усього набору M ознак, випадково обирають $m < M$ ознак і шукають найкраще розщеплення лише серед них.

Тоді кінцевий прогноз розраховується наступним чином (формула 4.12):

$$H(x) = \text{mode}\{T_1(x), T_2(x), \dots, T_B(x)\} \quad (4.12)$$

де B – кількість дерев у лісі,

m – кількість випадково обраних ознак при кожному розщепленні вузла,

$T_B(x)$ – прогноз B -го дерева для зразка x ,

$H(x)$ – кінцевий ансамблевий прогноз (за більшістю голосів дерев).

4.4.3 Вибір критерію розщеплення вузлів

Вибір критерію розщеплення у випадку Random Forests зазвичай не дає суттєвих відмінностей у кінцевій точності. Проте для конкретних застосувань, особливо в контексті класифікації музичних треків за жанрами, можуть бути деякі нюанси, на які варто звернути увагу.

Індекс Джині вважають дещо простішим з точки зору обчислень, адже для його розрахунку необхідно менше операцій. Ентропія теоретично забезпечує подібні результати, однак є трохи дорожчою обчислювально через логарифми.

Ентропія краще враховує рідкісні випадки, адже логарифмічна залежність сильніше «карає» неоднорідність у розподілі класів. У деяких випадках це може призвести до більш тонкого поділу, якщо жанрів багато та розподіл між ними є складним, в той час як індекс Джині схильний до швидшого ухвалення рішень.

Отже, якщо мета – максимізувати точність і немає критичних обмежень у часі, можна спробувати ентропію й перевірити, чи вона справді дає поліпшення на конкретному наборі даних. Якщо ж попередні експерименти показали несуттєву різницю між цими критеріями, доцільніше обрати індекс Джині.

4.5 Аналіз проблеми класифікації за багатьма жанрами

У випадку багатокласової класифікації, де існує більше ніж два жанри, задача ускладнюється тим, що нам потрібно визначити не просто, чи належить трек до одного з двох класів (наприклад, «рок» чи «не рок»), а знайти один правильний жанр серед багатьох. Для ансамблевих методів, спочатку розроблених для двокласових задач (як AdaBoost у своїй класичній формі), застосовують стратегії, що дозволяють узагальнити їх до багатокласової класифікації.

Одним із поширених підходів є розбиття багатокласової задачі на декілька двокласових підзадач. Наприклад, якщо є K жанрів, можна застосувати схему «один-проти-решти» (One-vs-Rest) або «один-проти-одного» (One-vs-One). Розглянемо ці підходи більш детально.

В підході One-vs-Rest для кожного жанру окремо тренується двокласовий класифікатор, який вирішує, чи належить трек до цього жанру, чи до будь-якого іншого. Отже, якщо у нас K жанрів, то ми отримаємо K двокласових моделей. Кожна з них дасть оцінку або голос «за» чи «проти» певного жанру. Наприклад, для жанру A є модель, що говорить: «Це A чи не A ?», для жанру B – «Це B чи не B ?» і так далі.

Потім, коли з'являється новий музичний трек, кожна з цих моделей дає свій прогноз разом із оцінками впевненості. Ми отримуємо K значень, кожне з яких відображає, наскільки модель упевнена, що трек належить до відповідного жанру. Фінальний жанр обирається як той, для якого модель «один-проти-решти» дала найбільшу впевненість або найбільший сумарний голос.

У підході One-vs-One для кожної пари жанрів тренується окремий двокласовий класифікатор. Якщо жанрів K , тоді моделі тренуються для кожної пари жанрів (наприклад, жанр A проти жанру B , жанр A проти жанру C і так далі). Загалом, буде $\frac{K(K-1)}{2}$ таких моделей.

Коли новий трек подається на вхід системі, кожна модель «один-проти-одного» вирішує, який із двох жанрів у парі більш підходить. У результаті трек «змагається» у своєрідному турнірі між усіма парами жанрів. Кожен класифікатор

«голосує» за перемогу одного з двох жанрів у своїй парі. Потім підраховують кількість перемог для кожного жанру у всіх парах, де він брав участь. Жанр, який набрав найбільше «голосів», оголошується кінцевим переможцем і вважається жанром, якому належить трек.

Незалежно від обраної схеми, головна ідея полягає в тому, що у багатокласовому випадку результати бінарних класифікацій порівнюються чи підсумовуються таким чином, щоб отримати остаточний жанр. Це може бути реалізовано за допомогою агрегування «голосів», обираючи жанру з найбільшим сумарним рейтингом або за допомогою інших стратегій інтерпретації результатів.

4.6 Теоретичні аспекти налаштування і підвищення ефективності моделей

Для кожного з описаних методів є важливим налаштування гіперпараметрів. Це включає вибір:

- для Bagging: кількість базових класифікаторів B , тип базового класифікатора;
- для AdaBoost: кількість ітерацій T , learning rate, тип слабкого класифікатора;
- для Random Forests: кількість дерев B , число ознак m при розщепленні вузла, максимальна глибина дерев.

Оптимізація гіперпараметрів може бути здійснена за допомогою крос-валідації та Баєсової оптимізації. При крос-валідації вихідні дані поділяються на так звані “folds”, а моделі тренуються та оцінюються на різних підмножинах даних для більш стабільної оцінки продуктивності. Баєсова оптимізація дозволяє вибрати гіперпараметри більш швидко, мінімізуючи кількість запусків моделей, але при її використанні можна пропустити набір параметрів, при якому модель матиме найбільшу точність.

Попередня обробка даних (масштабування, нормалізація, виділення ознак, зменшення розмірності) може суттєво вплинути на ефективність моделей. Обрання релевантних ознак та зменшення розмірності (наприклад, через PCA) допомагає моделі краще впоратися з високою складністю аудіоданих. Для

музичних треків доцільно використовувати спектральні ознаки, ритмічні та тональні характеристики, а потім застосовувати ансамблеві методи, що підвищують узагальнюючу здатність.

4.7 Теоретичні аспекти стійкості до перенавчання та адаптації до нових даних

Bagging, AdaBoost та Random Forests, хоча й є ансамблевими методами, вони мають різні підходи до зниження перенавчання. Bagging зменшує дисперсію за рахунок усереднення прогнозів незалежних класифікаторів. AdaBoost фокусується на складних зразках, покращуючи точність, але може бути більш чутливим до шуму. Random Forests знижують кореляцію між деревами, забезпечуючи кращу стійкість до перенавчання.

Таким чином, вибір конкретного ансамблю залежатиме від специфіки даних і задачі. Якщо важлива стабільність і стійкість до шуму – Bagging або Random Forests будуть надійними варіантами. Якщо пріоритет у підвищенні точності з урахуванням потенційного шуму, AdaBoost може бути доречним, але треба уважно ставитися до підбору параметрів та якості вхідних ознак.

4.8 Підготовка до експериментальних досліджень

Метою експериментального етапу є емпірична перевірка гіпотез, висунутих у процесі теоретичного дослідження, та виявлення найбільш ефективних ансамблевих методів у контексті класифікації музичних треків за жанрами. Для цього необхідно розробити чіткий план дій, який включатиме вибір та підготовку даних, налаштування моделей, визначення метрик оцінки, проведення крос-валідації та порівняння результатів.

Експеримент має початися з розподілу набору даних FMA (fma_small), що містить низку музичних треків із відомими жанровими анотаціями. Розподіл має відбутися у відповідному співвідношенні, наприклад 80% на тренування та 20% на тест. Такий поділ забезпечить незалежну оцінку продуктивності моделей.

Можливо створити також валідаційну вибірку, але зазвичай крос-валідація з тренувального набору буде більш надійною.

Далі необхідно вилучити релевантні ознаки із сирих аудіосигналів, або скористатися вже готовим набором ознак, підготованим заздалегідь дослідниками FMA. Головне – обрати ознаки, що показали свою ефективність у попередніх дослідженнях. Якщо необхідно, треба виконати нормалізацію або масштабування ознак (наприклад, до діапазону $[0;1]$ або стандартизацію), щоб уникнути домінування ознак із більшим масштабом. Також не буде зайвим розглянути можливість зменшення розмірності (наприклад, методом PCA), особливо якщо вихідний простір ознак дуже значний.

Для кожного методу визначаються гіперпараметри (описано у попередніх розділах) і визначаються методи оптимізації. Як більш пріоритетний метод розглядається Баєсова оптимізація.

Також важливим є вибір метрик для оцінки продуктивності, що найкраще відображають якість класифікації жанрів. Оскільки класи збалансовані, основною метрикою може бути Accuracy. Цю метрику можна доповнити F-мірою, повнотою та Precision. Також планується розглянути Confusion Matrix, щоб зрозуміти які жанри плутаються частіше.

Після вибору найкращих гіперпараметрів для кожного алгоритму треба провести крос-валідацію на тренувальному наборі, щоб отримати стабільну оцінку продуктивності. Для кожного алгоритму з найкращими параметрами зафіксувати середнє значення точності.

Залишається використати незалежну тестову вибірку, яка не використовувалася для навчання чи оптимізації, щоб отримати остаточну оцінку здатності моделі узагальнювати. Обчислюється точність, F1-міра та інші метрики на цьому тестовому наборі та порівнюються результати між моделями.

Таким чином, ми визначимо, яка модель дає найкращий компроміс між точністю, швидкістю (оцінкою часу навчання), використанням пам'яті та іншими критеріями, розглянутими у багатокритеріальному аналізі. Можна також

додатково розглянути вплив попередньої обробки даних (чи покращився результат після зменшення розмірності або нормалізації)

4.9 Побудова комбінованого ансамблю

Окрім індивідуального налаштування та оцінки кожного з трьох розглянутих методів, доцільно дослідити можливість їх одночасного поєднання в рамках більш складного ансамблю. Такий підхід дозволить скористатися перевагами кожного з методів та компенсувати їх недоліки.

Спочатку треба обрати схему комбінування методів:

- множинне голосування: створюється агрегатор прогнозів трьох найкращих моделей, отриманих після оптимізації гіперпараметрів (фінальний клас буде визначатися за більшістю голосів, наприклад, якщо Bagging та AdaBoost сходяться на одному жанрі, а Random Forests обирає інший, приймається той жанр, за який проголосували перші два);
- зважене голосування: за результатами попередніх експериментів можна визначити рівень довіри до кожної моделі та призначити їй вагу;
- багаторівневе ансамблювання (stacking): створюється мета-класифікатор, який навчається на вихідних логітах або ймовірностях, що повертаються трьома окремими моделями, цей мета-класифікатор може бути простою моделлю (наприклад, логістичною регресією), яка вчиться, як найкраще комбінувати результати трьох базових ансамблів.

Планується обрати stacking, тож необхідно буде провести додаткову крос-валідацію, щоб навчити мета-класифікатор.

Після цього залишається порівняти і проаналізувати, чи комбінований ансамбль дає вищу точність, F-міру чи інші метрики. Також варто перевірити, чи зменшилася дисперсія результатів, та чи модель стала стійкішою до шуму.

Якщо результати виявляться кращими, то комбінований ансамбль буде рекомендовано як фінальне рішення для класифікації музичних треків за жанрами.

5 ПРОГРАМНА РЕАЛІЗАЦІЯ

5.1 Загальна архітектура програмного забезпечення

Програмне середовище обране для виконання експериментів [12, 13], реалізовано здебільшого на мові Python із використанням низки добре відомих бібліотек машинного навчання та інструментів наукових обчислень, зокрема:

- numPy та pandas для зручної роботи з масивами та таблицями даних;
- scikit-learn для побудови моделей класифікації та інших утиліт;
- optuna для гнучкої й ефективної оптимізації гіперпараметрів;
- itertools для формування комбінацій параметрів при пошуку найкращої структури ознак і початкових налаштувань моделі;
- matplotlib для візуалізації результатів і залежностей.

Основна ідея полягає в тому, щоб надати програмі модульну структуру, де кожен етап реалізований як окремий блок коду, починаючи від завантаження та попередньої обробки даних, формування ознак і закінчуючи тестуванням моделей. Це дає можливість оперативно вносити зміни, проводити нові серії експериментів і розширювати порівняльний аналіз, додаючи нові алгоритми чи модифікуючи існуючі.

Послідовність етапів під час проведення одного типового циклу експериментів є наступною:

- завантаження даних і фільтрація за жанрами (шість обраних жанрів);
- попередня обробка (нормалізація, масштабування, вилучення сильно корельованих ознак) та формування підмножин ознак;
- початкове навчання простої моделі (decision tree), для визначення базових орієнтирів точності та виділення найбільш корисних гіперпараметрів;
- побудова ансамблевих моделей (Bagging, AdaBoost, RF) для кожної з яких реалізовано окремі функції налаштування та навчання;
- оптимізація гіперпараметрів за допомогою Optuna, щоб адаптивно звужити пошук у просторі параметрів і знайти найліпшу конфігурацію з

точки зору точності класифікації;

- створення Stacking-ансамблю, де мета-класифікатор навчається на out-of-fold прогнозах базових моделей;
- оцінювання результатів із використанням крос-валідації (CV=5) та аналіз стабільності точності.

Таким чином, архітектура програмного забезпечення орієнтована на модульність і прозору інтеграцію класичних та нових методів, а також гнучкий підхід до налаштувань.

5.2 Підготовка та відбір ознак

У межах поточного дослідження підготовці ознак приділено особливу увагу, адже саме якість вихідних дескрипторів (features) музичних треків багато в чому визначає успіх загальної класифікації. За результатами попередніх робіт, що було проаналізовано, було вирішено зосередитися на звукових та спектральних характеристиках (chroma, MFCC, Zero Crossing Rate, тощо), оскільки вони вважаються найбільш показовими для розрізнення жанрів.

Перший етап передбачав первинний відбір і фільтрацію ознак за кореляційним аналізом. Видалялися ознаки, що мали кореляцію з іншою ознакою вище за 0.95 (або нижче за -0.95). Цей поріг обрано з огляду на баланс між усуненням явних дублювань і збереженням більш тонкої інформації.

На наступному кроці розглядалися три напрямки:

- filter methods (Anova test): оцінка статистичних відмінностей між групами жанрів і вибір тих ознак, які найбільше пояснюють різницю в категоріях;
- wrapper methods: рекурсивне вилучення ознак (RFE) з використанням допоміжної моделі, а також генетичний алгоритм для оптимізації набору ознак;
- intrinsic methods (Preliminary Random Forest): використання попереднього навченого лісу для оцінки значущості ознак за критерієм зменшення невідповідності (impurity-based feature importance).

Практичні експерименти свідчать, що жоден із вищенаведених підходів до подальшого відбору ознак не покращив підсумкову точність класифікації. Найкращі результати досягалися тоді, коли лише усувалися найбільш корельовані дескриптори. Це пояснюється тим, що у вихідних даних `fma_small` є досить тонка мережа кореляцій між певними фоновими звуковими і спектральними характеристиками, і спроба вилучення частини з них руйнувала цю структуру, що негативно впливало на ефективність моделей.

Варто зазначити, що PCA та інші лінійні методи зменшення розмірності загалом також знижують точність. Ймовірно, це пов'язано з тим, що дані з `fma_small` містять багатовимірну структурну інформацію, і проекція у менший простір не може зберегти всі взаємозв'язки. Саме тому у фінальній реалізації вирішено не застосовувати активні методи відбору чи зменшення розмірності.

5.3 Використання `itertools` для первинного навчання

Ключовим моментом у розробці моделей для класифікації музичних жанрів є визначення найкращого набору ознак для моделювання. Перед тим як приступити до тренування складних моделей, було вирішено виконати перебір існуючих ознак і отримати проміжні результати точності на спрощеній моделі — `Decision Tree Classifier`. Роль цієї моделі полягала у виявленні найбільш влучного набору ознак, який забезпечує найбільшу різницю в результаті.

Для ефективної реалізації перебору комбінацій ознак використовувався модуль `itertools`, який дозволяє автоматично формувати декартові добутки можливих комбінацій дескрипторів. Цей підхід дозволяє систематично і повністю дослідити усі можливі підмножини наявних ознак.

Завдяки цьому механізму було автоматизовано процес генерування наборів, що значно скоротило час і ресурси необхідні на проведення експериментів. Крім того, використання `itertools` гарантувало повноту перебору комбінацій, залишаючи усі потенційно перспективні набори ознак.

Нижче наведено приклад програмного коду для визначення найкращої комбінації дескрипторів (див. рис. 5.1) із накладанням відповідних обмежень на

обчислювальні ресурси, що може використати програма. Окрім того, сам пошук виконується паралельно з використанням усіх ядер процесора.

```

all_combos = []
for k in range(2, max_comb_size+1):
    all_combos.extend(itertools.combinations(feature_groups.keys(), k))
rng.shuffle(all_combos)
all_combos = all_combos[:max_evals]

def _eval(combo):
    cols = sum((feature_groups[g] for g in combo), [])
    score = evaluate_combo(cols, X_full, y_full, sample_size, random_state)
    return combo, score

results = Parallel(n_jobs=n_jobs)(
    delayed(_eval)(cmb) for cmb in tqdm(all_combos, desc="Searching")
)

best_combo, best_score = max(results, key=lambda x: x[1])
return best_combo, best_score, results

```

Рисунок 5.1 – Уривок коду пошуку початкової комбінації ознак (рисунок створено самостійно)

Отримані результати перебору дозволили визначити найбільш інформативний набір ознак, який згодом було використано як основу для побудови складніших ансамблевих моделей.

5.4 Оптимізація гіперпараметрів за допомогою Optuna

У межах дослідження метою було знайти глобально найліпшу комбінацію значень параметрів у великому числовому просторі. Традиційні методи, як Grid Search, можуть виявитися надзвичайно ресурсомісткими, оскільки вони перебирають кожен набір параметрів у заданих діапазонах. Random Search, хоч і швидший, не гарантує руху до оптимуму. Натомість Optuna забезпечує більш «розумний» підхід через використання баєсових і еволюційних стратегій, які поступово звужують область пошуку після кожного набору випробувань.

Основний принцип обраної бібліотеки полягає в тому, що на кожному кроці вона вивчає, які конфігурації гіперпараметрів виявилися вдалими, та коригує процес вибору параметрів на наступних ітераціях таким чином, щоб більше спроб припадало на «перспективні» зони простору. Це дає змогу скоротити загальну

кількість потрібних ітерацій і підвищити ймовірність знайти оптимальні параметри.

Optuna керує поняттям `study` (об'єкт, в якому зберігається інформація про поточний експеримент). Кожен виклик методу `study.optimize()` запускає послідовність спроб, у межах яких виконується навчання моделі з певним набором гіперпараметрів. Результат точності передається назад в бібліотеку, і алгоритм ухвалює рішення, які саме гіперпараметри варто перевірити на наступних кроках.

Нижче наведено приклад коду для оптимізації логістичної регресії, що складається з чотирьох моделей (див. рис. 5.2).

```
def objective_4model(trial):
    """Objective function for 4-model stacking."""
    C = trial.suggest_float('C', 1e-3, 1e3, log=True)

    lr_meta = LogisticRegression(
        C=C,
        penalty='l2',
        solver='lbfgs',
        max_iter=1000,
        multi_class='auto',
        class_weight=None,
        random_state=999
    )

    lr_meta.fit(oof_probs_train_opt, y_train_opt)
    val_preds = lr_meta.predict(val_probs_opt)
    return accuracy_score(y_val_opt, val_preds)

study_4model = optuna.create_study(direction='maximize')
study_4model.optimize(objective_4model, n_trials=1000)
```

Рисунок 5.2 – Програмний код оптимізації моделі за допомогою Optuna (рисунок створено самостійно)

Варто підкреслити, що схожа стратегія використовувалася для базових моделей та ансамблів.

5.5 Реалізація ансамблевого навчання

Після пошуку найкращих дескрипторів, розробки базової моделі та оптимізації за допомогою Optuna, ми визначили в програмному коді базові ансамблі для Bagging, AdaBoost та Random Forest.

Для створення кожної моделі використовувалися функції-конструктори. Нижче наведено приклад програмного коду такої функції (див. рис. 5.3)

```
def create_bagging():
    best_params_bagging = {
        'n_estimators': 264,
        'max_samples': 0.892487331464192,
        'max_features': 0.67051206388554,
        'bootstrap': False,
        'max_depth': 11,
        'min_samples_split': 12
    }
    base_tree = DecisionTreeClassifier(
        max_depth=best_params_bagging['max_depth'],
        min_samples_split=best_params_bagging['min_samples_split'],
        random_state=213
    )
    return BaggingClassifier(
        estimator=base_tree,
        n_estimators=best_params_bagging['n_estimators'],
        max_samples=best_params_bagging['max_samples'],
        max_features=best_params_bagging['max_features'],
        bootstrap=best_params_bagging['bootstrap'],
        n_jobs=-1,
        random_state=42
    )
```

Рисунок 5.3 – Функція-конструктор для Bagging (рисунок створено самостійно)

Подібні параметри було отримано в результаті ретельного пошуку за допомогою Optuna для кожної моделі.

Хоча під час багатокритеріального аналізу модель SVC була виключена з огляду на її високу обчислювальну складність, проблеми з масштабуванням та складність налаштування, практичні експерименти продемонстрували, що саме ця модель досягає найвищої точності на обраному наборі даних.

Саме висока точність цієї моделі та її здатність добре узагальнювати складні взаємозв'язки в ознаковому просторі стала причиною її включення до фінального ансамблю. Для SVC було обрано ядро типу RBF (Radial Basis Function), оскільки саме воно забезпечує нелінійне перетворення ознакового простору та дозволяє ефективно розмежовувати складні, нелінійно розподілені класи музичних жанрів. Після тестування декількох типів ядер (лінійного, поліноміального, сигмоїдального), саме RBF показало найкращу здатність до узагальнення і стабільно демонструвало найвищу точність класифікації на тестових вибірках.

Після визначення базових ансамблів і SVC, наступним кроком стала побудова мета-класифікатора за допомогою `stacking`. Було використано логістичну регресію, що навчається на ймовірностях (`probability output`) чотирьох моделей (`Bagging`, `AdaBoost`, `Random Forest`, `SVC`), отриманих методом `Out-of-Fold (OOF)` прогнозів:

- для кожного з 5 фолдів крос-валідації навчалися базові моделі з оптимальними параметрами;
- на кожному фолді базові моделі формували прогнози у вигляді ймовірностей для відповідної валідаційної вибірки, після чого ці прогнози поєднувалися в єдину матрицю OOF-прогнозів, яка слугувала навчальними даними для мета-класифікатора;
- в результаті логістична регресія навчалася на цій матриці, по суті навчаючись, якій моделі більше “довіряти” для певних жанрів.

Таке поєднання моделей дозволило отримати результати, які перевершують точність моделей, що працюють поодиноці. Причиною цього є те, що базові моделі мають різні принципи навчання та різні помилки. Наприклад, `SVC` ефективно розмежовує складні нелінійні області, тоді як ансамблі на основі дерев (`Bagging`, `AdaBoost`, `RF`) краще обробляють складні структури даних із чіткими межами поділу. Таким чином, кожна з моделей доповнює слабкі місця інших, що й пояснює покращення результату після об'єднання.

5.6 Результати експериментів

Після реалізації всіх моделей і ансамблів було проведено серію експериментів із 5-кратною крос-валідацією, за результатами яких отримано підсумкову оцінку точності (`accuracy`) та `F1`-міри.

Для наочної демонстрації отриманих результатів було побудовано графіки за допомогою бібліотеки `matplotlib`, які чітко відображають ефективність кожної моделі та переваги `Stacking`-ансамблю над окремими класифікаторами.

Нижче наведено результати фінальних експериментів (див. рис. 5.4).

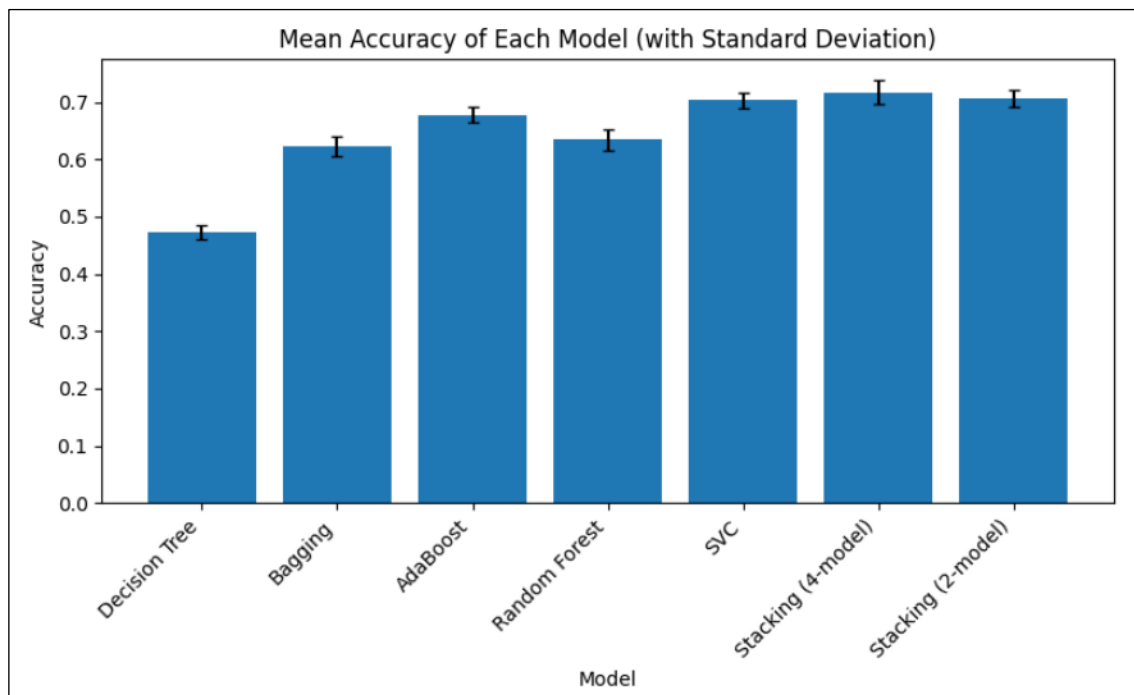


Рисунок 5.4 – Середня точність отриманих моделей (рисунок створено самостійно)

Stacking (2-model) складається лише з AdaBoost та SVC. Такий комбінований ансамбль показав покращення середньої точності лише на 0.2% порівняно з найкращою одиночною моделлю SVC. Це свідчить про те, що хоча ці дві моделі й відрізняються за своїм принципом роботи, вони недостатньо ефективно доповнюють одна одну, оскільки кількість різноманітних патернів помилок у такій парі залишається обмеженою. Проте, коли кількість базових моделей зростає до чотирьох (додаємо в ансамбль Bagging та Random Forest), можемо бачити, що Stacking (4-model) надає значно більш відчутне покращення (1-2.5% в залежності від фолду). Це пояснюється тим, що додаткові моделі на основі дерев рішень привносять більше варіативності у прогнозі ймовірності, суттєво підвищуючи узагальнюючу спроможність ансамблю.

Таким чином, мета-класифікатор на основі логістичної регресії отримує більше інформації для ефективного вибору правильного рішення, що безпосередньо відображається на прирості точності класифікації.

Нижче наведено окремий графік порівняння продуктивності SVC та комбінованого ансамблю з чотирьох моделей (див. рис. 5.5).

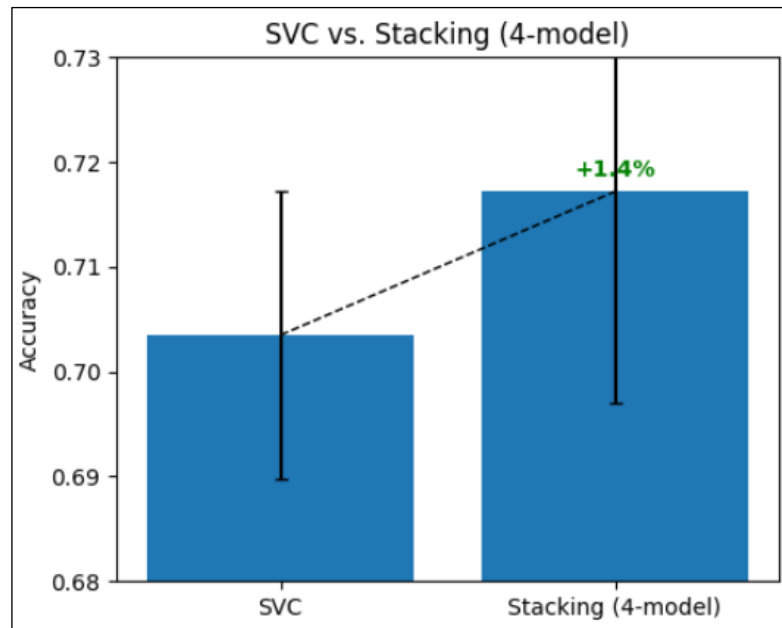


Рисунок 5.5 – Порівняння SVC та мета-класифікатора (рисунок створено самостійно)

Окрім того, було побудовано графік F1-міри кожної з моделей (див. рис. 5.6).

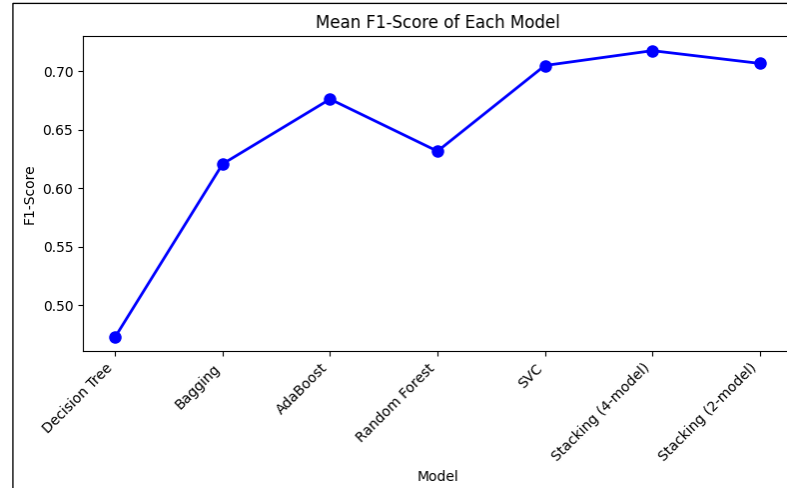


Рисунок 5.6 – F1-міра кожної моделі (рисунок створено самостійно)

F1-міра враховує також precision та повноту (recall), що робить її більш чутливою до різних типів помилок. Незважаючи на цей більш складний характер метрики, ми все одно спостерігаємо покращення якості класифікації комбінованого ансамблю з чотирьох моделей (SVC, Bagging, RF, AdaBoost). Це свідчить про те, що переваги ансамблю не обмежуються лише одним аспектом

якості, а проявляються комплексно.

Хоча за принципом Парето модель Bagging отримала найвищу оцінку через свої переваги у використанні пам'яті, можливості паралелізації та відносно швидкого навчання, на практиці окрема модель Bagging не демонструє найвищих результатів точності. Це пояснюється тим, що, попри ефективність зниження дисперсії за рахунок ансамблювання базових дерев, Bagging має схожі індуктивні припущення з іншими моделями на основі дерева, що обмежує його здатність виправляти систематичні помилки. Проте, даний результат узгоджується з обраними формальними критеріями, які були встановлені на етапі попереднього аналізу, і є задовільним з точки зору загальної ефективності та обчислювальної простоти. Крім того, Bagging робить значний внесок у продуктивність фінального stacking-ансамблю, оскільки його різноманітні прогностні патерни допомагають компенсувати слабкі сторони інших моделей, що в сукупності забезпечує підвищення загальної точності комбінованого класифікатора.

Варто зауважити, що спроба побудови ансамблю з трьох моделей, які базуються виключно на деревах рішень (AdaBoost, Bagging, RF), виявилася неефективною для підвищення точності класифікації. Аналіз показав, що всі базові моделі з дерева рішень мають схожі індуктивні припущення та властивості, що призводить до того, що вони роблять практично однакові помилки. У результаті комбінування їхніх прогнозів, навіть при застосуванні методів агрегації (напр. голосування), ансамбль не може компенсувати систематичні похибки, властиві кожній окремій моделі. Це свідчить про те, що для досягнення істотного приросту точності необхідно використовувати базові моделі з різними алгоритмічними підходами.

Таким чином, включення SVC до фінального ансамблю навіть після того, як він був вилучений зі списку на підставі принципу Парето (через високі витрати часу на налаштування, проблеми паралелізації та асимптотичну складність), знайшлося цілковито виправданим на практиці. В контексті реальних даних FMA і конкретних жанрів SVC підтвердив свою ефективність.

ВИСНОВКИ

У ході виконання роботи було досягнуто низку важливих результатів, що узгоджуються із поставленими завданнями щодо застосування ансамблевих методів для класифікації музичних треків за жанрами.

По-перше, було здійснено аналіз предметної галузі та виявлено основні чинники, які роблять задачу класифікації музичних жанрів складною. Серед них – варіативність музичних патернів, наявність змішаних стилів, а також суб'єктивність людського сприйняття музики. Цей аналіз підтвердив актуальність вибору саме ансамблевих методів, адже вони здатні ефективно працювати зі складними наборами ознак і більш стійко протистояти перенавчанню.

По-друге, було розглянуто задачу багатокритеріального вибору оптимальних моделей, яка включала оцінювання п'яти критеріїв (інтерпретованість, використання пам'яті, асимптотична складність, складність підбору параметрів, можливість паралелізації) та застосування лінійної адитивної згортки з ваговими коефіцієнтами. Унаслідок цього аналізу сформовано множину Парето, з якої метод Bootstrapping Ensemble (Bagging) показав найвищу корисність, що підтверджує його доцільність для задачі класифікації жанрів.

По-третє, було проведено теоретичне дослідження математичного апарату трьох обраних ансамблевих методів – Bagging, AdaBoost і Random Forests – включно з формулами та алгоритмічними схемами. З'ясовано, що кожний метод знижує ризик перенавчання різними шляхами: Bagging – за рахунок усереднення незалежних підмоделей, AdaBoost – шляхом посиленого навчання на «важких» зразках, а Random Forests – завдяки випадковому вибору ознак і бутстреп-вибіркам. Така різноманітність підходів свідчить про необхідність емпіричного тестування кожного з них у реальних умовах.

По-четверте, було поставлено низку завдань щодо налаштування гіперпараметрів, зокрема кількості базових класифікаторів, глибини дерев та параметрів швидкості навчання. Визначено, що оптимізація цих параметрів за допомогою басової оптимізації є важливим кроком у підготовці до практичної реалізації моделей.

По-п'яте, у межах планування експериментів із використанням набору даних FMA (fma_small) були сформульовані метрики оцінювання якості класифікації (Accuracy, Precision, Recall, F-міра) та запропоновано структуру проведення порівняльного аналізу результатів. Також окреслено можливість створення комбінованого ансамблю (stacking), що може додатково підвищити точність і стійкість моделей.

По-шосте, було програмно реалізовано усі найкращі обрані моделі класифікації музичних треків за жанром, розроблено комбінований ансамбль з чотирьох моделей, підтверджено достовірність покращення продуктивності від використання мета-класифікатора.

Наукова цінність роботи полягає у порівнянні та узгодженні підходів багатокритеріального аналізу (за принципом Парето) із реальними експериментами на даних FMA, а також у демонстрації важливості алгоритмічної гнучкості (SVC, незважаючи на «непріоритетність» у формальних критеріях, показав високі емпіричні результати).

Практична значущість полягає в розробці методичних засад (Optuna, stacking, попередній аналіз ознак) та готової програмної реалізації, яку можна розгорнути в інформаційних системах.

Щодо напрямків подальших досліджень, варто розглянути варіант розширення ансамблю за рахунок методів глибинного навчання (наприклад, прості CNN чи CRNN) і мультимодальних дескрипторів (метадані, тексти тощо).

Таким чином, усі поставлені задачі – від аналізу предметної галузі й багатокритеріального відбору методів до теоретичного дослідження й програмної реалізації – було успішно виконано. Отримані результати підтверджують перспективність застосування ансамблевих методів для класифікації музичних треків за жанрами й надають підґрунтя для подальших досліджень в цій сфері.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Mohamed Al Kilani, Volodymyr Kobziev. An Overview of Research Methodology in Information System (IS) // Open Access Library Journal. – Vol.03 No.11(2016), Article ID:71775. – 9 pages. DOI:10.4236/oalib.1103126
2. Методичні вказівки до лабораторних робіт та практичних занять з дисципліни «Теорія ігор та прийняття рішень» для студентів денної форми навчання спеціальностей 8.05010301 – «Програмне забезпечення систем», 8.05010302 – «Інженерія програмного забезпечення» / Мазурова О.О. – Харків: ХНУРЕ, 2016. – 34 с.
3. Liu Z., De Mori J., Abayomi K. Open Set Recognition for Music Genre Classification // Proceedings of the 23rd International Society for Music Information Retrieval Conference. – 2022.
4. Pons J., Lidy T., Serra X. Experimenting with musically motivated convolutional neural networks // Proceedings of the 20th International Society for Music Information Retrieval Conference (ISMIR). – 2019. – P. 287–293.
5. Choi K., Fazekas G., Sandler M., Cho K. Convolutional Recurrent Neural Networks for Music Classification // Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). – 2017.
6. Meng Y. Music Genre Classification: A Comparative Analysis of CNN and XGBoost Approaches with Mel-frequency cepstral coefficients and Mel Spectrograms // arXiv preprint arXiv:2401.04737. – 2024.
7. Zhang J. Music Genre Classification with ResNet and Bi-GRU Using Visual Spectrograms // arXiv preprint arXiv:2307.10773. – 2023.
8. Ru G., Zhang X., Wang J., Cheng N., Xiao J. Improving Music Genre Classification from Multi-Modal Properties of Music and Genre Correlations Perspective // arXiv preprint arXiv:2303.07667. – 2023.
9. Kadiyala A., Kumar A. Applications of Python to Evaluate the Performance of Bagging Methods // Environmental Progress & Sustainable Energy. – 2018. – DOI: 10.1002/ep.13018.
10. Žižka J., Dařena F., Svoboda A. Adaboost // Text Mining with Machine

Learning. 1st Edition / CRC Press, 2019. – 10 p. – eBook ISBN: 9780429469275

11. Altaf I., Butt M.A., Zaman M. Systematic Consequence of Different Splitting Indices on the Classification Performance of Random Decision Forest // 2022 2nd International Conference on Intelligent Technologies (CONIT). – DOI: 10.1109/CONIT55038.2022.9848372.

12. Movchan A. Ensemble-Based Approaches for Music Track Classification // Grail of Science. – No. 2(187), February 2025. – DOI: 10.36074/grail-of-science.21.02.2025

13. Movchan A. ML Stacking Experiment: Full code, Specification and presentation materials. GitHub repository. Accessed June 3, 2025 https://github.com/Climclaff/ML_Stacking_experiment

**ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ ЗА НАУКОВИМИ НАПРЯМАМИ
КЕРІВНИКА ТА НАУКОВЦІВ КАФЕДРИ ПРОГРАМНОЇ ІНЖЕНЕРІЇ**

1. Mohamed Al Kilani, Volodymyr Kobziev. An Overview of Research Methodology in Information System (IS) // Open Access Library Journal. – Vol.03 No.11(2016), Article ID:71775. – 9 pages. DOI:10.4236/oalib.1103126

2. Методичні вказівки до лабораторних робіт та практичних занять з дисципліни «Теорія ігор та прийняття рішень» для студентів денної форми навчання спеціальностей 8.05010301 – «Програмне забезпечення систем», 8.05010302 – «Інженерія програмного забезпечення» / Мазурова О.О. – Харків: ХНУРЕ, 2016. – 34 с.