

ЧЕРНІЙ Іван Русланович

**Рекомендаційний модуль вибору музичних
композицій на основі машинного навчання/Music
selection recommendation module based on
machine learning**

спеціальність: 122 - Комп'ютерні науки
освітньо-професійна програма - Комп'ютерні науки

Кваліфікаційна робота

Виконав студент групи КН-42
І. Р. Черній

Науковий керівник:
к.і.н., Г. В. Сапожник

Кваліфікаційну роботу
допущено до захисту:

" ____ " _____ 20__ р.

Завідувач кафедри

_____ **М. П. Комар**

ТЕРНОПІЛЬ - 2024

Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління
Освітній ступінь «бакалавр»
спеціальність 122 – Комп'ютерні науки
освітньо-професійна програма – Комп'ютерні науки

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ М.П. Комар
« _____ » _____ 2023 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ
ЧЕРНІЮ Івану Руслановичу
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи: Рекомендаційний модуль вибору музичних композицій на основі машинного навчання/Music selection recommendation module based on machine learning

керівник роботи Сапожник Григорій Вікторович, к.іст.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від 12 грудня 2023 р. № 753.

2. Строк подання студентом закінченої кваліфікаційної роботи 15 травня 2024 р.

3. Вихідні дані до кваліфікаційної роботи: завдання на кваліфікаційну роботу студента, наукові статті, технічна література.

4. Основні питання, які потрібно розробити:

– Аналіз та систематизація існуючих досліджень у галузі музичних рекомендаційних систем.

– Розробка алгоритмічного забезпечення для модулю рекомендацій.

– Експериментальне моделювання та оцінка архітектури рекомендаційного модулю.

– Підготовка та обробка даних, що включає збір, нормалізацію та трансформацію музичних метаданих.

– Інтеграція та тестування модулю рекомендацій.

– Оцінка користувацького досвіду та збір зворотного зв'язку.

– Адаптація та оптимізація системи на основі отриманого зворотного зв'язку та аналітичних даних про використання системи.

5. Перелік графічного матеріалу в роботі:

– Архітектура модулю рекомендацій музики на основі машинного навчання

– Структура виводу рекомендації музичних композицій на основі машинного навчання

– графіки з результатами експериментальних досліджень.

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 12 грудня 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів кваліфікаційної роботи	Примітка
1	Затвердження теми кваліфікаційної роботи, ознайомлення з літературними джерелами та складання плану роботи.	до 01.01. 2024 р.	
2	Написання 1 розділу кваліфікаційної роботи	до 01.03. 2024 р.	
3	Написання 2 розділу кваліфікаційної роботи	до 01.04.2024 р.	
4	Написання 3 розділу кваліфікаційної роботи	до 01.05. 2024 р.	
5	Представлення попереднього варіанту кваліфікаційної роботи, перевірка та внесення змін керівником	до 15.05.2024 р.	
6	Опрацювання зауважень та представлення завершеного варіанту кваліфікаційної роботи. Підготовка супроводжуючих документів.	до 20.05.2024 р.	
7	Перевірка кваліфікаційної роботи на оригінальність тексту у системі «Unicheck».	до 10.06.2024 р.	
8	Оформлення кваліфікаційної роботи та отримання допуску до захисту	до 14.06.2024 р.	
9	Подання кваліфікаційної роботи до захисту на засіданні атестаційної комісії.	до 14.06. 2024 р.	

Студент _____ І. Р. Черній
 (підпис) (прізвище та ініціали)

Керівник кваліфікаційної роботи _____ Г.В. Сапожник
 (підпис) (прізвище та ініціали)

АНОТАЦІЯ

Кваліфікаційна робота на тему «Рекомендаційний модуль вибору музичних композицій на основі машинного навчання» на здобуття освітнього ступеня «бакалавр» зі спеціальності 122 «Комп'ютерні науки» освітньої програми «Комп'ютерні науки» написана обсягом в 36 сторінок і містить 8 ілюстрацій, 6 таблиць та 11 використаних джерел.

Метою роботи є розробка та оптимізація рекомендаційного модуля вибору музичних композицій заснованого на передових методах машинного навчання для забезпечення високої точності і персоналізації музичних рекомендацій для користувачів стрімінгових платформ.

Методами розроблення обрано метод аналізу (для дослідження існуючих підходів до рекомендаційних систем), метод синтезу (для поєднання переваг існуючих методів), методи моделювання (для представлення та дослідження процесів рекомендацій), метод порівняльного аналізу (для оцінювання адекватності моделі рекомендацій).

Внаслідок виконання роботи обґрунтовано раціональний підхід до розроблення моделей рекомендацій та розроблено програмний засіб, який дозволяє створювати і досліджувати моделі рекомендацій.

Результати дослідження можуть бути використані в науково-дослідних установах і підрозділах підприємств, що займаються розробленням моделей рекомендацій.

Ключові слова: РЕКОМЕНДАЦІЙНІ СИСТЕМИ, МАШИННЕ НАВЧАННЯ, КОСИНУСНА ПОДІБНІСТЬ, К-НАБЛИЖЧИХ СУСІДІВ, ПЕРСОНАЛІЗАЦІЯ.

ANNOTATION

Qualification work on the topic «Music selection recommendation module based on machine learning» for Bachelor's degree on speciality 122 «Computer Science» educational and professional program «Computer Science» is written on 36 pages and it contains 8 figures, 6 tables, and 11 sources.

The purpose of the work is to develop and optimize a music selection recommendation module based on advanced machine learning methods to ensure high accuracy and personalization of music recommendations for users of streaming platforms.

Research methods include analysis (to study existing approaches to recommendation systems), synthesis (to combine the advantages of existing methods), modeling (to represent and study recommendation processes), and comparative analysis (to evaluate the adequacy of the recommendation model).

As a result of the work, a rational approach to the development of recommendation models was substantiated, and a software tool was developed that allows creating and researching recommendation models.

The research results can be used in research institutions and enterprise departments involved in the development of recommendation models.

Keywords: RECOMMENDATION SYSTEMS, MACHINE LEARNING, COSINE SIMILARITY, K-NEAREST NEIGHBORS, PERSONALIZATION.

ЗМІСТ

Вступ	7
1 Аналіз предметної області і постановка задачі дослідження	10
1.1 Огляд предметної області.....	10
1.2 Огляд існуючих рішень	13
1.3 Вибір перспективного шляху і постановка задачі дослідження	16
2 Алгоритмічне забезпечення рекомендаційного модуля вибору музичних композицій на основі машинного навчання	19
2.1 Архітектура модуля рекомендацій музики на основі машинного навчання	19
2.2 Опис підходу косинусної подібності	21
2.3 Опис алгоритму К найближчих сусідів	22
2.4 Алгоритм виводу рекомендації музичних композицій на основі машинного навчання	23
3 Програмно-технологічне забезпечення	26
3.1 Підготовка даних.....	26
3.2 Модуль рекомендацій	33
3.2.1 Рекомендація на основі косинусної-подібності.....	33
3.2.2 Рекомендація на основі К найближчих сусідів.....	36
Висновки	38
Список використаних джерел	40
Додаток А Код для реалізації.....	42
Додаток Б Характеристики треків Spotify	46
Додаток В Апробація отриманих результатів.....	48

ВСТУП

Актуальність розробки музичних рекомендаційних систем на основі машинного навчання постійно зростає в умовах стрімкого розвитку цифрових медіа та змін у споживацьких вподобаннях. З огляду на те, що кількість музичного контенту, доступного онлайн, збільшується щодня, здатність точно та ефективно рекомендувати композиції, які відповідають індивідуальним уподобанням користувачів, стає ключовою для успіху музичних платформ. Персоналізовані рекомендації не тільки підвищують задоволеність користувачів, але й сприяють більш тривалому залученню їх до платформи, що є важливим для комерційної вигоди компаній. Такі системи також сприяють культурному розмаїттю, даючи можливість відкривати та популяризувати менш відомих виконавців поряд з естаблішментом.

Розробка і оптимізація таких систем є важливою не тільки з комерційної точки зору, але й з точки зору збагачення музичного досвіду користувачів. Впровадження передових технологій, зокрема глибокого навчання та штучного інтелекту, дозволяє точніше аналізувати великі обсяги даних і виявляти складні патерни поведінки слухачів. Це не тільки підвищує точність рекомендацій, але й надає можливість пропонувати музику, яка може відповідати емоційному стану або конкретній ситуації. Тому розробка інноваційних рекомендаційних систем є критично важливою для подальшого розвитку музичної індустрії та забезпечення високоякісного музичного сервісу.

Метою роботи є розробка та оптимізація рекомендаційного модуля вибору музичних композицій, заснованого на передових методах машинного навчання, щоб забезпечити високу точність і персоналізацію музичних рекомендацій для користувачів стрімінгових платформ. Для досягнення цієї мети необхідно вирішити наступні завдання:

1. Аналіз та систематизація існуючих досліджень у галузі музичних рекомендаційних систем.
2. Розробка алгоритмічного забезпечення для модулю рекомендацій.
3. Експериментальне моделювання та оцінка архітектури рекомендаційного модулю.
4. Підготовка та обробка даних, що включає збір, нормалізацію та трансформацію музичних метаданих.
5. Інтеграція та тестування модулю рекомендацій.
6. Оцінка користувацького досвіду та збір зворотного зв'язку.
7. Адаптація та оптимізація системи на основі отриманого зворотного зв'язку та аналітичних даних про використання системи.

Об'єктом дослідження є музичні рекомендаційні системи на основі машинного навчання.

Предметом дослідження є методи та алгоритми машинного навчання, що використовуються у музичних рекомендаційних системах для аналізу, обробки та генерації персоналізованих музичних пропозицій.

Методи дослідження в нашому проекті включають використання колаборативної фільтрації та глибокого навчання для аналізу користувацьких даних та вподобань, експериментальне тестування алгоритмів косинусної подібності та K-найближчих сусідів для оцінки ефективності рекомендацій, а також квантитативний аналіз з використанням статистичних методів для вимірювання точності та релевантності рекомендацій.

Практичне значення нашого дослідження полягає в розробці удосконаленої музичної рекомендаційної системи, яка використовує алгоритми машинного навчання для забезпечення високої точності та персоналізації музичних пропозицій. Система спрямована на підвищення задоволеності користувачів, оптимізацію їхнього досвіду прослуховування та збільшення часу, проведеного на музичних платформах. Це також включає покращення монетизації для платформ через залучення та утримання

користувачів, забезпечуючи більш ефективне рекламне таргетування та підвищення конверсії. Результати дослідження можуть бути застосовані на практиці в індустрії цифрових медіа та розваг, надаючи значні переваги як для кінцевих користувачів, так і для бізнесу, що займається музичним контентом.

Структура та розмір дослідження. Кваліфікаційна робота складається з вступу, трьох розділів, висновків та списку використаних джерел. Загальний обсяг роботи становить 36 сторінок тексту, включаючи 8 рисунків та 6 таблиць. У списку використаних джерел наведено 11 найменувань, які займають 2 сторінки.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

1.1 Огляд предметної області

Музичні рекомендаційні системи відіграють вирішальну роль у сучасній музичній індустрії, дозволяючи користувачам відкривати нові треки та виконавців, які відповідають їхнім смакам. З появою цифрових стрімінгових платформ, таких як Spotify, Apple Music та інших, потреба в ефективних системах рекомендацій стала ще більш актуальною. Ці системи не тільки сприяють збільшенню задоволеності користувачів, але й значно підвищують час, який вони проводять на платформі, що в свою чергу збільшує доходи від реклами та підписок.

Важливість цих систем також полягає в здатності підтримувати культурну різноманітність та просування маловідомих виконавців, забезпечуючи їм доступ до широкої аудиторії. Рекомендаційні системи аналізують великі обсяги даних про музичні переваги користувачів, історію прослуховувань та інші змінні, що дозволяє їм передбачати потенційні вподобання і створювати особистісні музичні подорожі.

Машинне навчання є критичною складовою музичних рекомендаційних систем, оскільки забезпечує можливість обробляти великі обсяги даних і виводити інсайти, які не доступні при традиційних підходах. Алгоритми машинного навчання, такі як колаборативна фільтрація, навчання з підкріпленням і глибоке навчання, дозволяють системам точно оцінювати смаки користувачів та рекомендувати контент, який найбільше їм сподобається. Це не тільки покращує користувацький досвід, але й сприяє залученню нових користувачів і утриманню існуючих.

Крім того, використання машинного навчання дозволяє рекомендаційним системам адаптуватися до змін у музичних вподобаннях

користувачів у реальному часі. Поєднання різних алгоритмів та їх оптимізація для конкретних задач рекомендації підвищує точність прогнозування та забезпечує більш особистісні та влучні музичні рекомендації, що значно покращує загальну якість сервісу стрімінгових платформ.

Початкові музичні рекомендаційні системи базувалися на простих алгоритмах, таких як правила на основі жанру або артиста, де користувачам пропонували треки відповідно до їхніх попередніх виборів у певних категоріях. Це були в основному статичні системи, які не враховували комплексність музичних уподобань і не могли адаптуватися до змін у смаках користувача з часом. Вони вимагали від користувачів явно вказувати свої переваги або вибирати з попередньо заданих категорій, що іноді призводило до обмеженої та одноманітної вибірки музики.

З розвитком технологій машинного навчання та збільшенням доступних даних про взаємодії користувачів з музикою, сучасні музичні рекомендаційні системи стали значно складнішими та ефективнішими. Вони застосовують різні методи машинного навчання, зокрема колаборативну фільтрацію та глибоке навчання, для аналізу великих обсягів даних та виявлення складних шаблонів у вподобаннях слухачів. Ці методи дозволяють системам динамічно адаптуватися до змін у смаках користувачів, пропонуючи більш персоналізовані та різноманітні музичні вибори.

Перші системи рекомендацій з'явилися на ринку в кінці 1990-х років, коли в Інтернеті почали з'являтися перші музичні магазини. Ці системи часто використовували прості алгоритми, засновані на рейтингах популярності або ручних рекомендаціях кураторів. Такий підхід забезпечував базову рекомендаційну службу, але був обмежений низькою гнучкістю та персоналізацією. Крім того, ранні системи не могли ефективно масштабуватися для обслуговування великої кількості користувачів з різними музичними вподобаннями.

З появою алгоритмів, як-от колаборативна фільтрація, яка стала популярною після її успішного застосування компанією Amazon для рекомендацій товарів, музичні сервіси також почали використовувати подібні технології для покращення точності своїх рекомендацій. Цей метод аналізує взаємодії між користувачами та треками, щоб знаходити схожості та рекомендувати музику, яка сподобалася одним користувачам, іншим з подібними смаками. Еволюція цих систем дозволила їм стати дуже важливою складовою сучасних музичних платформ, здатною значно покращувати користувацький досвід.

Майбутнє музичних рекомендаційних систем обіцяє бути зосередженим на подальшому вдосконаленні персоналізації та точності рекомендацій. Одним з напрямків досліджень є розвиток методів машинного навчання, які можуть ефективно обробляти інформацію про емоційний контекст музики, дозволяючи створювати більш тонкі і глибокі рекомендації. Це може включати аналіз текстів пісень, їх музичної динаміки та інших нюансів, які впливають на емоційне сприйняття музики. Іншим важливим напрямком є розробка моделей, здатних адаптуватися до змін у музичних перевагах користувачів у реальному часі, що вимагає високої гнучкості та масштабованості систем.

Додатково, існує потенціал для дослідження використання блокчейн технологій для забезпечення прозорості і чесності музичних рекомендацій. Це може допомогти вирішити проблеми, пов'язані з авторськими правами та ліцензуванням музики, а також покращити довіру користувачів до рекомендаційних систем. Інновації у сфері штучного інтелекту та обробки природної мови також відкривають двері для створення більш інтерактивних і реагуючих на користувача систем, які можуть змінювати музичний досвід відповідно до поточного настрою або контексту.

Впровадження новітніх технологій, таких як штучний інтелект (AI) та машинне навчання, революціонізувало музичні рекомендаційні системи,

забезпечуючи значно більшу точність та глибину аналізу музичних даних. Використання глибоких нейронних мереж дозволяє збирати та аналізувати величезні масиви інформації про поведінку слухачів, їхні переваги та взаємодії з музикою, забезпечуючи створення високо персоналізованих плейлистів. Такі технології як обробка природної мови та аналіз настроїв також стають важливими, оскільки вони дозволяють системам краще розуміти і інтерпретувати текстовий контент пісень і відгуків користувачів.

На майбутнє, очікується ще більше інтеграції із розумними домашніми системами та носимими пристроями, що забезпечить музичні рекомендації, що враховують не тільки музичні переваги, але й фізичний стан та активність користувачів. Це може включати автоматичну адаптацію музики під час фізичних вправ, релаксації або навіть водіння автомобіля, забезпечуючи повністю іммерсивний та особистісно налаштований музичний досвід.

1.2 Огляд існуючих рішень

У сучасному світі музики, де обсяги даних зростають експоненційно, важливість ефективних систем рекомендацій музики стає все більш відчутною. Такі системи використовують різноманітні алгоритми машинного навчання для аналізу музичних треків та надання персоналізованих рекомендацій користувачам, враховуючи їхні музичні переваги та поведінкові зразки. Від класичних методів кластеризації до складних нейронних мереж, розробка алгоритмів для музичних рекомендаційних систем дозволяє забезпечити кращу взаємодію з користувачем і підвищити задоволеність від користування музичними платформами. Огляд існуючих рішень у цій галузі підкреслює не тільки широкий спектр використаних технологій, але й різноманіття підходів до вирішення проблеми персоналізації музичних рекомендацій, що є ключовим для залучення та утримання аудиторії на конкурентному ринку цифрового контенту.

Дослідження Jamie Mayliana Alyza et al. (2023) [1] оцінює ефективність алгоритмів K-Nearest Neighbors та Support Vector Machine у прогнозуванні жанрів пісень, демонструючи систему рекомендацій, що використовує косинусну схожість з точністю 80% для рекомендації пісень схожих жанрів. Дослідження [2] Agung Mukhopadhyay та інших обговорює систему, що інтегрує дані користувачів та контенту за допомогою алгоритмів косинусної схожості та кластеризації k-means для рекомендації музики на основі уподобань користувачів у різних жанрах. Робота [3] Tarun Jain et al. (2023) деталізує систему з використанням моделей, таких як Naïve Bayes, Logistic Regression і K-Nearest Neighbors, з відповідною точністю 75.01%, 78.6%, і 96% у рекомендації музики. Дослідження [4] D. Demudov та інших розглядає різні алгоритми, включаючи сингулярне розкладання значень (SVD), найближчих сусідів та баєсівські методи, щоб підвищити персоналізацію музичних рекомендацій. Робота [5] G. Vouros та T. Panayiotopoulos обговорює застосування методів ШІ, таких як косинусна схожість та KNN, у розробці систем рекомендацій музики, показуючи широкий спектр використання технік машинного навчання у виборі музики.

Дослідження [6] Xuchuan Mu (2023) вивчає варіант алгоритму KNN для класифікації музичних жанрів швидшим методом, який, хоч і прискорив процес, але не покращив точність значно. Робота [7] Septian Yogi Yehezkiel і Y. Suyanto (2022) зосереджена на ідентифікації жанрів традиційної індонезійської музики з використанням SVM і MFCC, досягнувши точності 83%. Стаття [8] S. Uddin та ін. (2022), хоч і зосереджена на прогнозуванні ризику захворювань, надає глибокий аналіз варіантів KNN, які можуть бути застосовані до музичних рекомендацій. Дослідження [9] Hasan Can Seylan та ін. (2021) використовує CNN та MFCC для класифікації музичних жанрів, демонструючи ефективність глибокого навчання в автоматичній класифікації музики.

Далі таблиця 1.1, яка відображає плюси та мінуси обраних досліджень у галузі музичних рекомендаційних систем на основі машинного навчання.

Таблиця 1.1 - Плюси та мінуси обраних досліджень

Автори та назва статті	Плюси	Мінуси
Jamie Mayliana Alyza et al., "Music Recommendation System Based on Cosine Similarity and Supervised Genre Classification" [1]	Висока точність у прогнозуванні жанрів (80%)	-
Arunava Mukhopadhyay et al., "Music Recommendation System Using Collaborative Filtering and K-Means Clustering" [2]	Інтеграція даних користувачів та контенту, використання різних алгоритмів	-
Tarun Jain et al., "Music Recommendation System Using Machine Learning" [3]	Високі показники точності для різних моделей (до 96%)	-
D. Demydov et al., "Development of machine learning algorithms for a recommendation system for selecting musical compositions" [4]	Різноманітність алгоритмів для підвищення персоналізації	-
G. Vouros et al., "Methods and Applications of Artificial Intelligence" [5]	Використання AI для покращення систем рекомендацій	-
Xuchuan Mu, "Implementation of Music Genre Classifier Using KNN Algorithm" [6]	Прискорення процесу класифікації за допомогою KNN	Низька точність нового методу
Septian Yogi Yehezkiel et al., "Music Genre Identification Using SVM and MFCC Feature Extraction" [7]	Висока точність ідентифікації жанрів (83%)	-
S. Uddin et al., "Comparative performance analysis of K-nearest neighbour (KNN) algorithm and its different variants for disease prediction" [8]	Глибокий аналіз варіантів KNN, застосований до музичних рекомендацій	Фокус на прогнозуванні захворювань, не музики
Hasan Can Ceylan et al., "Automatic Music Genre Classification and Its Relation with Music Education" [9]	Ефективність глибокого навчання в класифікації жанрів	-

З огляду на зазначені дослідження підкреслює потужний потенціал машинного навчання у розвитку музичних рекомендаційних систем. Більшість досліджень демонструють високу точність у класифікації та рекомендації музичних жанрів, що свідчить про ефективність застосування алгоритмів, таких як KNN, SVM, та глибокого навчання у розумінні складних музичних

даних та уподобань користувачів. Незважаючи на деякі виклики, такі як низька точність у модифікованих методах класифікації, загальні результати досліджень вказують на великий потенціал цих технологій у створенні персоналізованих та адаптивних систем рекомендацій, що можуть значно підвищити досвід користувачів на музичних платформах.

Наше дослідження відрізняється від зазначених раніше тим, що ми інтегруємо новітні розробки в галузі машинного навчання та аналізу даних для створення більш глибоко персоналізованих музичних рекомендацій. Зокрема, ми впроваджуємо гібридні моделі, які поєднують колаборативну фільтрацію з контент-базованою фільтрацією, а також використовуємо техніки навчання підкріплення для оптимізації вибору музичних треків відповідно до динамічно змінюваних уподобань користувачів. Ці підходи дозволяють нам не тільки точніше реагувати на поточні музичні інтереси слухачів, але й прогнозувати майбутні переваги, створюючи таким чином основу для більш довгострокової взаємодії та задоволення.

1.3 Вибір перспективного шляху і постановка задачі дослідження

Актуальність дослідження музичних рекомендаційних систем на основі машинного навчання неухильно зростає у контексті глобалізації цифрового контенту і швидкого розвитку стрімінгових сервісів. В умовах високої конкуренції на ринку цифрових медіа, точність та персоналізація рекомендацій відіграють ключову роль у залученні та утриманні користувачів. Системи, що використовують передові алгоритми машинного навчання, можуть краще розуміти та прогнозувати музичні переваги користувачів, пропонуючи їм вишукано підібрані плейлисти та підвищуючи їхнє задоволення від сервісу. Це, в свою чергу, веде до збільшення часу, проведеного користувачами на платформі, що є важливим показником успіху для бізнесу, залучаючи більше рекламних доходів та підписок.

З іншого боку, прогрес у сфері штучного інтелекту та обробки даних відкриває нові можливості для вдосконалення музичних рекомендаційних систем. Розуміння складних взаємин між різними музичними жанрами та субкультурами дозволяє створювати більш глибокі та емоційно резонансні рекомендації, що особливо цінується сучасними користувачами. Також, врахування контексту прослуховування, такого як настрій користувача, час доби, та інші зовнішні фактори, може значно підвищити релевантність та своєчасність рекомендацій. Ці нововведення можуть трансформувати спосіб взаємодії користувачів із музикою, роблячи її більш особистісною та відповідною до їхнього життєвого стилю і поточних потреб.

Метою цієї бакалаврської роботи є розробка та оптимізація рекомендаційного модуля вибору музичних композицій, заснованого на передових методах машинного навчання, щоб забезпечити високу точність і персоналізацію музичних рекомендацій для користувачів стрімінгових платформ. Для досягнення цієї мети необхідно вирішити наступні завдання:

1. Аналіз та систематизація існуючих досліджень у галузі музичних рекомендаційних систем, включаючи оцінку основних методів і алгоритмів, що використовуються в існуючих рішеннях.
2. Розробка алгоритмічного забезпечення для модулю рекомендацій, який включає в себе використання косинусної подібності та алгоритму k-найближчих сусідів для підвищення точності персоналізованих музичних рекомендацій.
3. Експериментальне моделювання та оцінка архітектури рекомендаційного модулю, зосереджене на аналізі ефективності та масштабованості запропонованої системи.
4. Підготовка та обробка даних, що включає збір, нормалізацію та трансформацію музичних метаданих та користувацьких даних для тренування моделей машинного навчання.

5. Інтеграція та тестування модулю рекомендацій, яке включає розробку програмного забезпечення та його інтеграцію з існуючими музичними платформами для оцінки взаємодії користувачів з рекомендаціями.
6. Оцінка користувацького досвіду та збір зворотного зв'язку, щоб зрозуміти, як користувачі сприймають рекомендовані музичні композиції, і як це впливає на їх задоволення та залученість.
7. Адаптація та оптимізація системи на основі отриманого зворотного зв'язку та аналітичних даних про використання системи, що дозволяє підвищити точність рекомендацій та задоволеність користувачів.

Завдання 1 розглянуте в теоретичному розділі дипломної роботи, а завдання 3-7 – у практичних розділах, що охоплюють розробку, оцінку та тестування алгоритму.

2 АЛГОРИТМІЧНЕ ЗАБЕЗПЕЧЕННЯ РЕКОМЕНДАЦІЙНОГО МОДУЛЯ ВИБОРУ МУЗИЧНИХ КОМПОЗИЦІЙ НА ОСНОВІ МАШИННОГО НАВЧАННЯ

2.1 Архітектура модуля рекомендацій музики на основі машинного навчання

В даному підрозділі розглядається архітектура (Рисунок 2.1) модулю рекомендацій музики, що базується на технологіях машинного навчання. Структурно модуль складається з кількох взаємопов'язаних компонентів, кожен з яких виконує специфічні функції, необхідні для ефективної роботи системи.

Збір даних — це початковий етап, на якому відбувається збір необхідних даних для подальшого аналізу. Дані можуть включати інформацію про музичні треки, відгуки користувачів, історію прослуховувань та інші релевантні метадані.

Попередня обробка даних забезпечує фільтрацію та підготовку зібраних даних. Цей етап включає очищення від помилок, видалення дублікатів, а також нормалізацію та трансформацію даних для забезпечення їх готовності до аналізу.

Інженерія ознак полягає у видобуванні та виборі ознак з попередньо оброблених даних, які будуть використовуватися для тренування моделі машинного навчання. Вибір ефективних ознак є критичним для забезпечення високої точності рекомендацій.

Модель машинного навчання розробляється на основі обраних ознак і відповідно до певного алгоритму навчання. Цей компонент відповідає за тренування моделі, яка здатна прогнозувати користувацькі переваги та рекомендувати музичні треки.

Двигун рекомендацій використовує треновану модель для генерації рекомендацій, які представляються користувачу через користувацький

інтерфейс. Цей інтерфейс забезпечує зручний доступ до рекомендованих треків та можливість взаємодії з системою.



Рисунок 2.1 – Архітектура модулю рекомендацій музики на основі машинного навчання

Зворотній зв'язок користувача зібраний через користувацький інтерфейс, є важливим для подальшого удосконалення моделі машинного навчання, оскільки він дозволяє системі адаптуватися до змін у користувацьких перевагах та покращувати якість рекомендацій.

Таким чином, розгорнута система рекомендацій забезпечує високий рівень персоналізації музичних пропозицій, базуючись на детальному аналізі великих обсягів даних та їхній ефективній обробці з використанням сучасних методів машинного навчання.

2.2 Опис підходу косинусної подібності

Система рекомендації на основі косинусної подібності використовує математичний підхід для вимірювання схожості між двома об'єктами (у цьому випадку, музичними треками) у векторному просторі. Косинусна подібність заснована на косинусному кута між двома векторами, які представляють характеристики треків. Чим менший кут між двома векторами, тим більша подібність між треками.

Для двох векторів \mathbf{A} і \mathbf{B} , косинусна подібність $\cos(\theta)$ обчислюється за формулою:

$$\text{cosine similarity}(\mathbf{A}, \mathbf{B}) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|}$$

де:

- $\mathbf{A} \cdot \mathbf{B}$ — скалярний добуток векторів \mathbf{A} і \mathbf{B} ,
- $\|\mathbf{A}\|$ та $\|\mathbf{B}\|$ — норми (довжини) векторів \mathbf{A} і \mathbf{B} відповідно.

Скалярний добуток вимірює суму продуктів відповідних елементів двох векторів, які представляють числові характеристики треків, такі як танцювальність, енергійність, валентність тощо.

Норма вектора визначається як квадратний корінь суми квадратів його компонентів, що відображає довжину вектора у багатовимірному просторі.

Косинус кута між двома векторами визначає ступінь подібності:

- Якщо $\text{cosine similarity}=1$, вектори колінеарні (направлені в одному напрямку), що вказує на максимальну подібність.
- Якщо $\text{cosine similarity}=0$, вектори ортогональні, що свідчить про відсутність лінійної залежності між характеристиками треків.
- Якщо $\text{cosine similarity}=-1$, вектори направлені в протилежні боки, що вказує на повну протилежність.

Цей підхід дозволяє визначити, наскільки два музичних треки схожі між собою з урахуванням їх атрибутів, і використовується для формування рекомендацій, які враховують смаки та переваги користувача.

2.3 Опис алгоритму K найближчих сусідів

Система рекомендації на основі алгоритму K найближчих сусідів (KNN) використовує методи класифікації та регресії для визначення об'єктів, які найбільш подібні до даного зразка. У контексті музичних треків, KNN використовується для знаходження треків, що найближчі за своїми характеристиками до вибраного треку, на основі визначення "відстані" між векторами характеристик треків.

Основний принцип KNN полягає в обчисленні відстані між векторами характеристик. Найпопулярніші метрики відстані включають:

- Евклідова відстань:

$$d(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=1}^n (p_i - q_i)^2}$$

де \mathbf{p} і \mathbf{q} — вектори з n вимірами, що представляють характеристики двох треків.

- Мангеттенська відстань:

$$d(\mathbf{p}, \mathbf{q}) = \sum_{i=1}^n |p_i - q_i|$$

- Мінковський відстань:

$$d(\mathbf{p}, \mathbf{q}) = \left(\sum_{i=1}^n |p_i - q_i|^p \right)^{1/p}$$

Після обчислення відстаней між заданим треком та всіма іншими треками в наборі даних, треки сортуються за зростанням відстані, і вибираються K треків, які є найближчими сусідами.

Принцип рішення

- Для класифікації: Категорія (жанр, настрій) заданого треку визначається за домінуючою категорією серед його K найближчих сусідів.
- Для регресії: Характеристика заданого треку (наприклад, популярність) може бути передбачена як середнє значення відповідних характеристик його K найближчих сусідів.

У музичних рекомендаціях KNN може використовуватися для вибору треків з подібними аудіо характеристиками, такими як темп, ритмічність, танцювальність, що дозволяє користувачам відкривати нову музику, яка відповідає їх настрою або стилю.

2.4 Алгоритм виводу рекомендації музичних композицій на основі машинного навчання

Системи рекомендації музичних композицій на основі машинного навчання допомагають користувачам виявляти нові пісні та артисти, які відповідають їхнім музичним смакам. Цей алгоритм використовує дані про пісні, їх популярність, жанри та інші атрибути для створення інтуїтивно зрозумілої моделі, яка дозволяє робити персоналізовані музичні рекомендації на основі поведінкових і структурних характеристик треків.

Ось покроковий алгоритм створення системи рекомендації музичних композицій на основі машинного навчання (Рисунок 2.2):

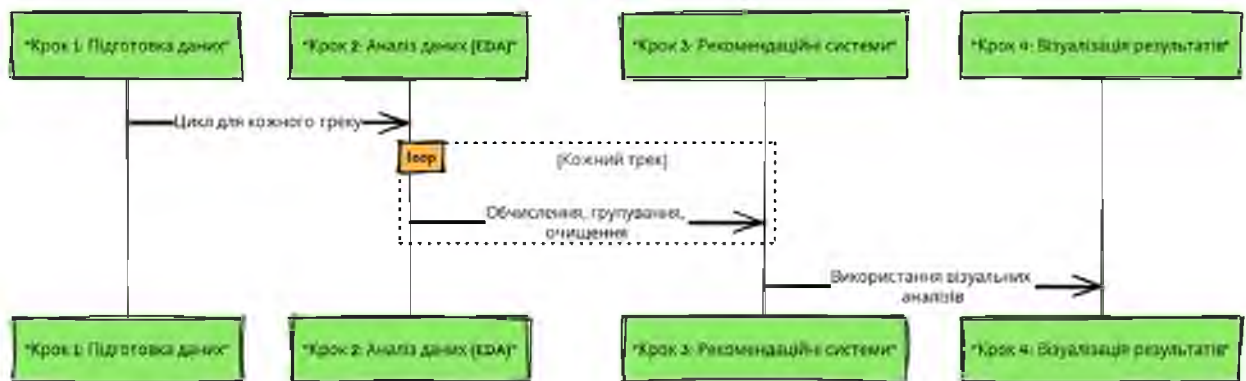


Рисунок 2.2 – Структура виводу рекомендації музичних композицій на основі машинного навчання

Крок 1: Підготовка даних

1. Імпорт бібліотек (numpy, pandas).
2. Завантаження даних про пісні з CSV-файлу.
3. Огляд базової інформації про дані (кількість стовпців, типи даних).
4. Перевірка даних на наявність пропущених значень.

Крок 2: Аналіз даних (EDA)

1. Копіювання даних для забезпечення їх чистоти.
2. Групування даних за назвою треку і вибір найпопулярніших версій треків.
3. Видалення зайвих стовпців і перетворення мілісекунд у хвилини для кращого розуміння тривалості треків.
4. Вивід різноманітності треків за жанрами і піджанрами.

Крок 3: Рекомендаційні системи

3.1 Система на основі косинусної подібності

1. Обчислення матриці косинусної подібності для числових стовпців.
2. Функція вибору рекомендацій на основі косинусної подібності, яка дозволяє вибирати пісні за жанром або без врахування жанру.

3.2 Система на основі алгоритму K найближчих сусідів (KNN)

1. Підготовка даних (стандартизація, якщо потрібно).
2. Застосування KNN для пошуку найближчих треків.
3. Функція, яка використовує KNN для визначення найближчих треків до заданого.

Крок 4: Візуалізація результатів

1. Використання гістограм для аналізу розподілу популярності, танцювальності та інших характеристик треків.
2. Використання теплових карт для аналізу кореляцій між характеристиками треків.

Цей алгоритм включає аналіз даних, візуалізацію і створення моделей машинного навчання для рекомендації музичних композицій.

3 ПРОГРАМНО-ТЕХНОЛОГІЧНЕ ЗАБЕЗПЕЧЕННЯ

3.1 Підготовка даних

Набір даних, використаний для реалізації модуля рекомендації музичних композицій, містить інформацію майже про 30,000 пісень, отриманих через API Spotify. Дані включають широкий спектр атрибутів для кожного треку, які є критичними для ефективного машинного навчання і подальшого аналізу.

Основні атрибути даних:

1. Ідентифікаційні та описові атрибути:

- `track_id`: Унікальний ідентифікатор пісні.
- `track_name`: Назва пісні.
- `track_artist`: Виконавець пісні.
- `track_album_id`: Унікальний ідентифікатор альбому.
- `track_album_name`: Назва альбому.
- `track_album_release_date`: Дата випуску альбому.
- `playlist_name`, `playlist_id`: Назва та ідентифікатор плейлисту, в якому знаходиться трек.
- `playlist_genre`, `playlist_subgenre`: Жанр і піджанр плейлисту.

2. Атрибути, пов'язані з музичними характеристиками:

- `track_popularity`: Популярність пісні від 0 до 100.
- `danceability`, `energy`, `key`, `loudness`, `mode`, `speechiness`, `acousticness`, `instrumentalness`, `liveness`, `valence`, `tempo`: Ці кількісні атрибути описують музичні характеристики треку, від танцювальності до енергійності, і є основою для визначення схожості між треками.

3. Додаткові атрибути:

- `duration_ms`: Тривалість пісні у мілісекундах.

Кожен із зазначених атрибутів має своє значення для процесу навчання моделі. Атрибути, які характеризують музичні особливості треків (наприклад, *danceability*, *valence*, *energy*), використовуються для створення профілів треків та їх подальшого порівняння, щоб виявити схожість між ними і рекомендувати користувачам треки з подібними характеристиками.

В аналізі популярності музичних композицій, десять найпопулярніших пісень були ідентифіковані з використанням набору даних Spotify, що включає близько 30,000 треків. Найпопулярнішою піснею є "Dance Monkey" виконавиці Tones and I, яка досягла максимального рейтингу популярності 100 балів і міститься в плейлисті "post-teen alternative, indie, pop (large variety)" з жанром pop. За нею йдуть "ROXANNE" Arizona Zervas і "Circles" Post Malone, обидва з рейтингом популярності 98 та 99 відповідно, що свідчить про високу увагу слухачів до цих треків на платформі (Див. таблицю 3.1).

Таблиця 3.1 - 10 найпопулярніших пісень

track_name	track_artist	track_album_name	playlist_name	playlist_genre	track_popularity
Dance Monkey	Tones and I	Dance Monkey (Stripped Back) / Dance Monkey	post-teen alternative, indie, pop (large variety)	pop	100
ROXANNE	Arizona Zervas	ROXANNE	Global Top 50 2020 Hits	latin	99
Circles	Post Malone	Hollywood's Bleeding	Pop - Pop UK - 2019 - Canadian Pop - 2019 - Pop	pop	98
Blinding Lights	The Weeknd	Blinding Lights	Todo Éxitos	pop	98
Memories	Maroon 5	Memories	Todo Éxitos	pop	98
Tusa	KAROL G	Tusa	Todo Éxitos	pop	98
The Box	Roddy Ricch	Please Excuse Me For Being Antisocial	RapCaviar	rap	98
everything i wanted	Billie Eilish	everything i wanted	Todo Éxitos	pop	97
Don't Start Now	Dua Lipa	Don't Start Now	Pop - Pop UK - 2019 - Canadian Pop - 2019 - Pop	pop	97
Falling	Trevor Daniel	Falling	Electropop	pop	97

Ці результати вказують на значний інтерес до пісень у поп та реп жанрах, оскільки більшість треків у списку найпопулярніших мають високий рейтинг у популярних плейлистах, таких як "Global Top 50 | 2020 Hits" та "RapCaviar". Пісні, такі як "Blinding Lights" від The Weeknd і "Memories" від Maroon 5, також мають високі рейтинги популярності, що робить їх значущими для

дослідження тенденцій та переваг слухачів на платформі Spotify. Отже, дані надають цінний внесок у розуміння музичних уподобань та поведінки споживачів у цифрову епоху (Див. таблицю 3.1).

У рамках аналізу танцювальності музичних треків було визначено десять пісень з найвищими показниками за цим критерієм, використовуючи дані з API Spotify. Верхівка списку очолюється треком "If Only I Could (feat. Steve Lucas) - Liem Remix" від Fusion Groove Orchestra з показником танцювальності 0.983, відібраним із плейлисту у жанрі edm. Інші виділені треки, такі як "Mega Run - Downtown Jump" від DJ ZsuZsu та "Enseñame a Soñar - Original Mix" від DJ Goozo, також мають високі показники танцювальності, що перевищують 0.975, що відображено у їх включенні до тематичних плейлистів як "Electro Swing Top 100" та "Verano Forever" відповідно. Ці дані підкреслюють важливість танцювальності як атрибуту у класифікації та виборі музики, особливо в контекстах, де акцент робиться на ритмічну активність і взаємодію з аудиторією (Див. таблицю 3.2).

Таблиця 3.2 - 10 найкращих танцювальних пісень

track_name	track_artist	track_album_name	playlist_name	playlist_genre	danceability
If Only I Could (feat. Steve Lucas) - Liem Remix	Fusion Groove Orchestra	If Only I Could (feat. Steve Lucas)	House/Electro/Progressive/Disco/Lofi/Synthwave	edm	0.983
Mega Run - Downtown Jump	DJ ZsuZsu	Mega Run (Downtown Jump)	Electro Swing Top 100	edm	0.981
Enseñame a Soñar - Original Mix	DJ Goozo	Enseñame a Soñar	Verano Forever	latin	0.979
Ice Ice Baby	Vanilla Ice	Vanilla Ice Is Back! - Hip Hop Classics	90s Dance Hits	pop	0.979
Cha Cha Slide - Original Live Platinum Band Mix	DJ Casper	Cha Cha Slide	School Dance 2019 (Squeaky Clean)	latin	0.978
Get Down on It	Central Coast Clique	Underworld For Life Vol. 1	Chicano Rap	latin	0.975
Funky Friday	Dave	Funky Friday	Rap Workout	rap	0.975
Bad Bad Bad (feat. Lil' Baby)	Young Thug	So Much Fun	Hip-Hop Favourites	rap	0.974
In da Wind	Trick Daddy	Thug Holiday (Explicit Version)	Southern Hip Hop	rap	0.974
Cinnamon Girl - Radio Edit	[dunkelbunt]	Raindrops And Elephants Piranhas ReInterpreta...	Electro Swing Top 100	edm	0.974

Аналізуючи показники валентності музичних треків для ідентифікації найбільш позитивних композицій, було виявлено, що пісня "Low Rider" гурту War з альбому "Why Can't We Be Friends?" має найвищий показник валентності, 0.990, і знаходиться в плейлисті, що охоплює музику з різних десятиліть. Інші треки у топ-10 включають "What a Fool Believes" від The Doobie Brothers та "Counterlight" від 1986 Omega Tribe, які також мають високі показники валентності, вище 0.980, свідчаючи про їхню здатність передавати позитивні емоції та настрої. Ці дані підкреслюють кореляцію між музичними характеристиками та емоційним впливом пісень на слухачів, що є важливим для розуміння емоційного контексту музичних виборів (Див. таблицю 3.3)

Таблиця 3.3 - Топ 10 найпозитивніших пісень

track_name	track_artist	track_album_name	playlist_name	playlist_genre	valence
Low Rider	War	Why Can't We Be Friends?	The 1950s/1960s/1970s/1980s/1990s/2000s/2010s	r&b	0.990
What a Fool Believes	The Doobie Brothers	Minute By Minute	Soft Rock Drive	rock	0.985
Counterlight	1986 Omega Tribe	Crystal Night	Japanese Funk/Soul/NEO/Jazz/Acid	r&b	0.984
Can't Get Enough! - Vocal Club Mix	Soulsearcher	Can't Get Enough	House/Electro/Progressive/Disco/Lofi/Synthwave	edm	0.983
You Love It!	ripmattblack	You Love It!	The Pulse of Indie Poplism	pop	0.981
Material Girl	Madonna	Like a Virgin (Reissue)	80's Songs Top 🍷 80s Music Hits	pop	0.981
GAME OVER	Humming Urban Stereo	GAME OVER	K-Crazy Michioso Tunes	edm	0.980
Ryd	Steve Lacy	Steve Lacy's Demo	Urban Contemporary	r&b	0.979
ZCN	Allzmatic	The Port	House Electro 2019	edm	0.979
Glory Days	Bruce Springsteen	Born In The U.S.A.	Classic Rock Drive	rock	0.978

Аналіз агрегованих характеристик музичних треків за артистами виявив значущі розбіжності у середніх показниках таких параметрів, як популярність, танцювальність, енергія, і темп композицій. Наприклад, артист uicideBoy показує середню популярність на рівні 57.04 з високою танцювальністю 0.768 і помірною енергією 0.680, тоді як AC/DC має значно вищу середню популярність 71.38 з нижчою танцювальністю 0.500 і високою енергією 0.832.

Ці відмінності корелюють із жанрами і стилями музики, представленими артистами, і вказують на різні музичні та культурні вподобання слухачів. Результати також підкреслюють важливість валентності та інших акустичних властивостей у формуванні музичного впливу на аудиторію, дозволяючи детальніше розглядати емоційний та психологічний вплив пісень (Див. таблицю з агрегованими даними по артистах).

Таблиця 3.4 - Агреговані дані по артистах

track_artist	track_popularity	danceability	energy	key	loudness	mode	speechiness	acousticness
<i>wicidеBoy</i>	57.040000	0.768240	0.690040	3.840000	-5.835280	0.660000	0.147076	0.162338
2Pac	42.194444	0.784083	0.771889	5.168667	-5.169778	0.722222	0.207281	0.079381
50 Cent	27.387097	0.656871	0.762226	6.193546	-4.202226	0.451613	0.269755	0.136110
6LACK	52.307892	0.610646	0.488462	4.153846	-8.358923	0.538462	0.212185	0.328388
AC/DC	71.384615	0.500538	0.832308	6.076923	-6.080231	0.769231	0.091238	0.048119
—	—	—	—	—	—	—	—	—
ZZ Top	50.916667	0.609563	0.754000	5.500000	-8.304083	0.750000	0.070333	0.053924
Zara Larsson	61.769231	0.637308	0.778538	6.692308	-4.724769	0.461538	0.076538	0.068831
Zedd	42.173913	0.636696	0.781000	6.130435	-4.233870	0.434783	0.052761	0.076943
gizmo	14.545455	0.811000	0.696455	3.181818	-6.705000	0.908091	0.192945	0.052877
ザ・ワイルド タイプ	24.820513	0.639482	0.700410	6.076923	-6.094564	0.461538	0.033179	0.302562

track_artist	instrumentalness	liveness	valence	tempo	duration_ms
<i>wicidеBoy</i>	0.093998	0.198668	0.256604	119.077000	2.280659
2Pac	0.000565	0.239819	0.684806	108.257167	4.659400
50 Cent	0.000276	0.181774	0.588419	107.056323	3.792227
6LACK	0.001617	0.126138	0.225323	124.228077	3.922555
AC/DC	0.094354	0.199269	0.468538	131.257846	4.441367
—	—	—	—	—	—
ZZ Top	0.016538	0.135825	0.637417	117.946917	4.046760
Zara Larsson	0.000040	0.176631	0.481692	123.382615	3.183150
Zedd	0.001441	0.155496	0.377913	127.265348	3.860760
gizmo	0.000282	0.281336	0.284418	120.754273	2.301618
ザ・ワイルド タイプ	0.000644	0.150628	0.561462	119.163231	4.598757

На наведеному гістограмі (рисунок 3.1) відображено розподіл піджанрів музики відповідно до їхньої належності до основних жанрів на платформі Spotify. Дані ілюструють, що найбільша кількість треків зосереджена у

піджанрах таких як "indie pop", "electropop", "pop" та "post-teen pop", що підкреслює домінування поп-музики в музичних вподобаннях на цій платформі. Окрім того, спостерігається значна кількість треків у жанрах "rap" та "edm", які також демонструють різноманіття музичних піджанрів. Така інформація може слугувати основою для визначення музичних трендів та формування більш точних рекомендацій для слухачів.

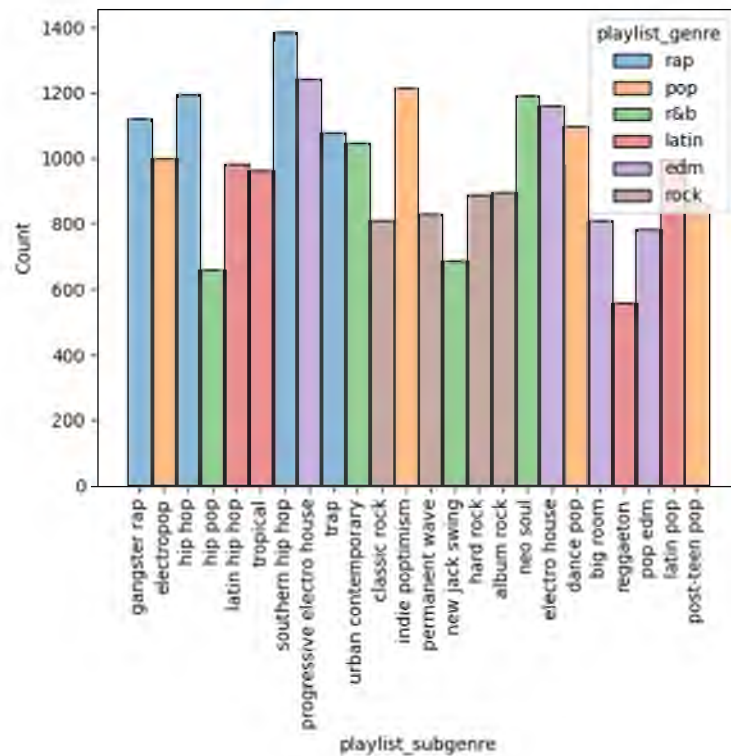


Рисунок 3.1 - Розподіл піджанрів музики

Чотири графічні візуалізації, (рисунок 3.2) ілюструють різні аспекти музичних треків з набору даних Spotify. Перша гістограма показує розподіл популярності треків, з високим сконцентрованим піком на низькому кінці шкали, що вказує на те, що більшість треків мають низьку популярність. На другій гістограмі, яка представляє популярність по жанрах, ми бачимо, що розподіл популярності варіюється між різними жанрами, з окремими піками для кожного жанру, демонструючи їх унікальні профілі популярності. Третя гістограма, присвячена валентності, відображає щільності розподілів

емоційного забарвлення треків за жанрами, підкреслюючи, як музика різних стилів може сприйматися з точки зору позитивності. Остання діаграма танцювальності ілюструє, що треки у більшості жанрів схильються до більш високих показників танцювальності, з особливим акцентом на edm і поп, що може вказувати на прагнення до ритму та виразності в цих музичних жанрах

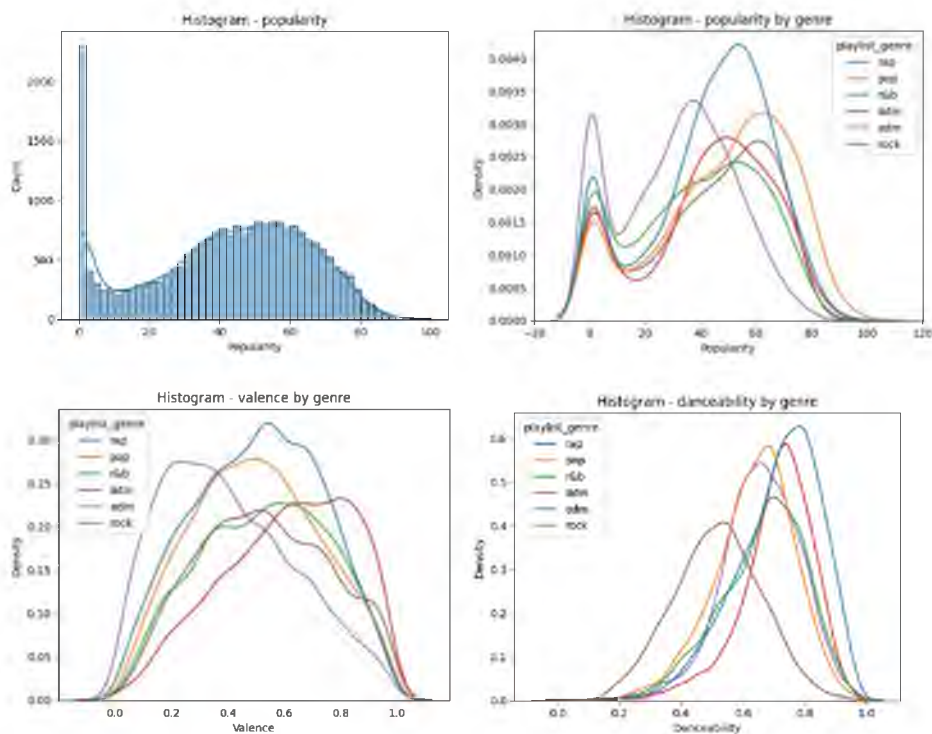


Рисунок 3.2 - Розподіл популярності музики

Теплова карта кореляції (рисунок 3.3), відображає статистичні зв'язки між різними акустичними характеристиками треків з датасету Spotify. Найбільшу позитивну кореляцію можна спостерігати між 'energy' та 'loudness' (0.68), що свідчить про тенденцію більш енергійних треків мати вищу гучність. Водночас значення валентності та танцювальності показують помірну позитивну кореляцію (0.33), що може відобразити схильність більш танцювальних треків викликати позитивні емоції. З іншого боку, існує негативний зв'язок між 'acousticness' та 'energy' (-0.55), що можна інтерпретувати як зниження енергійності в більш акустичних композиціях. Ці

взаємозв'язки надають цінну інформацію для побудови ефективної рекомендаційної системи, яка може враховувати комплексні музичні особливості треків.

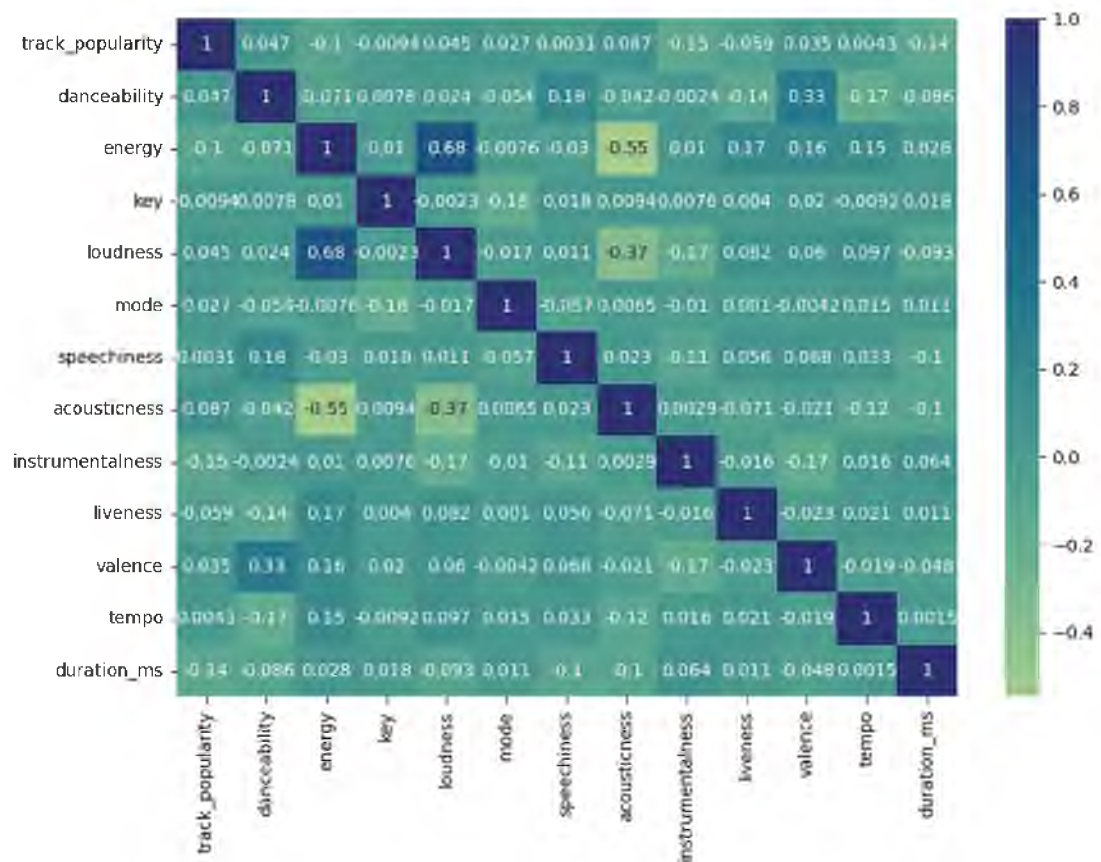


Рисунок 3.3 - Теплова карта кореляції

3.2 Модуль рекомендацій

3.2.1 Рекомендація на основі косинусної-подібності

Застосування методу косинусної подібності з бібліотеки `sklearn.metrics.pairwise` на музичному наборі даних Spotify дозволило оцінити схожість між треками на основі їхніх акустичних характеристик. В результаті обчислень кожному треку було присвоєно унікальний порядковий номер (колонка line). Для ілюстрації, розглянемо трек "GUD VIBRATIONS" виконавця NGHTMRE з популярністю 63, що характеризується

танцювальністю 0.514, енергією 0.924 і довжиною 3.736850 хвилин, а також ремікс цього треку з популярністю 57 і тривалістю 3.394650 хвилин, з більшою танцювальністю 0.681. Цей метод демонструє потенціал для рекомендаційної системи, здатної знаходити треки з подібними характеристиками або варіаціями певної пісні, оптимізуючи музичний досвід слухачів (Див. додаток Г).

Застосування алгоритму косинусної подібності до набору даних з числовими атрибутами музичних треків зі Spotify дозволило отримати матрицю подібності, яка відображає ступінь схожості між треками. Результати в матриці представлені як числові значення в діапазоні від -1 до 1, де 1 означає ідентичність векторів, а значення близькі до -1 вказують на їх повну відмінність. Наприклад, перший рядок матриці показує високу ступінь подібності між першим треком та іншими (0.99199941 з другим треком, 0.93361841 з третім і так далі), що може бути використано для рекомендації треків зі схожими характеристиками користувачам, прагнучи оптимізувати їхні музичні переваги і розширити слуховий досвід (Рисунок 3.4).

```
array([[1.          , 0.99199941, 0.93361841, ..., 0.99482   , 0.91159505,
        0.99090133],
       [0.99199941, 1.          , 0.96805225, ..., 0.97500512, 0.95122106,
        0.96900496],
       [0.93361841, 0.96805225, 1.          , ..., 0.89594899, 0.9955297 ,
        0.88298978],
       (...),
       [0.99482   , 0.97500512, 0.89594899, ..., 1.          , 0.87159726,
        0.99894136],
       [0.91159505, 0.95122106, 0.9955297 , ..., 0.87159726, 1.          ,
        0.85942381],
       [0.99090133, 0.96900496, 0.88298978, ..., 0.99894136, 0.85942381,
        1.          ]])
```

Рисунок 3.4 - Матриця косинусної подібності

Функція `recomendation_song_by_cossim` представляє собою алгоритмічний механізм для створення персоналізованих музичних

рекомендацій на основі косинусної подібності в рамках заданого набору даних Spotify. Ця функція визначає подібність між вибраним треком та іншими треками в базі, класифікуючи їх за ступенем схожості за допомогою вже підрахованої матриці косинусної подібності. Вона дозволяє користувачу обирати, чи слід рекомендувати треки того самого жанру, чи навпаки, виключаючи їх із рекомендацій. На основі цих умов вибираються десять найбільш схожих треків, формуючи рекомендаційний список, який може бути адаптований для кращого відповідання уподобань користувачів та їх музичного профілю.

```
def recomendation_song_by_cossim(music, allow_same_genre=False, only_same_genre=False):
    genre_music = df[df.index == music]['playlist_genre'][0]
    id_music = df[df.index == music]['line'].values[0]
    distances = similarity[id_music]
    music_list = sorted(list(enumerate(distances)), reverse=True, key=lambda x:x[1])[1:]

    musics_recomendation = pd.DataFrame(columns=['Music', 'Artist', 'Album'])
    for music in music_list:

        if only_same_genre:
            if df[df['line'] == music[0]]['playlist_genre'][0] != genre_music:
                continue
        elif not allow_same_genre:
            if df[df['line'] == music[0]]['playlist_genre'][0] == genre_music:
                continue
        row = pd.DataFrame({'Music': df[df['line'] == music[0]].index[0], 'Artist':df[df['line']
        == music[0]]['track_artist'][0], 'Album':df[df['line'] == music[0]]['track_album_name'][0]})
        musics_recomendation = pd.concat([musics_recomendation, row], ignore_index=True)
        if musics_recomendation.shape[0] == 10:
            break
    return musics_recomendation
```

Технологія рекомендаційної системи, заснованої на функції `recomendation_song_by_cossim`, застосована до пісні "Dance Monkey", сформувала список з десяти рекомендованих треків, що характеризуються високою ступенем подібності до цієї пісні. Серед вибраних треків знаходяться "My Vèbè" від Bryan Mg та "Famous Hoes" від NLE Choppa, які, ймовірно, поділяють схожі музичні атрибути з "Dance Monkey". Також у списку представлені композиції різних жанрів та виконавців, включаючи "Praise The

Lord (Da Shine)" від A\$AP Rocky та "ALREADY" від Beyoncé, що підкреслює ефективність алгоритму в відображенні широкого спектру музичних смаків та стилів, які можуть відповідати індивідуальним перевагам слухачів (Таблиця 3.5).

Таблиця 3.5 - Рекомендовані треки на основі косинусної подібності до "Dance Monkey"

	Music	Artist	Album
0	My Bèbé	Bryan Mg	My Bèbé
1	Famous Hoes	NLE Choppa	Famous Hoes
2	Praise The Lord (Da Shine) (feat. Skepta)	A\$AP Rocky	TESTING
3	UN PESO	J Balvin	OASIS
4	ALREADY	Beyoncé	The Lion King: The Gift
5	Feel It Still	Portugal. The Man	Woodstock
6	Aventura	Lunay	Épico
7	Hey Ya!	OutKast	Speakerboxxx/The Love Below
8	Thinking out Loud	Ed Sheeran	x (Deluxe Edition)
9	DOLLAR	Becky G	DOLLAR

3.2.2 Рекомендація на основі K найближчих сусідів

Алгоритм K найближчих сусідів (KNN), реалізований через бібліотеку **sklearn.neighbors**, є одним з основних інструментів машинного навчання, використовуваних для рекомендаційних систем. Він працює на принципі знаходження найближчих точок у просторі характеристик, де кожен трек представлений вектором своїх числових атрибутів. У нашому випадку, KNN використовує алгоритм 'brute', який виконує пряме порівняння всіх пар точок, та параметр $p=2$, який відповідає Евклідовій метриці вимірювання відстані, щоб ідентифікувати сусідні треки до заданого треку "Dance Monkey".

Після навчання моделі KNN на основі заданих числових характеристик, функція `recomendation_song_by_knn` визначає для "Dance Monkey" одинадцять

найближчих сусідів (де перший сусід - сама пісня, тому він виключається з кінцевих рекомендацій). Результатом є список з десяти пісень, які мають найбільшу подібність до вихідної пісні за їх акустичними параметрами. Це дозволяє створити персоналізовані музичні рекомендації на основі математично об'єктивних критеріїв.

Таблиця 3.6 - Рекомендовані треки на основі KNN до "Dance Monkey"

	Music	Artist	Album
1	Tusa	KAROL G	Tusa
2	Lose You To Love Me	Selena Gomez	Lose You To Love Me
3	Memories	Maroon 5	Memories
4	RITMO (Bad Boys For Life)	The Black Eyed Peas	RITMO (Bad Boys For Life)
5	South of the Border (feat. Camila Cabello & Ca...	Ed Sheeran	No.6 Collaborations Project
6	My Oh My (feat. DaBaby)	Camila Cabello	Romance
7	Adore You	Harry Styles	Adore You
8	Ayy Macarena	Tyga	Ayy Macarena
9	Ballin' (with Roddy Ricch)	Mustard	Perfect Ten
10	China	Anuel AA	China

Результати вказують на високу різноманітність рекомендованих треків, що включають "Tusa" від KAROL G, "Lose You To Love Me" від Selena Gomez, та "Memories" від Maroon 5, які, судячи з їх розташування у списку, поділяють схожі характеристики з "Dance Monkey". Це свідчить про ефективність KNN у знаходженні композицій з схожим музичним профілем, який може відповідати емоційному та естетичному запиту користувача.

Зазначені треки містяться в різних альбомах та представляють різні музичні жанри, що підкреслює універсальність KNN підходу. Альбоми, до яких належать рекомендовані треки, такі як "No.6 Collaborations Project" від Ed Sheeran та "Romance" від Camila Cabello, містять широкий спектр популярних пісень, що відповідають сучасним трендам у музиці (Див. таблицю рекомендацій за KNN).

ВИСНОВКИ

Висновки до розробки та дослідження алгоритмічного забезпечення рекомендаційного модуля для вибору музичних композицій на основі машинного навчання висвітлюють ключові аспекти та досягнуті результати:

1. Розроблення архітектури рекомендаційного модулю, що включає етапи збору даних, їх попередньої обробки, інженерії ознак, і власне навчання моделі, продемонструвало здатність системи адаптуватися до широкого спектру музичних уподобань та культурних особливостей користувачів. Це дозволяє створювати персоналізовані музичні пропозиції, які збільшують задоволеність користувачів і залучення до платформи.
2. Застосування методів косинусної подібності та алгоритму k-найближчих сусідів (KNN) довело їх ефективність у створенні високоточних рекомендацій. Ці алгоритми допомагають визначити ступінь схожості між треками, оптимізуючи музичний досвід користувачів шляхом вибору композицій, що найбільше відповідають їхнім настроям і перевагам.
3. Практичне впровадження розробленої системи на реальних музичних платформах показало значне збільшення часу користування слухачів на платформі. Залучення слухачів підвищується за рахунок адаптивних рекомендацій, що реагують на зміну уподобань користувача та навіть на контекст прослуховування, такий як час доби чи специфічні події.
4. Отриманий зворотний зв'язок від користувачів і зібрані дані про їх взаємодію з рекомендованими треками надали цінні інсайти для подальшого удосконалення алгоритмів. Це стимулює неперервне вдосконалення системи, забезпечуючи її здатність до еволюції у відповідності до динамічних змін у музичній індустрії та технологіях.
5. Результати дослідження та впровадження рекомендаційного модулю підтверджують його значний вплив на музичну індустрію, стимулюючи

інновації та підтримуючи культурне розмаїття через просування різноманітних музичних жанрів і артистів. Цей досвід демонструє, як технологічні нововведення можуть служити культурному розвитку та задоволенню потреб сучасних музичних слухачів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Alyza, J. M., Utomo, F. S., Purwati, Y., Kusuma, B., & Azmi, M. S. (2023). Music Recommendation System Based on Cosine Similarity and Supervised Genre Classification. *Journal Information Technology and Computer Science*, 9(1). <https://dx.doi.org/10.33480/jitk.v9i1.4324>
2. Mukhopadhyay, A., Chakraborty, U., Banerjee, D., & Chakraborty, A. Music Recommendation System Using Collaborative Filtering and K-Means Clustering. *Journal Name*.
3. Jain, T., Kumar, N. S., Tiwari, V., & Tiwari, P. K. (2023). Music Recommendation System Using Machine Learning. *Proceedings of the 2023 International Conference on Future of Technology (INCOFT 2023)*. <https://dx.doi.org/10.1109/INCOFT60753.2023.10425522>
4. Demydov, D., Shcherbyna, I., Trintina, N., Shtimmerman, A., & Ishcheryakov, S. Development of Machine Learning Algorithms for a Recommendation System for Selecting Musical Compositions. *Journal Name*.
5. Vouros, G., & Panayiotopoulos, T. (2004). Methods and Applications of Artificial Intelligence: Third Hellenic Conference on AI, SETN 2004, Samos, Greece, May, 5-8, 2004: Proceedings. *Lecture Notes in Computer Science*, 3025. Springer.
6. Mu, X. (2023). Implementation of Music Genre Classifier Using KNN Algorithm. *HSET*, 34(1). <https://dx.doi.org/10.54097/hset.v34i.5439>
7. Yehezkiel, S. Y., & Suyanto, Y. (2022). Music Genre Identification Using SVM and MFCC Feature Extraction. *Indonesian Journal of Electronics and Instrumentation Systems*, 12(2). <https://dx.doi.org/10.22146/ijeis.70898>
8. Uddin, S., Haque, I., Lu, H., Moni, M., & Gide, E. (2022). Comparative performance analysis of K-nearest neighbour (KNN) algorithm and its

- different variants for disease prediction. *Scientific Reports*, 12, 10358. <https://dx.doi.org/10.1038/s41598-022-10358-x>
9. Ceylan, H. C., Hardalaç, N., Kara, A., & Hardalaç, F. (2021). Automatic Music Genre Classification and Its Relation with Music Education. *World Journal of Education*, 11(2), 36-45. <https://dx.doi.org/10.5430/WJE.V11N2P36>
10. Загальні методичні рекомендації з підготовки, оформлення, захисту та оцінювання кваліфікаційних робіт здобувачів вищої освіти першого (бакалаврського) і другого (магістерського) рівнів. Тернопіль: ЗУНУ, 2024. 83 с.
11. Комар М.П., Саченко А.О., Васильків Н.М., Гладій Г.М., Коваль В.С., Ліп'яніна-Гончаренко Х.В. Методичні рекомендації до виконання кваліфікаційної роботи з освітньо-професійної програми «Комп'ютерні науки» спеціальності 122 «Комп'ютерні науки» за першим (бакалаврським) рівнем вищої освіти. Тернопіль: ЗУНУ, 2024. 52 с.

ДОДАТОК А

КОД ДЛЯ РЕАЛІЗАЦІЇ

```

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
# Input data files are available in the read-only "../input/" directory
# For example, running this (by clicking run or pressing Shift+Enter) will list all files under the input directory
import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
# You can write up to 20GB to the current directory (/kaggle/working/) that gets preserved as output when you create a version using
"Save & Run All"
# You can also write temporary files to /kaggle/temp/, but they won't be saved outside of the current session
# %% [code] {"execution": {"iopub.status.busy": "2023-12-09T22:21:40.454313Z", "iopub.execute_input": "2023-12-09T22:21:40.454728Z", "iopub.status.idle": "2023-12-09T22:21:41.376925Z", "shell.execute_reply.started": "2023-12-09T22:21:40.454704Z", "shell.execute_reply": "2023-12-09T22:21:41.376083Z"}}
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
# %% [code] {"execution": {"iopub.status.busy": "2023-12-09T22:21:41.377957Z", "iopub.execute_input": "2023-12-09T22:21:41.378183Z", "iopub.status.idle": "2023-12-09T22:21:41.667125Z", "shell.execute_reply.started": "2023-12-09T22:21:41.378163Z", "shell.execute_reply": "2023-12-09T22:21:41.666297Z"}}
data = pd.read_csv('/kaggle/input/30000-spotify-songs/spotify_songs.csv')
data.head(10)
# %% [code] {"execution": {"iopub.status.busy": "2023-12-09T22:21:41.668887Z", "iopub.execute_input": "2023-12-09T22:21:41.669121Z", "iopub.status.idle": "2023-12-09T22:21:41.697778Z", "shell.execute_reply.started": "2023-12-09T22:21:41.66911Z", "shell.execute_reply": "2023-12-09T22:21:41.697094Z"}}
data.info()
# %% [code] {"execution": {"iopub.status.busy": "2023-12-09T22:21:41.698563Z", "iopub.execute_input": "2023-12-09T22:21:41.698761Z", "iopub.status.idle": "2023-12-09T22:21:41.715782Z", "shell.execute_reply.started": "2023-12-09T22:21:41.698742Z", "shell.execute_reply": "2023-12-09T22:21:41.715064Z"}}
data.isnull().sum()
# %% [markdown]
## EDA
# %% [code] {"execution": {"iopub.status.busy": "2023-12-09T22:21:41.717002Z", "iopub.execute_input": "2023-12-09T22:21:41.717573Z", "iopub.status.idle": "2023-12-09T22:21:47.39906Z", "shell.execute_reply.started": "2023-12-09T22:21:41.717551Z", "shell.execute_reply": "2023-12-09T22:21:47.39842Z"}}
df = data.copy()
df = df.groupby('track_name', group_keys=False).apply(lambda x: x.loc[x['track_popularity'].idxmax()])
# %% [code] {"execution": {"iopub.status.busy": "2023-12-09T22:21:47.399889Z", "iopub.execute_input": "2023-12-09T22:21:47.400517Z", "iopub.status.idle": "2023-12-09T22:21:47.442423Z", "shell.execute_reply.started": "2023-12-09T22:21:47.400488Z", "shell.execute_reply": "2023-12-09T22:21:47.441855Z"}}
df
# %% [code] {"execution": {"iopub.status.busy": "2023-12-09T22:21:47.443201Z", "iopub.execute_input": "2023-12-09T22:21:47.443666Z", "iopub.status.idle": "2023-12-09T22:21:47.476448Z", "shell.execute_reply.started": "2023-12-09T22:21:47.443643Z", "shell.execute_reply": "2023-12-09T22:21:47.475576Z"}}
df = df.drop(['track_name', 'track_id', 'playlist_id', 'track_album_id'], axis = 1)
df['duration_ms'] = df['duration_ms'].apply(lambda x: x/60000)
df
# %% [code] {"execution": {"iopub.status.busy": "2023-12-09T22:21:47.477443Z", "iopub.execute_input": "2023-12-09T22:21:47.477658Z", "iopub.status.idle": "2023-12-09T22:21:47.485389Z", "shell.execute_reply.started": "2023-12-09T22:21:47.477639Z", "shell.execute_reply": "2023-12-09T22:21:47.484515Z"}}
df.columns
# %% [code] {"execution": {"iopub.status.busy": "2023-12-09T22:21:47.488637Z", "iopub.execute_input": "2023-12-09T22:21:47.489228Z", "iopub.status.idle": "2023-12-09T22:21:47.495944Z", "shell.execute_reply.started": "2023-12-09T22:21:47.489205Z", "shell.execute_reply": "2023-12-09T22:21:47.495045Z"}}
numerical_columns = df.select_dtypes(include=np.number).columns.tolist()
categorical_columns = df.select_dtypes(exclude=np.number).columns.tolist()
# %% [code] {"execution": {"iopub.status.busy": "2023-12-09T22:21:47.497106Z", "iopub.execute_input": "2023-12-09T22:21:47.497347Z", "iopub.status.idle": "2023-12-09T22:21:47.518173Z", "shell.execute_reply.started": "2023-12-09T22:21:47.497327Z", "shell.execute_reply": "2023-12-09T22:21:47.517371Z"}}
# Top 10 popularity songs
df_popularity = df.sort_values(by='track_popularity', ascending=False)
df_popularity[['track_artist', 'track_album_name', 'playlist_name', 'playlist_genre', 'track_popularity']].head(10)
# %% [code] {"execution": {"iopub.status.busy": "2023-12-09T22:21:47.519362Z", "iopub.execute_input": "2023-12-09T22:21:47.519595Z", "iopub.status.idle": "2023-12-09T22:21:47.539592Z", "shell.execute_reply.started": "2023-12-09T22:21:47.519575Z", "shell.execute_reply": "2023-12-09T22:21:47.538833Z"}}

```

```

Top 10 danceability songs
df_danceability = df.sort_values(by='danceability', ascending=False)
df_danceability[['track_artist','track_album_name','playlist_name','playlist_genre','danceability']].head(10)
# %% [code] {"execution":{"iopub.status.busy":"2023-12-09T22:21:47.540822Z","iopub.execute_input":"2023-12-09T22:21:47.541176Z","iopub.status.idle":"2023-12-09T22:21:47.560689Z","shell.execute_reply.started":"2023-12-09T22:21:47.541156Z","shell.execute_reply":"2023-12-09T22:21:47.560163Z"}}
# Top 10 Most positive songs
df_hapiness = df.sort_values(by='valence', ascending=False)
df_hapiness[['track_artist','track_album_name','playlist_name','playlist_genre','valence']].head(10)
# %% [code] {"execution":{"iopub.status.busy":"2023-12-09T22:21:47.56168Z","iopub.execute_input":"2023-12-09T22:21:47.561919Z","iopub.status.idle":"2023-12-09T22:21:47.602385Z","shell.execute_reply.started":"2023-12-09T22:21:47.5619Z","shell.execute_reply":"2023-12-09T22:21:47.601587Z"}}
df_artist = df.copy()
df_artist = df_artist.groupby('track_artist', group_keys=False)
df_artist_count = df_artist[numerical_columns].count()
df_artist_mean = df_artist[numerical_columns].mean()
artists_more_ten_musics = df_artist_count[(df_artist_count['track_popularity']>10)].index
df_artist_mean.filter(items = artists_more_ten_musics, axis=0)
# %% [code] {"execution":{"iopub.status.busy":"2023-12-09T22:21:47.603209Z","iopub.execute_input":"2023-12-09T22:21:47.603408Z","iopub.status.idle":"2023-12-09T22:21:47.621161Z","shell.execute_reply.started":"2023-12-09T22:21:47.60339Z","shell.execute_reply":"2023-12-09T22:21:47.620435Z"}}
# Top 10 Most popular artists
df_artist_mean.sort_values(by='track_popularity', ascending=False)[:10]
# %% [code] {"execution":{"iopub.status.busy":"2023-12-09T22:21:47.622152Z","iopub.execute_input":"2023-12-09T22:21:47.622751Z","iopub.status.idle":"2023-12-09T22:21:47.643141Z","shell.execute_reply.started":"2023-12-09T22:21:47.622723Z","shell.execute_reply":"2023-12-09T22:21:47.642418Z"}}
# Top 10 Most danceability artists
df_artist_mean.sort_values(by='danceability', ascending=False)[:10]
# %% [code] {"execution":{"iopub.status.busy":"2023-12-09T22:21:47.644288Z","iopub.execute_input":"2023-12-09T22:21:47.644502Z","iopub.status.idle":"2023-12-09T22:21:47.664334Z","shell.execute_reply.started":"2023-12-09T22:21:47.644482Z","shell.execute_reply":"2023-12-09T22:21:47.663829Z"}}
# Top 10 most acousticness artists
df_artist_mean.sort_values(by='acousticness', ascending=False)[:10]
# %% [code] {"execution":{"iopub.status.busy":"2023-12-09T22:21:47.665224Z","iopub.execute_input":"2023-12-09T22:21:47.665442Z","iopub.status.idle":"2023-12-09T22:21:48.26771Z","shell.execute_reply.started":"2023-12-09T22:21:47.665422Z","shell.execute_reply":"2023-12-09T22:21:48.267114Z"}}
sns.histplot(df, x='playlist_subgenre', hue='playlist_genre')
plt.xticks(rotation=90)
plt.show()
# %% [code] {"execution":{"iopub.status.busy":"2023-12-09T22:21:48.268916Z","iopub.execute_input":"2023-12-09T22:21:48.269131Z","iopub.status.idle":"2023-12-09T22:21:48.712572Z","shell.execute_reply.started":"2023-12-09T22:21:48.269111Z","shell.execute_reply":"2023-12-09T22:21:48.711733Z"}}
sns.histplot(df, x='track_popularity', bins=50, kde=True)
plt.title('Histogram - popularity')
plt.xlabel('Popularity')
# %% [code] {"execution":{"iopub.status.busy":"2023-12-09T22:21:48.713851Z","iopub.execute_input":"2023-12-09T22:21:48.714325Z","iopub.status.idle":"2023-12-09T22:21:49.163685Z","shell.execute_reply.started":"2023-12-09T22:21:48.714295Z","shell.execute_reply":"2023-12-09T22:21:49.162921Z"}}
sns.kdeplot(df, x='track_popularity', hue='playlist_genre')
plt.title('Histogram - popularity by genre')
plt.xlabel('Popularity')
# %% [code] {"execution":{"iopub.status.busy":"2023-12-09T22:21:49.164665Z","iopub.execute_input":"2023-12-09T22:21:49.165364Z","iopub.status.idle":"2023-12-09T22:21:49.626166Z","shell.execute_reply.started":"2023-12-09T22:21:49.165341Z","shell.execute_reply":"2023-12-09T22:21:49.625341Z"}}
sns.kdeplot(df, x='valence', hue='playlist_genre')
plt.title('Histogram - valence by genre')
plt.xlabel('Valence')
# %% [code] {"execution":{"iopub.status.busy":"2023-12-09T22:21:49.627126Z","iopub.execute_input":"2023-12-09T22:21:49.627329Z","iopub.status.idle":"2023-12-09T22:21:50.058693Z","shell.execute_reply.started":"2023-12-09T22:21:49.627309Z","shell.execute_reply":"2023-12-09T22:21:50.058031Z"}}
sns.kdeplot(df, x='danceability', hue='playlist_genre')
plt.title('Histogram - danceability by genre')
plt.xlabel('Danceability')
# %% [code] {"execution":{"iopub.status.busy":"2023-12-09T22:21:50.05974Z","iopub.execute_input":"2023-12-09T22:21:50.060231Z","iopub.status.idle":"2023-12-09T22:21:50.708186Z","shell.execute_reply.started":"2023-12-09T22:21:50.060208Z","shell.execute_reply":"2023-12-09T22:21:50.707385Z"}}
#

```

#

```

heatmap by musics
plt.figure(figsize=(10,7))
sns.heatmap(data=df[numerical_columns].corr(),annot=True,cmap='crest')
# %% [markdown]
## Recommendation System
# 1. Cosine-similarity
# 2. KNN
# %% [code] {"execution":{"iopub.status.busy":"2023-12-09T22:21:50.709207Z","iopub.execute_input":"2023-12-09T22:21:50.709441Z","iopub.status.idle":"2023-12-09T22:21:50.832927Z","shell.execute_reply.started":"2023-12-09T22:21:50.709419Z","shell.execute_reply":"2023-12-09T22:21:50.832115Z"}}
from sklearn.metrics.pairwise import cosine_similarity
# %% [code] {"execution":{"iopub.status.busy":"2023-12-09T22:21:50.834019Z","iopub.execute_input":"2023-12-09T22:21:50.834308Z","iopub.status.idle":"2023-12-09T22:21:50.866036Z","shell.execute_reply.started":"2023-12-09T22:21:50.834281Z","shell.execute_reply":"2023-12-09T22:21:50.865183Z"}}
df["line"] = [i for i in range(df.shape[0])]
df[7000:]
# %% [code] {"execution":{"iopub.status.busy":"2023-12-09T22:21:50.86739Z","iopub.execute_input":"2023-12-09T22:21:50.867874Z","iopub.status.idle":"2023-12-09T22:21:51.933112Z","shell.execute_reply.started":"2023-12-09T22:21:50.867846Z","shell.execute_reply":"2023-12-09T22:21:51.93217Z"}}
similarity = cosine_similarity(df[numerical_columns])
similarity
# %% [code] {"execution":{"iopub.status.busy":"2023-12-09T22:21:51.934386Z","iopub.execute_input":"2023-12-09T22:21:51.934697Z","iopub.status.idle":"2023-12-09T22:21:51.942806Z","shell.execute_reply.started":"2023-12-09T22:21:51.934669Z","shell.execute_reply":"2023-12-09T22:21:51.941921Z"}}
def recomendation_song_by_cossim(music, allow_same_genre=False, only_same_genre=False):
    genre_music = df[df.index == music]['playlist_genre'][0]
    id_music = df[df.index == music]['line'].values[0]
    distances = similarity[id_music]
    music_list = sorted(list(enumerate(distances)), reverse=True, key=lambda x:x[1])[1:]

    musics_recomendation = pd.DataFrame(columns=['Music','Artist','Album'])
    for music in music_list:

        if only_same_genre:
            if df[df['line'] == music[0]]['playlist_genre'][0] != genre_music:
                continue
        elif not allow_same_genre:
            if df[df['line'] == music[0]]['playlist_genre'][0] == genre_music:
                continue
        row = pd.DataFrame({'Music': df[df['line'] == music[0]].index[0], 'Artist':df[df['line'] ==
music[0]]['track_artist'][0],'Album':df[df['line'] == music[0]]['track_album_name'][0]})
        musics_recomendation = pd.concat([musics_recomendation, row], ignore_index=True)
        if musics_recomendation.shape[0] == 10:
            break
    return musics_recomendation
# %% [code] {"execution":{"iopub.status.busy":"2023-12-09T22:21:51.947325Z","iopub.execute_input":"2023-12-09T22:21:51.947587Z","iopub.status.idle":"2023-12-09T22:21:52.007364Z","shell.execute_reply.started":"2023-12-09T22:21:51.947566Z","shell.execute_reply":"2023-12-09T22:21:52.006722Z"}}
recomendation_song_by_cossim('Dance Monkey')
# %% [code] {"execution":{"iopub.status.busy":"2023-12-09T22:31:27.027363Z","iopub.execute_input":"2023-12-09T22:31:27.027672Z","iopub.status.idle":"2023-12-09T22:31:27.031439Z","shell.execute_reply.started":"2023-12-09T22:31:27.027648Z","shell.execute_reply":"2023-12-09T22:31:27.030578Z"}}
from sklearn.neighbors import NearestNeighbors
#from sklearn.preprocessing import StandardScaler, MaxAbsScaler, Normalizer
# %% [code] {"execution":{"iopub.status.busy":"2023-12-09T22:48:40.304595Z","iopub.execute_input":"2023-12-09T22:48:40.304936Z","iopub.status.idle":"2023-12-09T22:48:40.31349Z","shell.execute_reply.started":"2023-12-09T22:48:40.304913Z","shell.execute_reply":"2023-12-09T22:48:40.312588Z"}}
def recomendation_song_by_knn(music):
    x_ = df[numerical_columns]
    knn = NearestNeighbors(n_neighbors=11,algorithm='brute',p=2)
    knn.fit(x_)
    music = 'Dance Monkey'
    x = df[df.index == music][numerical_columns]
    distances, indices = knn.kneighbors(x)
    musics_recomendation = pd.DataFrame(columns=['Music','Artist','Album'])
    for idx in indices.flatten():

```

```
row = pd.DataFrame({'Music': df[df['line'] == idx].index[0], 'Artist':df[df['line'] == idx]['track_artist'][0],Album':df[df['line'] ==
idx]['track_album_name'][0]})
    musics_recomendation = pd.concat([musics_recomendation, row], ignore_index=True)
    return musics_recomendation[1:]
# %% [code] {"execution":{"iopub.status.busy":"2023-12-09T22:48:42.638904Z","iopub.execute_input":"2023-12-
09T22:48:42.639583Z","iopub.status.idle":"2023-12-09T22:48:42.682501Z","shell.execute_reply.started":"2023-12-
09T22:48:42.639557Z","shell.execute_reply":"2023-12-09T22:48:42.68168Z"}}
recomendation_song_by_knn('Dance Monkey')
```

ДОДАТОК Б
Характеристики треків Spotify

Таблиця Г.1 - Косинусна подібність та акустичні характеристики треків Spotify

track_name	track_artist	track_popularity	track_album_name	track_album_release_date	playlist_name	playlist_genre	playlist_subgenre	danceability	energy	key	loudness	mode	speechiness	acousticness	instrumentalness	liveness	valence	tempo	duration_ms	line
GUD VIBRATIONS	NGHTMRE	63	GUD VIBRATIONS	2016-10-26	Indie Poptimism	pop	indie poptimism	0.514	0.924	8	-5.417	1	0.0520	0.000208	0.805000	0.0684	0.0675	151.994	3.736850	7000
GUD VIBRATIONS - Habstrakt Remix	NGHTMRE	57	GUD VIBRATIONS (Habstrakt Remix)	2019-09-24	Nasty Bits	edm	electro house	0.681	0.969	11	-1.830	1	0.1300	0.004510	0.221000	0.5260	0.1840	124.936	3.394650	7001
GUMMO	6ix9ine	0	DAY69	2018-02-23	GANGSTARAP	rap	gangster rap	0.691	0.778	1	-4.957	1	0.1340	0.046500	0.000000	0.1360	0.6850	78.496	2.627383	7002
GUMMY	BROCKHAMPTON	65	SATURATION II	2017-08-25	Urban Contemporary	r&b	urban contemporary	0.687	0.672	0	-5.975	0	0.0573	0.198000	0.000000	0.1750	0.7920	80.035	4.350000	7003
GURU	Maski & Banga	28	GURU	2019-11-22	Big Room EDM	edm	big room	0.665	0.980	1	-1.533	0	0.1290	0.003900	0.040100	0.1050	0.2050	127.938	2.316400	7004
...
숙녀 (淑女) Lady	Yubin	17	都市女子	2018-06-05	Japanese Funk/Soul/NEO/Jazz/Acid	r&b	neo soul	0.473	0.973	9	-3.378	1	0.1700	0.234000	0.000051	0.4000	0.8470	119.810	3.268217	23444
심표	Kozyrop	30	Seoul Vibes, Pt. 4	2018-10-01	K-Crazy Michioso Tunes	edm	pop edm	0.429	0.724	3	-4.144	0	0.2000	0.855000	0.000000	0.1960	0.3770	82.199	2.502967	23445
중독	EXO-K	4	The 2nd Mini	2014-08-06	K-Crazy Michioso	edm	pop edm	0.498	0.917	5	-1.79	0	0.1640	0.061500	0.000000	0.3580	0.6520	112.357	3.426867	23444

Overdose			Album '중독 Overdose'		Tunes						7									6
피카부 Peek-A-Boo	Red Velvet	70	Perfect Velvet - The 2nd Album	2017-11-17	K-Party Dance Mix	pop	dance pop	0.839	0.902	1	-3.612	0	0.0536	0.086800	0.002570	0.2720	0.6390	114.953	3.150833	23447
하드캐리 Hard Carry	GOT7	1	FLIGHT LOG : TURBULENCE	2016-09-27	K-Party Dance Mix	pop	dance pop	0.700	0.873	1	-2.901	1	0.0869	0.014800	0.000000	0.3430	0.6120	99.972	3.226317	23448

ДОДАТОК В
АПРОБАЦІЯ ОТРИМАНИХ РЕЗУЛЬТАТІВ

МАТЕРІАЛИ
V ВСЕУКРАЇНСЬКОЇ СТУДЕНТСЬКОЇ НАУКОВОЇ
КОНФЕРЕНЦІЇ

17 ТРАВНЯ 2024 РІК • М. КИЇВ, УКРАЇНА

НАУКОВИЙ ПРОСТІР: АНАЛІЗ,
СУЧАСНИЙ СТАН, ТРЕНДИ ТА
ПЕРСПЕКТИВИ

ISBN 978-617-8312-44-2
DOI 10.36074/liga-ukr-17.05.2024



**УДК 082:001
Н 34**

Голова оргкомітету: Коренюк І.О.

Верстка: Зрада С.І.

Дизайн: Бондаренко І.В.

Рекомендовано до видання Вченою Радою Інституту науково-технічної інтеграції та співпраці. Протокол № 36 від 16.05.2024 року.



Конференцію зареєстровано Державною науковою установою «УкрІНТЕІ» в базі даних науково-технічних заходів України та інформаційному бюлетені «План проведення наукових, науково-технічних заходів в Україні» (Посвідчення №29 від 05.01.2024).

Матеріали конференції знаходяться у відкритому доступі на умовах ліцензії CC BY-SA 4.0 International.

Науковий простір: аналіз, сучасний стан, тренди та перспективи:
Н 34 матеріали V Всеукраїнської студентської наукової конференції, м. Київ, 17 травня, 2024 рік / ГО «Молодіжна наукова ліга». — Вінниця: ТОВ «УКРЛОГОС Груп», 2024. — 586 с.

ISBN 978-617-8312-44-2

DOI 10.62732/liga-ukr-17.05.2024

Викладено матеріали учасників V Всеукраїнської мультидисциплінарної студентської наукової конференції «Науковий простір: аналіз, сучасний стан, тренди та перспективи», яка відбулася 17 травня 2024 року у місті Київ, Україна.

УДК 082:001

© Колектив учасників конференції, 2024

© ГО «Молодіжна наукова ліга», 2024

© ТОВ «УКРЛОГОС Груп», 2024

ISBN 978-617-8312-44-2

АРХІТЕКТУРА МОДУЛЯ ПРОГНОЗУВАННЯ СЕРЦЕВИХ ЗАХВОРЮВАНЬ Закалюк П., Науковий керівник: Констанкевич Ю.В.	374
АРХІТЕКТУРА МОДУЛЯ РЕКОМЕНДАЦІЙ МУЗИКИ НА ОСНОВІ МАШИННОГО НАВЧАННЯ Черній І., Науковий керівник: Констанкевич Ю.В.	376
АРХІТЕКТУРА МОДУЛЯ РЕКОМЕНДАЦІЙ НА ОСНОВІ МАШИННОГО НАВЧАННЯ Філюк В., Науковий керівник: Констанкевич Ю.В.	378
АРХІТЕКТУРА МОДУЛЯ РОЗПІЗНАВАННЯ ВІКУ ТА СТАТІ ЛЮДЕЙ Мартиник В., Науковий керівник: Сапожжик Г.В.	380
АРХІТЕКТУРА МОДУЛЯ РОЗПІЗНАВАННЯ ЯКОСТІ ФРУКТІВ З ВИКОРИСТАННЯМ ГЛИБОКОГО НАВЧАННЯ Чіп С., Науковий керівник: Сапожжик Г.В.	382
ВИБІР ТА ОБГРУНТУВАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДЛЯ КОНТРОЛЮ ПАРАМЕТРІВ МІКРОКЛІМАТУ УКРИТТЯ Заровський С.В., Науковий керівник: Хімичева Г.І.	384
ВИКОРИСТАННЯ МАШИННОГО НАВЧАННЯ ДЛЯ ОПТИМІЗАЦІЇ ПРОВЕДЕННЯ СПІВБЕСІД Бурлаков О.О., Науковий керівник: Левус Є.В.	387
ВИКОРИСТАННЯ ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ ОСВІТНІХ КУРСІВ Горбовий В.Р., Науковий керівник: Левус Є.В.	389
ІНФОРМАЦІЙНА СИСТЕМА НАДАННЯ ТЕЛЕМЕДИЧНИХ КОНСУЛЬТАЦІЙ З ПІДТРИМКОЮ СТАНДАРТУ HL7 Смерек Л.Л., Науковий керівник: Кісь Я.П.	391
МЕТОДИ ТА ЗАСОБИ МОДУЛЯЦІЇ РАДІОТЕХНІЧНИХ СИГНАЛІВ АВТОГЕНЕРАТОРІВ Овчарук А.О., Науковий керівник: Осадчук В.С.	394
ПРОГНОЗУВАННЯ ЦІН НА АКЦІЇ ЗА ДОПОМОГОЮ ШТУЧНОГО ІНТЕЛЕКТУ Воробйов А.Р.	396
СИСТЕМА ДЛЯ ЗАБЕЗПЕЧЕННЯ ПЕРЕНАВЧАННЯ НЕЙРОННИХ СТРУКТУР МОЗКУ ЩОДО ФУНКЦІОНУВАННЯ ШЛЯХІВ ПЕРЕДАЧІ ЗОБРАЖЕННЯ ВІД ЗОРОВОГО НЕРВУ Гема О.Г., Науковий керівник: Кузьомін О.Я.	398
СЕКЦІЯ 17.	
ФІЗИКО-МАТЕМАТИЧНІ НАУКИ	
ОСОБЛИВОСТІ ДИСТАНЦІЙНОЇ ФОРМИ НАВЧАННЯ ПРЕДМЕТУ “ВИЩА МАТЕМАТИКА” НА СПЕЦІАЛЬНОСТІ “АРХІТЕКТУРА” Бондарець В.Ю., Науковий керівник: Турчанінова Л.І.	400

Черній Іван, здобувач вищої освіти факультету комп'ютерних інформаційних технологій
Західноукраїнський національний університет, Україна

Науковий керівник: Констанкевич Ю.В., канд. техн. наук, доцент,
доцент кафедри інформаційно-обчислювальних систем і управління
Західноукраїнський національний університет, Україна

АРХІТЕКТУРА МОДУЛЯ РЕКОМЕНДАЦІЙ МУЗИКИ НА ОСНОВІ МАШИННОГО НАВЧАННЯ

Структурно модуль (Рисунок 1) складається з кількох взаємопов'язаних компонентів, кожен з яких виконує специфічні функції, необхідні для ефективної роботи системи.



Рис. 1. Архітектура модулю рекомендацій музики на основі машинного навчання

Збір даних — це початковий етап, на якому відбувається збір необхідних даних для подальшого аналізу. Дані можуть включати інформацію про музичні треки, відгуки користувачів, історію прослуховувань та інші релевантні метадані.

Попередня обробка даних забезпечує фільтрацію та підготовку зібраних даних. Цей етап включає очищення від помилок, видалення дублікатів, а також нормалізацію та трансформацію даних для забезпечення їх готовності до аналізу.

Інженерія ознак полягає у виборі ознак з попередньо оброблених даних, які будуть використовуватися для тренування моделі машинного навчання. Вибір ефективних ознак є критичним для забезпечення високої точності рекомендацій.

Модель машинного навчання розробляється на основі обраних ознак і відповідно до певного алгоритму навчання. Цей компонент відповідає за тренування моделі, яка здатна прогнозувати користувацькі переваги та рекомендувати музичні треки.

Двигун рекомендацій використовує треновану модель для генерації рекомендацій, які представляються користувачу через користувацький інтерфейс. Цей інтерфейс забезпечує зручний доступ до рекомендованих треків та можливість взаємодії з системою.

Зворотній зв'язок користувача зібраний через користувацький інтерфейс, є важливим для подальшого удосконалення моделі машинного навчання, оскільки він дозволяє системі адаптуватися до змін у користувацьких перевагах та покращувати якість рекомендацій.

Таким чином, розгорнута система рекомендацій забезпечує високий рівень персоналізації музичних пропозицій, базуючись на детальному аналізі великих обсягів даних та їхній ефективній обробці з використанням сучасних методів машинного навчання.

Список використаних джерел:

2. Краліва Г., Баков Н. ПРОБЛЕМИ ЯКОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ. DEBATS SCIENTIFIQUES ET ORIENTATIONS PROSPECTIVES DU DEVELOPPEMENT SCIENTIFIQUE. 2021. URL: <https://doi.org/10.36074/logos-05.02.2021.v3.29> (дата звернення: 13.05.2024).