

ЩЕРБАК ВЛАДИСЛАВ ВОЛОДИМИРОВИЧ

Допускається до захисту:
завідувач кафедри
інформаційних технологій,
д-р техн. наук, доцент
_____ Тетяна НЕСКОРОДЕВА
« _____ » _____ 2022р.

**РОЗРОБКА РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ ДЛЯ АНАЛІЗУ
МУЗИЧНОГО КОНТЕНТУ**

Спеціальність 122 «Комп'ютерні науки»
Кваліфікаційна (магістерська) робота

Науковий керівник:
Римар П. В., старший викладач
кафедри інформаційних технологій

Науковий консультант:
Нескородева Т.В.,
д-р техн. наук, доцент

(підпис)

Оцінка: _____ / _____ / _____
(бали за шкалою ЄКТС/за національною шкалою)

Голова ЕК: _____
(підпис)

АНОТАЦІЯ

Щербак В.В. Розробка рекомендаційної системи для аналізу музичного контенту. Спеціальність 122 “Комп’ютерні науки”, Освітня програма “Комп’ютерні технології обробки даних”. Донецький національний університет імені Василя Стуса, Вінниця, 2022.

У кваліфікаційній роботі досліджено рекомендаційні системи та алгоритми підбору музичного контенту. Показано реалізацію своєї рекомендаційної системи підбору музичного контенту.

Ключові слова: мета-данні, рекомендаційна-система, mrs, fma, алгоритми пошуку, API.

50с., 17 рис., 1 табл., 1 дод., 12 джерел.

ANNOTATION

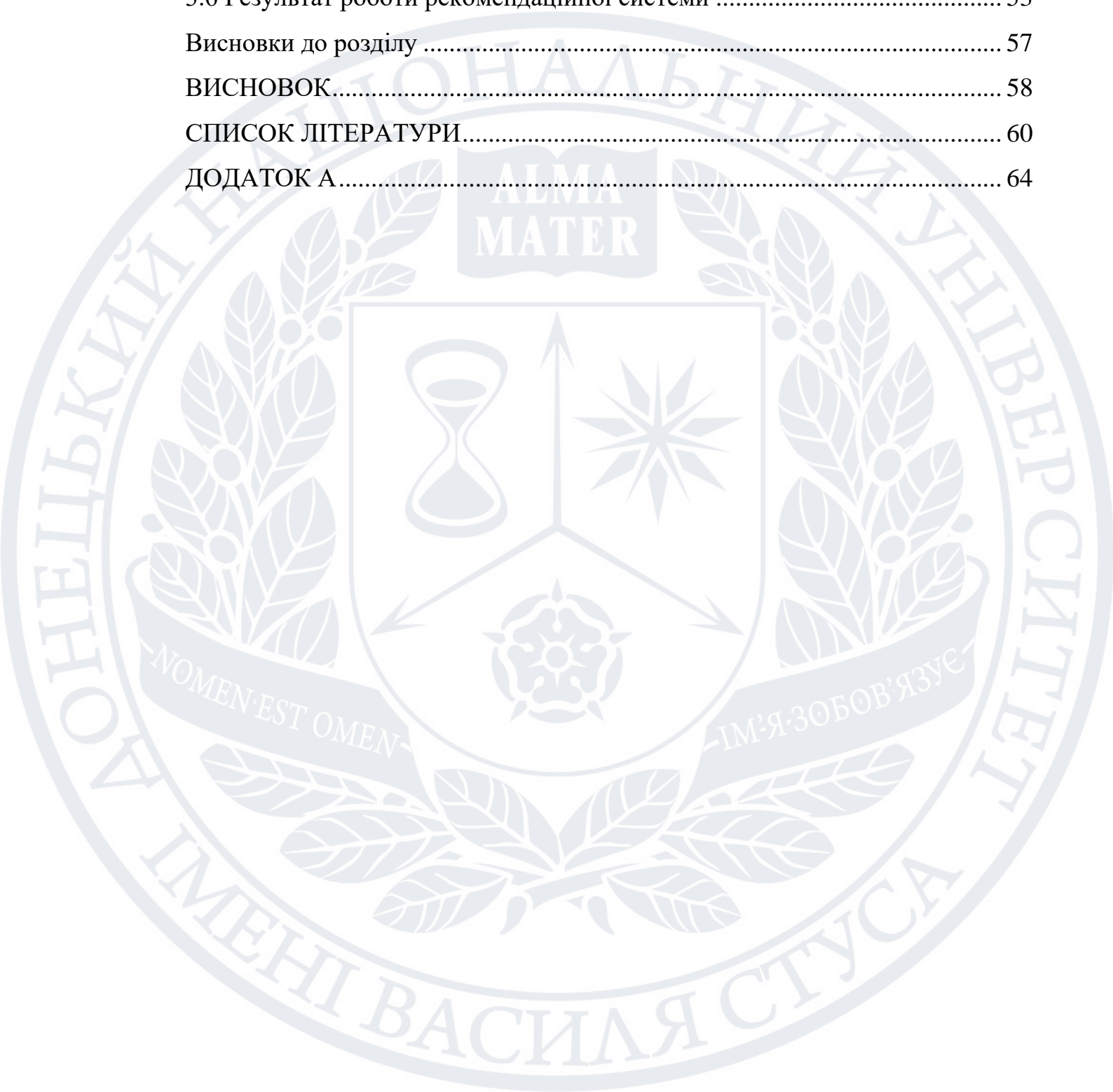
Shcherbak V.V. Development of a recommendation system for the analysis of musical content. Specialty 122 "Computer science", Educational program "Computer data processing technologies". Vasyl' Stus Donetsk National University, Vinnytsia, 2022. Recommendation systems and algorithms for the selection of musical content were investigated in the qualification work. The implementation of its recommendation system for selecting music content is shown.

Keywords: meta-data, recommender-system, mrs, fma, search algorithms, API.

50p., 17 fig., 1 tab., 1 a., 12 ref.,

ВСТУП	5
РОЗДІЛ 1. МАШИННЕ НАВЧАННЯ.РЕКОМЕНДАЦІЙНИ СИСТЕМИ ТА АЛГОРИТМИ ПІДБОРУ	7
1.1 Машинне навчання.....	7
1.2 Рекомендаційна система.....	10
1.3 Фільтрування вмісту	11
1.4 Колаборативна фільтрація.....	13
1.5 Гібридні методи фільтрації	16
1.6 Методи інформаційного пошуку в РС	17
1.7 Проблеми, пов'язані з РС.....	18
1.8 Майбутні напрями РС	21
Висновки до розділу	22
РОЗДІЛ 2. ПОНЯТТЯ МУЗИЧНОГО АУДІО-СТРИМІНГУ ТА АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ	23
2.1 Поняття музичного аудіо-стрімінгу	23
2.2 Pandora Internet Radio.....	24
2.3 Spotify	25
2.4 YouTube Music.....	26
2.5 Визначення параметрів оцінювання музики	27
2.6 Визначення вимог до системи	30
2.7 Аналіз алгоритму та систем фільтрація.....	31
Висновки до розділу	37
РОЗДІЛ 3. ПОНЯТТЯ ВЕБ-ДОДАТКУ, АРХІТЕКТУРА ВЕБ-ДОДАТКІВ, ОБРАННЯ МОВ ПРОГРАМУВАННЯ, ІНСТРУМЕНТІВ РОЗРОБКИ. ПОБУДОВА ВЕБ-ДОДАТКУ	38
3.1 Поняття веб-додатку та архітектура веб-додатків.....	38
3.2 Вибір технологій	40
3.3 Початок роботи.....	43
3.4 Структура проекту	44

3.5 Практична реалізація рекомендаційної системи обраними мовами програмування.....	45
3.6 Результат роботи рекомендаційної системи	53
Висновки до розділу	57
ВИСНОВОК.....	58
СПИСОК ЛІТЕРАТУРИ.....	60
ДОДАТОК А.....	64



ВСТУП

На сучасному етапі процес діджиталізації виступає одним з провідних факторів економічного зростання підприємств та всього суспільства. Зростання діджиталізації зумовлює зміни у зовнішньому середовищі, зокрема вплив науково-технічного прогресу на формування нових моделей споживчої поведінки та посилення інформаційних процесів. Використання цифрових технологій у своїй діяльності значно сприяє збільшенню спектру можливостей.

Завдяки впровадженню діджиталізації можна легко здійснювати обробку будь-якої інформації, що в результаті пришвидшує роботу людини. Якщо поєднати різну інформацію, то можна отримати дуже зручну рекомендаційну систему. Дана система здійснює аналіз пошукових запитів та визначає вподобання, на основі яких підбирає рекомендації.

Наприклад, проаналізувавши пошукові запити та покупки в інтернет-магазині конкретної людини, рекомендаційна система зможе підібрати актуальні рекомендації на основі її вподобань.

Такий самий принцип рекомендацій мають музичні додатки. Розглянемо детальніше принцип роботи рекомендаційної системи саме в цих додатках.

Музика посідає одне з важливих місць в житті кожної людини. Люди слухають її скрізь, навіть інколи не помічаючи її фонове звучання. Завдяки розвитку сучасних технологій та цифровізації з'явилася можливість самостійно обирати жанр музики, виконавців та насолоджуватися улюбленими піснями у музичних додатках. Це дуже зручно, враховуючи швидкий ритм життя. Перевагою є те, що ці додатки можна встановити на будь-який пристрій. Не важливо чи людина їде у подорож, на роботу, займається спортом, у неї завжди буде можливість увімкнути будь-яку пісню. Наступною перевагою музичних додатків є можливість створювати власний плей-лист. Можна створювати їх виходячи з різноманітних жанрів чи особистих вподобань. І саме тут в роботу вступає рекомендаційна система для підбору музичного контенту.

При реєстрації в музичному додатку користувачеві пропонується обрати улюблений жанр та виконавців з запропонованого переліку. Після цього

система розуміє, яку саме музику необхідно запропонувати конкретному користувачеві, виходячи з його уподобань. Але якщо користувач не хоче проходити це опитування, то система пропонує йому найбільш популярну музику серед всіх слухачів, та дивиться на реакцію до запропонованих треків.

Після створення власного плей-листу, рекомендаційна система здійснить його аналіз та сформує власну підбірку пісень, на основі ваших вподобань. Це дуже зручно, адже, користувачеві не потрібно витратити час на пошук музики, за нього це зробить система.

Актуальність. Зараз все більше і більше послуг переходить в інтернет, тому кожному сайту, який надає послуги такі як: купівля товару, прослуховування музичного контенту, перегляд фільмів та серіалів потрібно мати рекомендаційну систему.

Мета магістерської роботи - розробка додатку, який надає змогу отримати рекомендації музичного контенту.

Для цього потрібно:

- зробити огляд існуючих аналогів;
- зробити аналіз існуючих алгоритмів для побудови рекомендаційних систем, виявили їх переваги та недоліки;
- розробити ефективну модель формування рекомендацій на основі алгоритму фільтрування вмісту.

Об'єктом дослідження є рекомендаційна система по наданню музичного контенту.

Предметом дослідження є алгоритми за допомогою яких можна розробити рекомендаційну систему.

РОЗДІЛ 1. МАШИННЕ НАВЧАННЯ. РЕКОМЕНДАЦІЙНИ СИСТЕМИ ТА АЛГОРИТМИ ПІДБОРУ

Фільтрування інформації - це процес аналізу великих обсягів даних та знаходження лише тих даних, які повністю відповідають його потребам. Існує два види систем фільтрування інформації: фільтрувальні та рекомендаційні. За приклад фільтрувальних інтерфейсів можна взяти фільтри електронної пошти, вони сортують вхідні листи по категоріям та атрибутами, завдяки цьому користувач може швидко знайти потрібну йому інформацію в своїй пошті. За приклад систем з рекомендаційним інтерфейсом можна взяти інтернет магазин товарів, який на основі переглянутих користувачем товарів, робить рекомендації що можуть сподобатись йому. Також до таких систем можна віднести сервіси по перегляду фільмів і прослуховування музичного контенту. Перед розглядом алгоритмів для розробки рекомендаційної системи спочатку треба розглянути що таке машинне навчання

1.1 Машинне навчання

Машинне навчання — це концепція, згідно з якою комп'ютерна програма може навчатися та адаптуватися до нових даних без втручання людини.

Машинне навчання – це сфера штучного інтелекту (ШІ), яка підтримує актуальність вбудованих алгоритмів комп'ютера незалежно від змін у світовій економіці.

Різні сектори економіки мають справу з величезною кількістю даних, доступних у різних форматах із різних джерел. Величезний обсяг даних, відомий як великі дані, стає легкодоступним і доступним завдяки прогресивному використанню технологій, зокрема передових обчислювальних можливостей і хмарного сховища. Компанії та уряди усвідомлюють величезні знання, які можна отримати, опрацювавши великі дані, але їм не вистачає ресурсів і часу, необхідних для аналізу багатства інформації. Таким чином,

заходи штучного інтелекту використовуються різними галузями для збору, обробки, передачі та обміну корисною інформацією з наборів даних. Одним із методів ШІ, який все частіше використовується для обробки великих даних, є машинне навчання. Різні додатки даних машинного навчання формуються за допомогою складного алгоритму або вихідного коду, вбудованого в машину чи комп'ютер. Цей код програмування створює модель, яка ідентифікує дані та будує прогнози навколо даних, які вона ідентифікує. Модель використовує параметри, вбудовані в алгоритм, щоб сформувані шаблони для процесу прийняття рішень. Коли нові або додаткові дані стають доступними, алгоритм автоматично налаштовує параметри, щоб перевірити наявність змін шаблону, якщо такі є. Однак модель не повинна змінюватися.

Машинне навчання часто використовують в таких випадках як:

- фінансове консультування та управління портфелем;
- двигуни для передбачування рекомендацій (наприклад інтернет магазин, музичний додаток,);
- прогноз ризику смертності від Covid-19;
- машинне навчання проти ботів;

Існує багато алгоритмів машинного навчання, нижче розглянуто найбільш прості та популярні:

- Алгоритм K Means Clustering — це тип неконтрольованого навчання, який використовується для категоризації немаркованих даних, тобто даних без визначених категорій або груп. Алгоритм працює шляхом знаходження груп у даних із кількістю груп, представлених змінною K. Потім він працює ітеративно, щоб призначити кожному даним одну з K груп на основі наданих функцій.
- Дерево рішень — це деревоподібна структура, схожа на блок-схему, яка використовує метод розгалуження, щоб проілюструвати всі можливі результати рішення. Кожен вузол дерева представляє перевірку певної змінної, а кожна гілка є результатом цієї перевірки.

- Random forest — це метод ансамблевого навчання, який поєднує кілька алгоритмів для отримання кращих результатів для класифікації, регресії та інших завдань. Кожен окремий класифікатор слабкий, але в поєднанні з іншими може дати чудові результати. Алгоритм починається з «дерева рішень» (деревоподібного графіка або моделі рішень), а вхідні дані вводяться зверху. Потім він рухається вниз по дереву, а дані сегментуються на все менші набори на основі конкретних змінних.
- Алгоритм K-Nearest-Neighbor оцінює, наскільки ймовірно, що точка даних буде членом тієї чи іншої групи. По суті, він переглядає точки даних навколо однієї точки даних, щоб визначити, до якої групи вона насправді належить. Наприклад, якщо одна точка знаходиться на сітці, і алгоритм намагається визначити, до якої групи ця точка даних (група А або група В, наприклад) він перегляне точки даних поруч із ним, щоб побачити, до якої групи входить більшість точок.

Переваги та недоліки машинного навчання.

Машинне навчання бачило випадки використання, починаючи від прогнозування поведінки клієнтів і закінчуючи формуванням операційної системи для самокерованих автомобілів.

Коли справа доходить до переваг, машинне навчання може допомогти підприємствам глибше зрозуміти своїх клієнтів. Збираючи дані про клієнтів і співвідносячи їх із поведінкою з часом, алгоритми машинного навчання можуть вивчати асоціації та допомагати командам пристосовувати розробку продукту та маркетингові ініціативи до попиту клієнтів.

Багато продуктових компаній використовують машинне навчання, щоб запропонувати користувачу найбільш підходящу послугу. Наприклад Rozetka, використовує алгоритми для рекомендацій товарі, на основі раніше куплених

або раніше переглянутих товарах. Google використовує машинне навчання, щоб показувати рекламу у пошукових запитах.

Але машинне навчання має недоліки. По-перше, це може бути дорого. Проектами машинного навчання зазвичай керують спеціалісти з обробки даних, які отримують високу зарплату. Ці проекти також потребують програмної інфраструктури, яка може бути дорогою.

Існує також проблема упередженості машинного навчання. Алгоритми, навчені на наборах даних, які виключають певні популяції або містять помилки, можуть призвести до неточних моделей світу, які, в кращому випадку, будуть невдалими, а в гіршому – дискримінаційними. Коли підприємство базує основні бізнес-процеси на упереджених моделях, воно може завдати шкоди регуляторним органам і репутації.

1.2 Рекомендаційна система

Системи рекомендацій – це системи, які призначені для рекомендацій користувачам на основі багатьох різних факторів. Ці системи передбачають найімовірніший продукт, який користувачі найімовірніше придбають і який представляє інтерес. Такі компанії, як Netflix, Amazon тощо, використовують системи рекомендацій, щоб допомогти своїм користувачам визначити правильний продукт або фільми для них.

Система рекомендацій працює з великим обсягом інформації, фільтруючи найважливішу інформацію на основі даних, наданих користувачем, та інших факторів, які враховують уподобання та інтереси користувача. Він визначає відповідність між користувачем і елементом і визначає схожість між користувачами та елементами для рекомендації. І користувачі, і послуги, що надаються, отримали вигоду від таких систем. Завдяки таким системам якість і процес прийняття рішень також покращилися.

Системи рекомендацій є ефективними інструментами для фільтрації онлайн-інформації, яка широко поширена через зміну звичок користувачів комп'ютерів, тенденції персоналізації та нові доступи до Інтернету. Навіть

незважаючи на те, що останні рекомендаційні системи видатні у наданні точних рекомендацій, вони страждають від різноманітних обмежень і проблем, таких як масштабованість, холодний запуск, розрідженість тощо.

Через існування різноманітних методів вибір методів стає складною роботою під час створення програми -орієнтовані системи рекомендацій. Крім того, кожна техніка має власний набір особливостей, переваг і недоліків, що викликає ще більше питань, на які слід звернути увагу. Ця стаття спрямована на систематичний огляд різноманітних нещодавніх внесків у область рекомендаційних систем, зосереджуючись на різноманітних додатках, таких як книги, фільми, продукти тощо. Спочатку аналізуються різні застосування кожної рекомендаційної системи. Потім виконується алгоритмічний аналіз різних систем рекомендацій і складається таксономія, яка враховує різні компоненти, необхідні для розробки ефективної системи рекомендацій. Крім того, оцінюються та зазначаються зібрані набори даних, платформа моделювання та показники продуктивності, зосереджені на кожному внеску. Нарешті, цей огляд надає вкрай необхідний огляд поточного стану досліджень у цій галузі та вказує на існуючі прогалини та проблеми, щоб допомогти нащадкам розробити ефективну систему рекомендацій.

Існує три основні підходи до рекомендацій:

1. Фільтрування вмісту (*item-based*).
2. Колаборативна фільтрація (*user-based*).
3. Гібридний підхід.

1.3 Фільтрування вмісту

Алгоритми фільтрації на основі вмісту призначені для рекомендації продуктів на основі накопичених знань користувачів. Ця техніка спрямована на порівняння інтересів користувача з характеристиками продукту, тому важливо надати значущу характеристику продуктів у системі.[1]

Перед розробкою системи першочерговим завданням має бути вибір улюблених функцій кожного покупця. Ці дві стратегії можна застосовувати у можливій комбінації. По-перше, користувачеві надається список функцій, щоб вибрати найбільш цікаві. По-друге, алгоритми зберігають облік усіх продуктів, вибраних користувачем у минулому, і формують дані про поведінку клієнта.

Профіль покупця змінюється навколо вибору, смаків і вподобань покупця та формує рейтинг покупця. Він включає в себе, скільки разів окремий покупець натискає на зацікавлені продукти або скільки разів подобається ці продукти в списку бажань. Фільтрування на основі вмісту полягає в схожості між елементами. Близькість і схожість продукту вимірюються на основі подібного вмісту товару. Коли ми говоримо про контент, він включає жанр, категорію предмета тощо. Візьмемо для прикладу системи рекомендацій у фільмах.

Припустімо, є чотири фільми, у яких користувачеві спочатку подобаються лише два фільми. Проте 3-й фільм схожий на 1-й за жанром, тому система автоматично запропонує 3-й фільм. Це те, що автоматично генерується системою рекомендацій на основі вмісту на основі подібності вмісту.

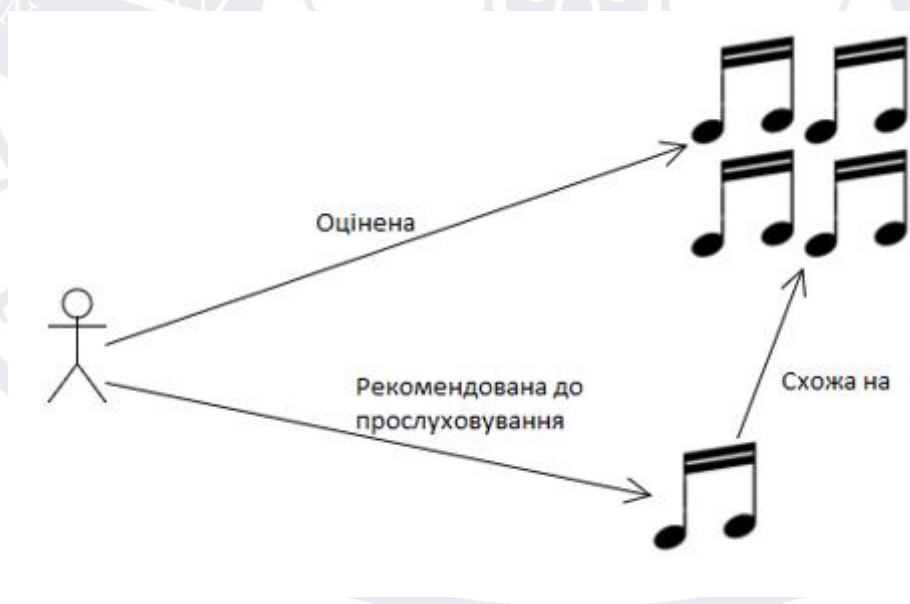


Рис. 1.1 Алгоритм фільтрування вмісту

Система фільтрації на основі вмісту не вимагає жодної інформації про покупця, оскільки пропозиція стосується лише покупця та полегшує масштабування для багатьох покупців. Ця система вловлює інтереси користувачів і пропонує товари, якими користується небагато покупців.

1.4 Колаборативна фільтрація

Колаборативна фільтрація базується на тому, що існують зв'язки між продуктами та інтересами людей. Багато систем рекомендацій використовують спільну фільтрацію, щоб знайти ці зв'язки та дати точну рекомендацію продукту, який може сподобатися користувачеві чи зацікавити його. Демонстрація алгоритму представлена на рисунку 1.2

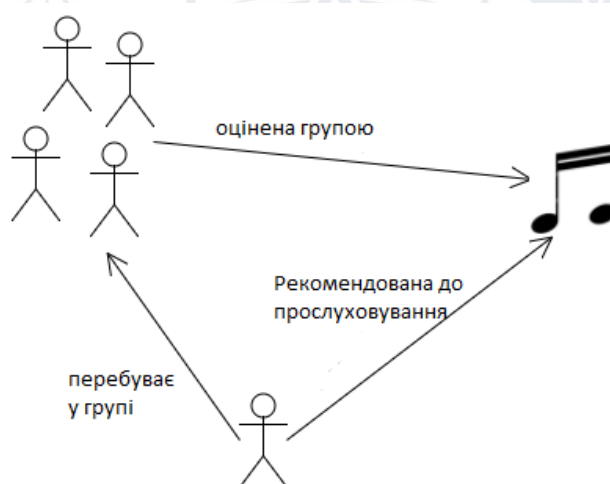


Рис. 1.2 Алгоритм колаборативної фільтрації

Колаборативна фільтрація має два підходи: на основі користувачів і на основі змісту. Колаборативна фільтрація на основі користувачів базується на схожості користувачів або сусідстві. Спільна фільтрація на основі елементів базується на подібності між елементами.

У колаборативній фільтрації на основі користувачів у нас є активний користувач, для якого націлена рекомендація. Механізм спільної фільтрації спочатку шукає схожих користувачів. Це користувачі, які поділяють моделі

рейтингу активних користувачів. Спільна фільтрація ґрунтується на цій подібності на таких речах, як історія, уподобання та вибір, який користувачі роблять під час покупки, перегляду або задоволення від чогось. Наприклад, музичні композиції, які подібні користувачі оцінили високо. Потім він використовує оцінки цих схожих користувачів, щоб передбачити можливі оцінки активного користувача для муз. композиції, які він раніше не слухав. Наприклад, якщо 3 користувачі схожі або є сусідами за цікавими музичними композиціями, ми можемо порекомендувати активному користувачеві муз. композицію, яку вже бачили її сусіди (Рис 1.3)

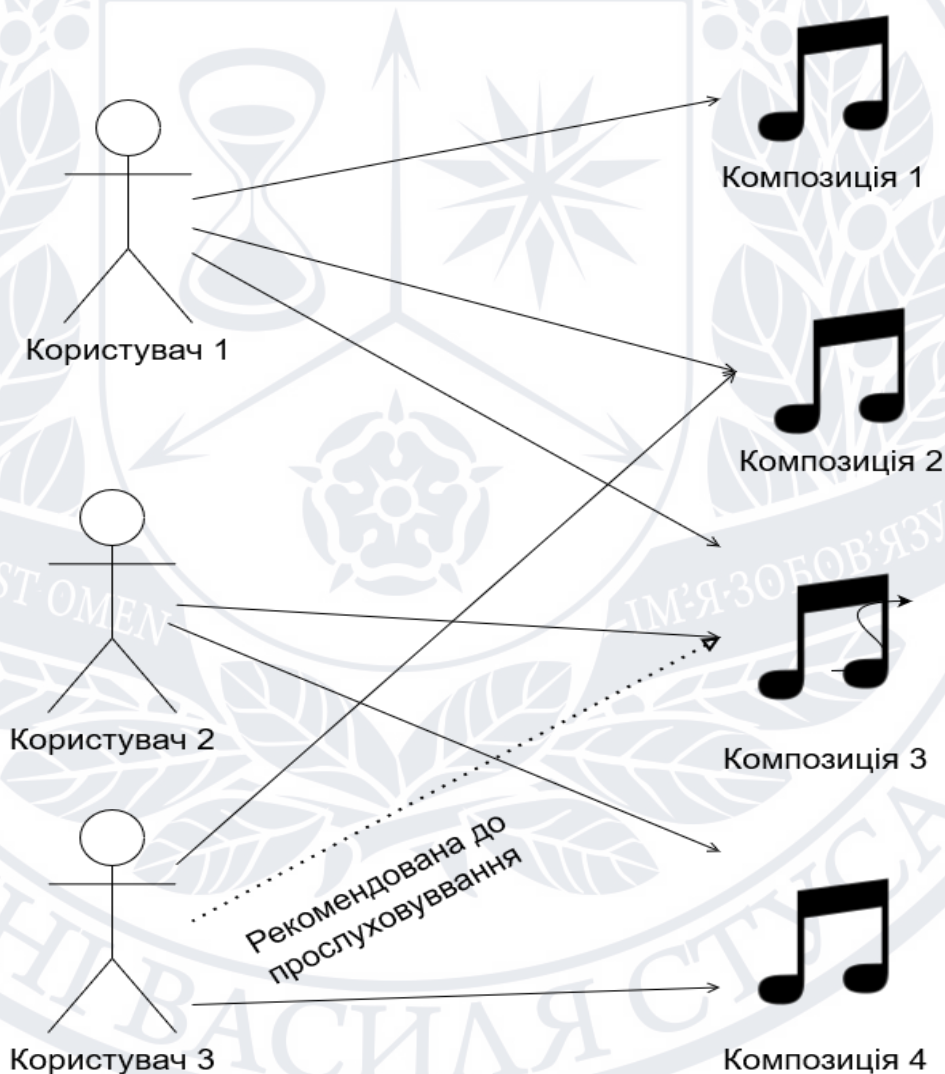


Рис.1.3 - Колаборативна фільтрація на основі користувачів

Колаборативна фільтрація на основі вмісту дає рекомендації на основі взаємодії користувача з продуктом у минулому. Припущення, що лежить в основі алгоритму, полягає в тому, що користувачам подобаються подібні продукти, яким він поставив гарний відгук та не подобаються схожі продукти з тими, де в нього поганий відгук, тому вони дають однакові оцінки подібним продуктам.

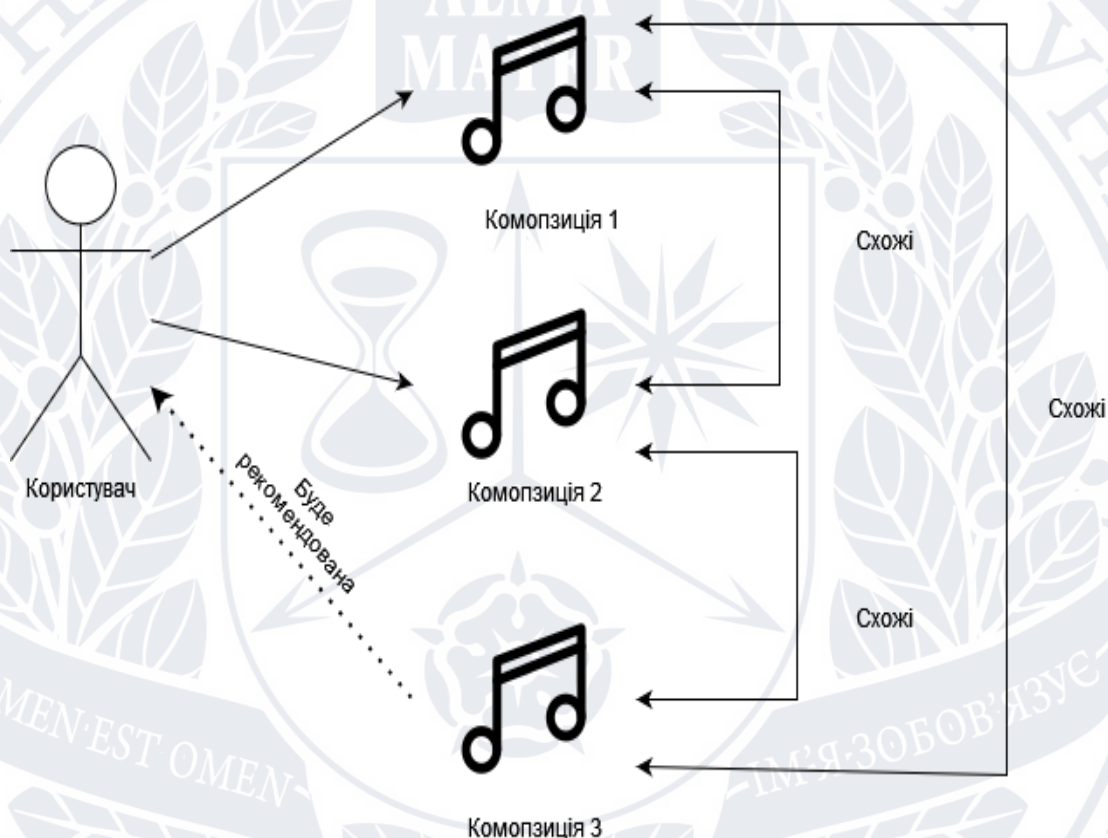


Рис. 1.4 Колаборативна фільтрація на основі вмісту

У підході, орієнтованому на користувача, рекомендація ґрунтується на користувачах тої групи, з якими користувач має спільні переваги. Наприклад, оскільки Користувачу 1 і Користувачу 3 сподобалися Елемент 3 і Елемент 4, ми вважаємо їх схожими або сусідніми користувачами та рекомендуємо Елемент 1, який Користувач 1 позитивно оцінив для Користувача 3. У підході на основі елементів подібні елементи будувати околиці на поведінці користувачів. Однак зауважте, що він не базується на їхньому вмісті. Наприклад, Елемент 1 і

Елемент 3 вважаються сусідами, оскільки вони були позитивно оцінені як Користувачем 1, так і Користувачем 2. Отже, Елемент 1 можна рекомендувати Користувачеві 3, оскільки він уже виявив інтерес до Елементу 3. Тому рекомендації тут такі: на основі елементів у сусідстві, які користувач може віддати перевагу.

1.5 Гібридні методи фільтрації

Більшість систем рекомендацій зараз використовують гібридний підхід (Рис. 1.5), поєднуючи спільну фільтрацію, фільтрацію на основі вмісту та інші підходи. Немає жодних причин, чому кілька різних технік одного типу не можуть бути змішані. Гібридні підходи можна реалізувати декількома способами: створюючи прогнози на основі контенту та на основі співпраці окремо, а потім об'єднуючи їх; шляхом додавання можливостей на основі контенту до підходу на основі співпраці (і навпаки); або шляхом уніфікації підходів в одну модель. Декілька досліджень, які емпірично порівнюють ефективність гібридних методів із суто спільними методами та методами, заснованими на вмісті, продемонстрували, що гібридні методи можуть надати більш точні рекомендації, ніж чисті підходи. Ці методи також можна використовувати для подолання деяких загальних проблем у системах рекомендацій, таких як холодний запуск і проблема розрідженості, а також вузьке місце інженерії знань у підходах, заснованих на знаннях. Netflix є хорошим прикладом використання гібридних систем рекомендацій. Веб-сайт дає рекомендації, порівнюючи звички перегляду та пошуку подібних користувачів (тобто спільне фільтрування), а також пропонуючи фільми, які мають спільні характеристики з фільмами, які користувачі високо оцінили (фільтрування на основі вмісту).

Є декілька основних варіантів комбінування різних методів у рекомендаційних системах гібридного типу:

1. Введення єдиної моделі, яка використовує характеристики обох

методів.

2. Інтегрування item-based характеристик в user-based системи.
3. Інтегрування user-based характеристик у item-based системи.
4. Інтегрування item-based та user-based підходів окремо.

Спростовано цей метод можна зобразити так:



Рис. 1.5 – Гібридна модель рекомендаційної системи

1.6 Методи інформаційного пошуку в РС

Численні джерела інформації наповнили цифровий світ необмеженими даними. Сценарій був перебільшений завдяки інтерактивній участі людей. Для ефективної та плідної рекомендації РС необхідно вивчити всі можливі зони для отримання й аналізу інформативних даних, щоб зрозуміти переваги людей і смаки. Для виконання цієї роботи кожен РС використовує деякі методи пошуку інформації. Згадано деякі з найпопулярніших методів пошуку інформації, які використовуються в RS нижче.

- Машинне навчання

Надає суб'єкту можливість навчатися штучно, без програмування явно. Він застосовує різні алгоритми, такі як логістична регресія, рішення дерева, навчання правил асоціації, кластер, байєсовські мережі та опорна векторна машина, тощо.

- **Логістична регресія**

Використовується для прогнозування дискретних змінних за допомогою неперервної дискретних даних.

- **Дерево рішень**

Це потужна техніка, яка допомагає вибрати варіант серед кількох альтернатив. У РС він використовується для розрахунку та прогнозування відсутніх переваг користувачів.

- **Вивчення правила асоціації**

Використовується для виділення частих моделей, асоціацій, кореляції або причинно-наслідкові структури з набору даних користувачів і елементів для рекомендацій.

1.7 Проблеми, пов'язані з РС

- **Обмежений аналіз вмісту**

У CBRS точність рекомендацій залежить від обсягу введених користувачем даних. Якщо RS не містить достатньої інформації про користувача, продуктивність рекомендація буде низькою. Жодна система CBR не може надати відповідних пропозицій, якщо проаналізований вміст не містить достатньо інформації, щоб розрізнити елементи, які подобаються користувачеві з предметів, які не подобаються користувачеві. Ця проблема відома як обмежена проблема аналізу контенту. Щоб скласти точну рекомендацію, повний домен потрібна інформація. Наприклад, РС для фільмів повинен мати всю інформацію пов'язані з певним фільмом (наприклад, жанр, актори, режисери тощо). Але зібравши все інформація, пов'язана з певним доменом, дуже складна, особливо для мультимедіа такі елементи, як зображення, аудіо-

та відеопотоки тощо. Отже, цю проблему також називають а проблема залежності домену. Цю проблему можна вирішити шляхом прийняття KBRS.

- Надмірна спеціалізація

Мета РС — допомогти користувачам досліджувати нові продукти. Важливою рисою є різноманітність хорошого РС. На жаль, деякі алгоритми рекомендацій можуть робити протилежність. Вони, як правило, рекомендують популярні та високо оцінені товари, які подобаються конкретному користувачу. Щоб подолати цю проблему, необхідно розробити нові гібридні підходи, які підвищать ефективність процесу рекомендацій. Методи навчання, які застосовуються до CBF, намагаються знайти найбільше відповідні документи на основі поведінки користувача в минулому. Такий підхід, обмежує користувача документами, подібними до тих, що вже були. Це відомо як проблема надмірної спеціалізації.

- Холодний старт

Коли новий елемент або новий користувач вводяться в РС, система не матиме минулого записи (рейтинги, уподобання, історія пошуку тощо), на основі яких рекомендацій має бути зроблено. Це є відома проблема холодного запуску. Це також називається проблемою нового користувача або новим елементом проблема. Рішення цієї проблеми передбачає використання демографічної інформації користувач отриманого з його профілю. Це рішення є недостатнім і не повністю правильним, оскільки користувачі з однаковими демографічними характеристиками можуть проявляти різні інтереси до певного предмету.

- Розрідженість

На практиці РС працюють з дуже великими наборами даних. Отже, матриця елементів користувача, яка використовується для CF надзвичайно розріджена, що негативно впливає на ефективність прогнозів або рекомендації систем CF. Це також має місце, коли користувач, використавши деякі конкретного продукту, не став оцінювати його. В інших випадках користувачі не оцінюють

предмети, які є їм не відомі. Щоб подолати цю проблему, РС використовує підхід, званий методом кластеризації. Метод кластеризації уточнює дані відповідно до уподобань користувача отже, це полегшує рекомендацію товарів. На жаль, є певні проблеми ще не вирішені у випадку багаторівневої кластеризації.

- Масштабованість

Оскільки RS працюють із великими наборами даних, складність RS зростає у випадку величезного кількості користувачів і мільйони окремих елементів. Багато систем повинні реагувати негайно, ознайомитись із вимогами в Інтернеті та виробити рекомендації для всіх користувачів на основі їх покупок та історії рейтингів, що вимагає високої масштабованості елементів.

- Синоніміка

Синонімія відноситься до проблеми кількох слів, які мають подібні значення. Більшість RS не можуть знайти те саме чи подібне з різними назвами (синонімами). Через цю нездатність деякі асоційовані виникають проблеми. Наприклад, «дитячий фільм» і «дитяче кіно» в основному позначають однакові елементи, але системи CF на основі пам'яті не знайдуть відповідності між ними схожість обчислень.

- Аббревіатура

Якщо РС не знайомий зі скороченнями, які користувачі часто використовують під час онлайн взаємодії, він не зможе розпізнати предмет, який шукає користувач. Це створює помилкову рекомендацію. Рішення полягає в тому, щоб класифікувати скорочене слова з їх повними формами та помістити обидві назви в один список.

- Проблема чорної скриньки

Ефективність РС підвищується зі збільшенням прозорості рекомендацій. Задоволеність користувача в рекомендації заплутана. Проблема виникає, коли система є непрозорою для кінцевого користувача, що спричиняє зниження рівня довіри до системи. Потенціал рекомендації, таким чином, зменшується

проблемами чорної скриньки, які викликають у користувачів недовіру, що призводить до невдачі спроба рекомендації. Ця проблема вирішується шляхом надання інформація для користувача щодо основи, на якій зроблені рекомендації їх, наприклад, шаблони інтересів інших користувачів, профіль, створений їхніми власними інтерес тощо. Це спонукає користувачів шукати товар, оскільки вони піддаються певному впливу за вибором інших користувачів, які також купили той самий товар. Надаючи прозорі причини для користувача, такі як оцінки елемента, надані схожими користувачами, користувачі мотивовані довіряти рекомендаціям, а також рекомендовані. Наприклад, Netflix виконує фільм рекомендації своїм користувачам на основі їхніх величезних фонових алгоритмів.

1.8 Майбутні напрямки РС

На сьогоднішній день майже всі РС були розроблені для продавців, виробників і сервісу провайдерів, тобто вони призначені для залучення потенційних клієнтів. В майбутньому РС будуть не тільки обмежені бізнесом, але й матимуть набагато більший вплив на нас та наше щоденне життя. Ці системи стануть дійсно необхідним інструментом у кожній сфері.

Майбутні RS не будуть прив'язані лише до заявки на купівлю-продаж продукції; скоріше це стане свого роду особистим консультантом, який допоможе в усіх сферах життя, надаючи важливі пропозиції. Ідеальна РС повинна бути схожим на того, хто знає нас краще, ніж ми знаємо себе. Вони повинні відчувати потребу і запропонувати інстинктивно, навіть якщо люди не висловлюються явно. Кілька інших сфер, подібних до Інтернету речі (IoT) (Al-Fuqaha та ін., 2015), Інтернет всього (IoE) (Янг та ін., 2017; Miraz et al., 2015), ШІ, когнітивні обчислення (Pramanik et al., 2018a), афективні обчислення (когнітивна наука та психологія) (Тао і Тан, 2005) тощо, відіграватимуть значну роль у майбутніх РС. Вони застосовуватимуться для більш широкого спектру застосувань і також зможе краще відображати користувачів і продукти, краще розуміючи не лише користувачів а й продукти.

Майбутні RS зітруть межу між пошуком і рекомендаціями. Фактично RS буде невід'ємною частиною майбутньої пошукової системи, яка зможе запропонувати персоналізований пошук. Користувачам дуже важко зрозуміти обґрунтування. РС буде більш відкритими. Якщо люди розуміють схему рекомендацій, вони шукатимуть більше розсудливо. РС буде більш інтуїтивно зрозумілим і постійно покращуватиме якість рекомендації, враховуючи цикли відгуків користувачів із різних джерел. Вони також будуть більш гнучкими, підтримуючи рейтинги за багатьма критеріями.

Майбутні РС придумують інноваційні моделі рекомендацій з використанням навчання з підкріпленням або розширенням нейронної мережі, яка дозволить їм бути точно враховувати контекст, час і настрій.

Висновки до розділу

У першому розділі було розглянуто поняття машинного навчання, рекомендаційної системи та алгоритми підбору. Також була розглянута та проаналізована фільтрація та її види. Наведено та обґрунтовано поняття *item\user-based* систем та розглянута єдина рекомендаційна модель. Для роботи вирішено взяти алгоритм фільтрації вмісту (*item-based*).

РОЗДІЛ 2. ПОНЯТТЯ МУЗИЧНОГО АУДІО-СТРИМІНГУ ТА АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ

2.1 Поняття музичного аудіо-стрімінгу

Аудіо-стрімінг – це технологія, яка дозволяє безперервно передавати аудіофайли, такі як музика, подкасти, голос за кадром, лекції тощо, через Інтернет, – це потокове аудіо. Основна перевага потокового аудіо полягає в можливості слухати аудіо без завантаження. Потокове аудіо поміщається в невеликі пакети даних, коли воно надходить, і після початкової буферизації протягом кількох секунд відбувається перехід до відтворення. Оскільки аудіо відтворюється безперервно, на пристрій одночасно надходить більше аудіопакетів, і відтворення аудіо відбувається без будь-яких збоїв.

Існує велика кількість аудіо-стрімінгових сервісів. Всі вони пропонують користувачеві якщо не всі, то більшу частину існуючих музичних композицій. Умовно ці сервіси можна розділити на два типи: радіо та аудіотеки. Відмінність між ними полягає в тому, що в радіо немає простої можливості переключати музичний контент, це можна зробити за допомогою смс або дзвінка і не у всіх радіо сервісах. В аудіотеці користувач сам може обрати, яку композицію він хоче прослухати, далі на основі його вибору він зможе прослуховувати схожі композиції або композиції цього ж автора. Цей функціонал є, майже, в стандартних функціях в кожному музичному сервісі. Всі сервіси пропонують прослуховування онлайн, але деякі з них мають платну підписку, яка надає змогу скачувати музичні композиції собі на смартфон або персональний комп'ютер. До мінусів аудіотеки можна віднести те що всі пісні базуються на основі тих що користувач вже слухав і прослухати нову пісню в іншому жанрі ймовірність дуже мало, на відміну від радіо, де музичний контент може кардинально відрізнятись.

Щоб остаточно завершити опис аудіо-стрімінгу потрібно переглянути приклади сервісів та ознайомитись з їх особливостями, також потрібно

розглянути конкретні приклади музичних сервісів. Існує велика кількість музичних додатків, що дозволяють слухати музичний контент і аналізують поведінку користувача та треки. Наприклад Google Play Music, Apple Music, Spotify, YouTube Music та подібних до них. Нижче буде розглянуто не всі, а лише декілька з них, бо всі вони хоч і є різними продуктами різних компаній але функціонал у них дуже схожий.

2.2 Pandora Internet Radio

Pandora – це програма для потокового передавання та пошуку музики. Це дозволяє створювати «станції» на основі пісень, виконавців або жанрів, які подобаються користувачеві, які потім відтворюють пов'язані пісні. Для кожної станції можна вибрати пісні, які подобаються, а які ні, що змусить Pandora (рис 2.1) відтворювати більше пісень, схожих на ті, які подобаються користувачу.

Pandora для рекомендацій музичного контенту використовує Music Genome Project. Він приписує приблизно 450 атрибутів до кожної музичної композиції. Пізніше Music Genome Project був доповнений системою рекомендації на основі пісень які сподобались користувачеві. Використовуючи кнопки Like/Dislike користувач може залишити відгук, це вплине на пісні які будуть відтворюватись користувачу. До мінусів цього веб-сервісу можна віднести доступність, сервіс працює тільки в США.

Для створення рекомендацій Pandora Internet Radio використовує такі типи моделей:

- Аудіо-моделі, які аналізують музичні характеристики такі як частоти, використання електрогітар та інші.
- Моделі колаборативної фільтрації, які аналізують поведінку групи, до якої відноситься користувач або одного користувача.

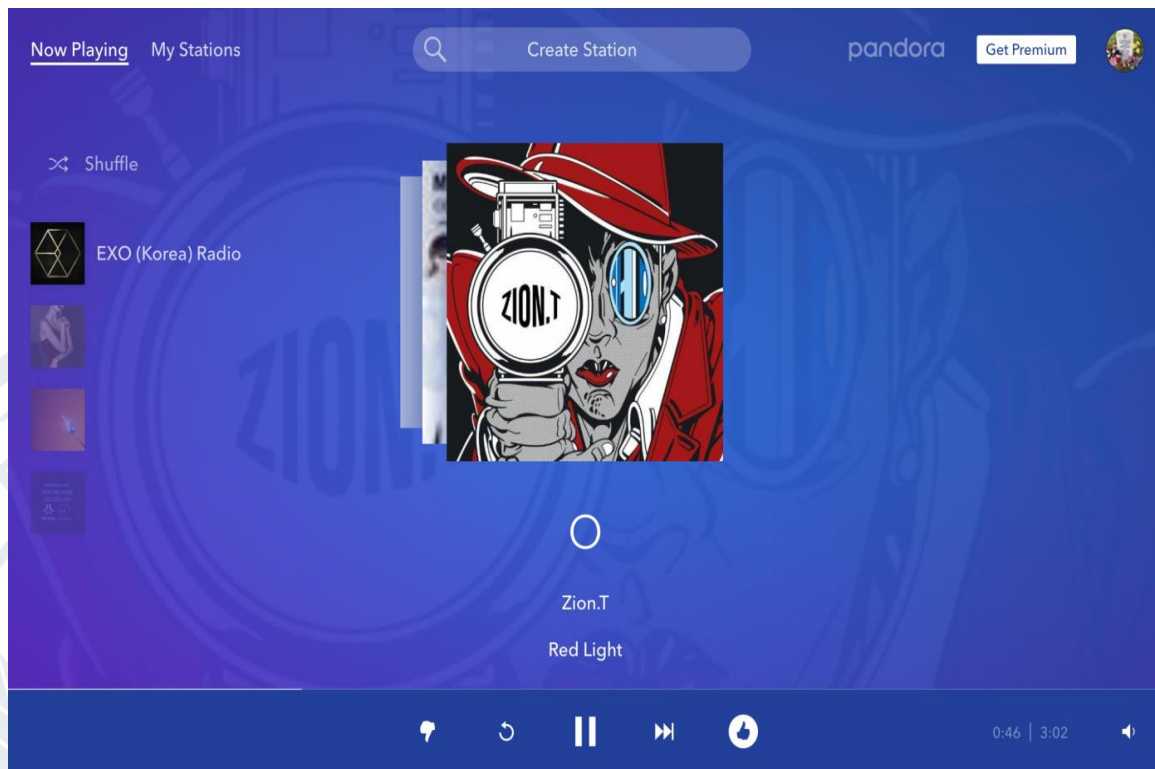


Рис. 2.1. Веб-сайт Pandora Internet Radio

2.3 Spotify

Spotify — інтернет-сервіс потокового аудіо, за допомогою якого можна безкоштовно слухати музичний контент від різних авторів. Також Spotify (рис 2.2) надає законний спосіб транслювати музику онлайн і слухати подкасти. Компанія ліцензує треки від великих і другорядних лейблів для своєї великої музичної бібліотеки. Він виплачує правласникам нерозголошену суму залежно від кількості прослуховувань кожного треку.

Для створення рекомендацій Spotify використовує такі типи моделей:

- Моделі колаборативної фільтрації, які аналізують поведінку групи, до якої відноситься користувач або одного користувача.
- Моделі обробки природної мови, які аналізують текст музичної композиції.
- Аудіо-моделі, які аналізують музичні характеристики такі як частоти, використання електрогітар та інші.

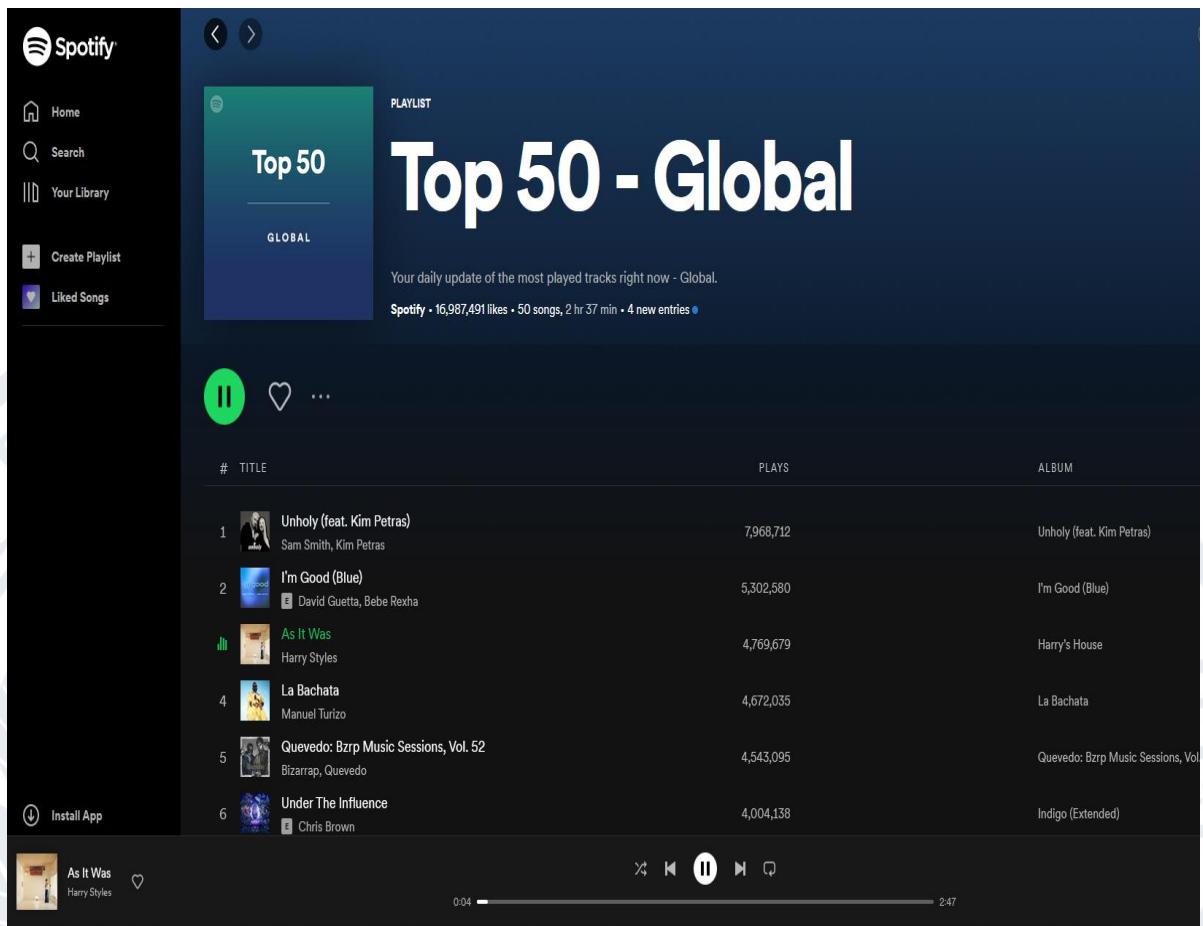


Рис. 2.2 Веб-додаток Spotify

2.4 YouTube Music

YouTube Music (Рис. 2.3) - інтернет-сервіс потокового аудіо від компанії Google. Цей музичний сервіс надає можливість безкоштовно прослуховувати музику, але з деякими обмеженнями. В безкоштовній версії присутня реклама між музичними композиціями. Також до недоліків можна віднести якість звуку, яка нижче ніж в конкурентів.

Для створення рекомендацій YouTube Music використовує такі типи моделей:

- Моделі колаборативної фільтрації, які аналізують поведінку групи, до якої відноситься користувач або одного користувача.
- Моделі обробки природної мови, які аналізують текст музичної композиції.
- Аудіо-моделі, які аналізують музичні характеристики такі як частоти, використання електрогітар та інші.

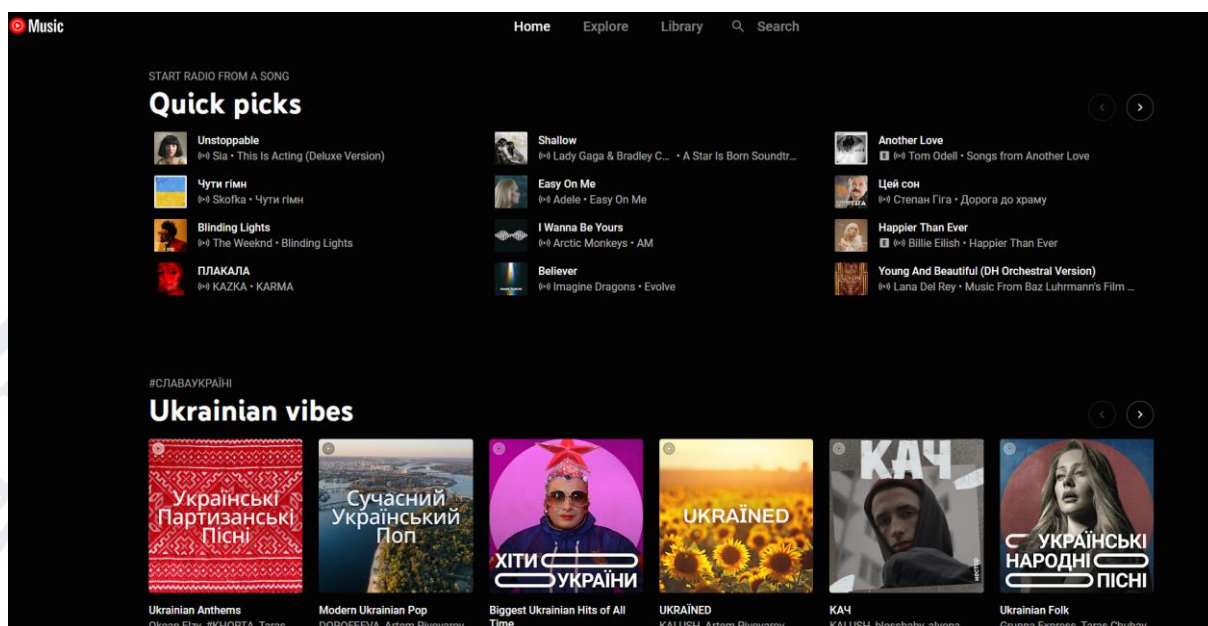


Рис. 2.3 Веб-додаток YouTube Music

2.5 Визначення параметрів оцінювання музики

Властивості звуку залежать переважно від частоти коливань, отже, будемо використовувати саме частотні показники для аналізу музичних композицій. Частота – це швидкість вібрації, яка визначає висоту звуку. Це корисно або має значення лише для музичних звуків, де є чітко регулярна форма хвилі. Частота вимірюється як кількість хвильових циклів, які відбуваються за одну секунду. Одиницею вимірювання частоти є Герц (скорочено Гц). Звуки, що мають меншу частоту, називаються інфразвуком, а мають більшу частоту – ультразвуком. При цьому інфразвукові коливання хоч і не можуть бути почуті, але можуть уловлюватися відчутно. Більшість звуків, що використовуються в музиці, мають частоти, укладаються у межі від субконтрактави до п'ятої октави. Наприклад, звучання звичайного 88-клавішного фортепіано має діапазон від ноти «Ля» субконтрактави (приблизно 27 Гц) до ноти «До» п'ятої октави (4200 Гц). Також, не варто забувати, що весь спектр музичного звуку, як правило, включає не тільки чисті звуки основної частоти, а й гармонію звуків з частотами, які кратні основній частоті, а обертони музичних звуків розташовані

у всьому діапазоні доступних для людського слуху частот. Ґрунтуючись на цих даних, можна виділити кілька діапазонів частот, що уособлюють основні параметри кожної музичної композиції:

1. Низькі басы (10-80 Гц) - це найнижчі ноти, від яких резонує кімната, а драти починають гудіти. Якщо знехтувати цими звуками то музика буде здаватися ненасиченою, а звук втратить свою глибину.
2. Верхні басы (80–200 Гц) – це верхні ноти басових інструментів та найнижчі ноти таких інструментів, як гітара. Якщо втратити цей регістр, то разом із ним загубиться і відчуття сили звуку. Нехтування цим діапазоном частот призведе до втрати відчуття сили звуку.
3. Низькі середні (200 – 500 Гц) – тут розміщується майже весь ритм та акомпанемент, це регістр гітари.
4. Середні середні (500 – 2500 Гц) – соло скрипок, соло гітар, фортепіано, вокал. Музику, в якій не вистачає цих частот, зазвичай називають "занудною" або "похмурою".
5. Верхні середні (2,5 – 5 кГц) – у цьому діапазоні мало нот, тільки найвищі ноти фортепіано та деяких інших інструментів, тут багато гармоніки та обертонів. Посилення цієї частини спектру дозволяє досягти яскравого звуку, що іскристий, що створює ефект присутності.
6. Низькі високі (5 – 10 кГц) – тут зустрічається з найсильніше спотворенням високих частот і де шипіння плівки (для любителів касетного запису) стає найпомітнішим, тому що тут дуже мало інших звуків, здатних приховати це. Хоча люди, які теоретично можуть чути і вищі тони, ці частоти вважаються межею сприйняття..
7. Верхні високі (10 – 20 кГц) – остання октава, це найтонші та найніжніші високі частоти. Якщо цей діапазон частот буде неповноцінний, то людина відчує дискомфорт при прослуховуванні записів.

З погляду рекомендаційних систем, які використовують методи фільтрації вмісту, номери частот будуть виступати як ключові слова в алгоритмі підбору, а значення сили звуку для кожної конкретної частоти – вагами ключових слів.

Таким чином, загальна кількість ключових слів $K_{i,s}$ у композиції s дорівнюватиме кількості частот F_i , помноженому на тривалість програвання t :

$$K_{i,s} = F_i \times t. \quad (2.1)$$

Оскільки число ключових слів, що виходить, вкрай велике, слід трохи узагальнити значення частот і позбутися повторень. Для цього знайдемо середні значення w_f частот протягом всієї композиції, рівні сумі всіх значень $w_{i,j}$ конкретної частоти f_j , поділене на тривалість програвання:

$$w_f = \frac{\sum_{i=1}^t w_{i,j}}{t}. \quad (2.2)$$

Тепер маємо близько 20 тисяч значень, які можемо згрупувати за описаними вище діапазонами D і отримати загальне значення параметра w_d кожного з них. Знайдемо суму значень w_f частот у кожному діапазоні і розділимо результат, що вийшов, на число частот $f_{i,d}$ в конкретному діапазоні.

$$w_d = \frac{\sum_{i=1}^{f_{i,d}} w_f}{f_{i,d}}. \quad (2.3)$$

Отримані значення параметрів можна вважати узагальненими для оцінювання музики та буде застосовувати їх у алгоритмі підбору для виявлення схожості звуку між композиціями.

2.6 Визначення вимог до системи

На підставі вищенаведених вивчення рекомендаційних систем, аналізу предметної галузі та оцінки аналогів було сформульовано визначення веб-додатку музичного аудіо-стрімінгу з рекомендаційною системою, а також виявлено функціональні та нефункціональні вимоги до системи та складено варіанти використання. Веб-додаток музичного аудіо-стрімінгу з рекомендаційною системою – це розташований у мережі Інтернет сервіс для кінцевого користувача, який відтворює музичні композиції послідовно, змінюючи їх після закінчення прослуховування відповідно до деякого алгоритму підбору рекомендованої композиції.

Функціональні вимоги до системи:

1. Система надає користувачеві можливість слухати музику, запуск композиції здійснюється після пошуку на старті програми .
2. Система підтримує стандартний функціонал аудіо програвача, такий як Redirect/Play , зміна пошукових рекомендацій, доступно фільтри, Виконавець\Жанр\Пісня. Є можливість переходу на платформу YouTube для прослуховування композицій, також є утиліта яка надає змогу зробити поширену фільтрацію по пошуковому матеріалу.
3. Функція пошуку, виконання якої призводить до зміни композиції.
4. Зміна композиції, внаслідок якої система підбирає та запускає відтворення нової композиції, вибраної відповідно до результату роботи алгоритму підбору рекомендацій.

Нефункціональні вимоги до системи:

1. Аналіз музичної композиції ґрунтується на частоті звукових коливань. Параметрами є узагальнені значення сили звуку на певних діапазонах частот.

2. Параметри композиції визначаються на момент пошуку та обрання композиції у медіатеці.

3. Робота алгоритму підбору здійснюється під час пошуку і обрання музичної композиції, при зміні композиції відтворюється найбільш підходящий підібраний варіант.

2.7 Аналіз алгоритму та систем фільтрація

Алгоритм колаборативної фільтрації на основі користувачів вважає, що користувачі зі схожими інтересами та хобі мають однакові оцінки більшості інших елементів. Під час рекомендацій цільовим користувачам, як довідкова інформація, використовується рейтингова інформація сусідніх користувачів із подібними захопленнями. Процес рекомендації можна розділити на три етапи: оцінка опису, пошук найближчого сусіда та прогнозування оцінки та створення списку рекомендацій.

Опис рейтингу

Оцінку товару користувача зазвичай можна виразити за допомогою матриці порядку ixr . Серед них u представляє кількість користувачів, p представляє кількість елементів, а елемент t в i -му рядку та стовпці J представляє оцінку елемента J користувачем i , що відображає перевагу користувача щодо елемента, як показано в таблиці 1.

	Project 1	Project j	Project P
User 1	$t_{1,1}$	$t_{1,j}$	$t_{1,P}$
User i	$t_{i,1}$	$t_{i,j}$	$t_{i,P}$
User u	$t_{u,1}$	$t_{u,j}$	$t_{u,P}$

Таблиця 1. Матриця оцінки проекту користувача.

Пошук найближчих сусідів

На цьому етапі шляхом обчислення подібності між користувачами користувачі *Top-N* з найбільшою подібністю вибираються як набір найближчих сусідів цільового користувача. Пошук найближчого сусіда є ключем до алгоритму *CF* на основі користувача. Зазвичай використовувані користувачем методи обчислення подібності включають в себе косинусну подібність, кореляційну подібність (також звану кореляцією коефіцієнта Пірсона) і модифіковану косинусну подібність.

Прогнозування оцінки та створення рекомендації

Після визначення найближчого сусіднього набору, наступні можуть давати рекомендації на основі оцінок користувачів набору. Результат рекомендації колаборативної фільтрації включає прогнозоване користувачем значення балів для елементів без оцінки та набору найпопулярніших елементів. Результати багатьох попередніх досліджень приділяють більше уваги точності передбачень і вважають, що чим ближче прогнозоване значення балу до фактичного балу, тим вища якість системних рекомендацій. Однак низька помилка передбачення не може гарантувати хорошу якість рекомендації, і іноді набір рекомендацій *Top-N* є більш значущим для системи рекомендацій.

Припустімо, що набір предметів, спільно оцінений користувачем i та користувачем j , $\in I$, тоді можна використати кореляцію Пірсона, щоб отримати подібність між $sim(i, j)$

$$sim(i, j) = \frac{\sum (E_{i,n} - \bar{E}_i)(E_{j,n} - \bar{E}_j)}{\sqrt{\sum (E_{i,n} - \bar{E}_i)(E_{j,n} - \bar{E}_j)^2}} \quad (1)$$

Прогнозована оцінка користувача для елемента без оцінки: припускаючи, що $N(n) = (n_1 \dots n_l)$ це набір найближчих сусідів цільового користувача, $S_{n,j}$ прогнозована оцінка користувача u для елемента без оцінки i може бути виражена як:

$$S_{n,j} = \frac{\sum (t_{v_i} - t_j) sim(n, v)}{\sum sim(n, v)} + \bar{t}_n \quad (2)$$

Коефіцієнт використовується для вимірювання ступеня перекриття двійкових даних, який визначається наступним чином:

$$sim(i, j) = \frac{|E_i \cap E_j|}{|E_i \cup E_j|} \quad (3)$$

Алгоритм визначає ключове слово K , а важливість w для тексту D можна виразити такою формулою:

$$w_{i,j} = \frac{e_i}{\max e_{ij}} \log \frac{I}{e_i} \quad (4)$$

Прогноз популярності на основі вдосконаленої колаборативної фільтрації та персоналізованих музичних рекомендацій

Алгоритм персоналізованої музичної рекомендації на основі популярності базується на музичній рекомендації. Він вивчає проблему персональної персоналізованої музичної рекомендації для користувачів персоналізованої музичної рекомендації, а також, поєднує технологію рекомендацій на основі вмісту, множинну лінійну регресію популярності та байєвську систему. Технологія прийняття рішень повністю інтегрує контекстну інформацію в персоналізовану музичну рекомендацію, що не тільки вирішує проблему неможливості рекомендувати нові цікаві місця для користувачів у технології рекомендацій на основі вмісту, але й однобічність контекстної інформації в традиційних рекомендаціях технології. Алгоритм процесу такий:

- (1) Збирання контекстної інформації користувача в музичних рекомендаціях і класифікація контекстної інформації про налаштування користувача, середовище та контекстну інформацію про пристрій.
- (2) Контекстна інформація про переваги користувача та персоналізований список музичних ресурсів, такі як вхідні дані, використовуючи технологію рекомендацій на основі вмісту для виведення набору рекомендацій щодо ресурсів $R1$, який цікавить користувача.
- (3) Значення статистики популярності персоналізованого музичного ресурсу на стороні сервера використовується як вхідне, використовуючи технологію множинної лінійної регресії для обчислення прогнозованого значення популярності елемента персоналізованого музичного ресурсу та вибору персоналізованого музичного ресурсу з більшим прогнозованим значенням популярності та результатом ресурс рекомендує колекцію Rz .
- (4) На основі набору R і Rz обчисліть новий набір ресурсів $R3$, який користувач персоналізованої музичної рекомендації не виявив.
- (5) Встановлення $R3$ та контекстної інформації навколишнього середовища як вхідні дані та використання технологію прийняття рішень, щоб отримати персоналізований набір рекомендацій музичних ресурсів R .

- (6) Відповідно до інформації про контекст пристрою користувача, ресурси в наборі R рекомендовані користувачеві у форматі, який підтримується пристроєм користувача.

Насправді подібність між користувачами пов'язана не лише з оцінкою продукту, але й із перевагами користувача щодо певного типу товару. Якщо два елементи рейтингу користувачів мають подібні атрибути, можна вважати, що вони також мають значну схожість між користувачами.

Проект може містити декілька категорій атрибутів; наприклад, пісня може бути і в категорії поп, і в категорії чіл. Параметр атрибута цільового користувача для елемента має бути подібним до параметра атрибута сусіда для елемента. Наприклад, якщо в оціночних піснях цільового користувача забагато класичної музики, згенеровані сусідні користувачі, швидше за все, оцінять класичну музику. Таким чином, поєднання подібності атрибутів предметів із традиційним методом обчислення подібності на основі предметів може покращити точність рекомендацій системи рекомендацій.

Основне припущення вдосконаленого алгоритму спільної фільтрації полягає в тому, що користувачі з подібними уподобаннями дадуть однакові оцінки тим самим елементам. Таким чином, після створення найближчого сусіднього набору цільового користувача оцінку цільового користувача для неоцінених елементів можна передбачити на основі оцінки користувача в найближчому сусідньому наборі. На даний момент два широко використовувані методи прогнозування представлені такою формулою:

$$P_{n,i} = \frac{\sum \text{sim}(i, j) \times E_{i,j}}{\sum \text{sim}(i, j)},$$

$$P_{n,j} = \frac{\sum (E_{i,j} - E_i) \times \text{sim}(i, j)}{\sum \text{sim}(i, j)} + \bar{E}_n.$$

(5)

$P_{n,j}$ TT представляє передбачувану оцінку користувача U для проекту I , E представляє найближчий сусідній набір цільового користувача I та представляє середню оцінку користувача I та користувача J за всіма оціненими елементами відповідно.

Декомпозицію одиничних значень можна виконувати в автономному режимі, коли вона використовується для обчислення декомпозиції матриці, і її можна добре поєднувати з технологією спільної фільтрації, щоб полегшити виявлення характеристик користувачів і елементів і виробити рекомендації на цій основі. У реальній системі рекомендацій масивні дані про користувачів і елементи зазвичай призводять до дуже складних моделей прогнозування. У той же час через наявність великої кількості пропущених балів результати прогнозу часто не задовільні. А завдяки зменшенню розмірності можна покращити щільність даних, краще вирішити проблему розрідженості даних і водночас знайти приховану інформацію в даних. Алгоритм колаборативної фільтрації має хорошу застосовність. Для кожного елемента його характеристики можуть бути складені з різних розмірів з різними пропорціями; так само для кожного користувача різні виміри також можна розділити на різні виміри відповідно до інтересів користувача.

Саме так, виходячи з загальнодоступних даних, працюють найвідоміші системи з алгоритмами колаборативної фільтрації. Звісно, кожна система використовує свої технології, або гібрид технологій, які вони не будуть розкривати. Найвідоміші системи, які базуються на даному алгоритмі: Spotify, Pandora, Apple Music.

Відмінність від аналогів

Розроблена система, хоч і базується на алгоритмі фільтрації змісту має свої відмінності від аналогів. Обраний алгоритм, частково, використовується з елементами машинного навчання, тобто він аналізує пошукові композиції за додатковими параметрами, такими як: аналіз частот, коливань, аналіз тексту та інші.

Розроблений гібрид, виконує поставлені цілі, але не може конкурувати з крупними музичними платформами, отже, можна виділити плюси та мінуси.

Плюси:

- 1) Швидкість. Алгоритм, виконує запити, оброблює інформацію, та видає рекомендації дуже швидко.
- 2) Зручність та простота використання. Рекомендаційна система, нативно зрозуміла для використання, має мінімальний, але потужний функціонал, що спрощує роботу з нею.
- 3) Точність. Алгоритм, використовує поєднання бібліотек, та потужних модулів, та інструментів машинного навчання, що збільшує точність видачі рекомендацій, на відмінно, від класичного варіанту.

Мінуси:

- 1) Автономність. Алгоритму потрібен сервіс, з бібліотекою музичних композицій, так як результати рекомендацій базуються її на параметрах.

Висновки до розділу

На підставі вищенаведених вивчення рекомендаційних систем, аналізу предметної галузі та оцінки аналогів було сформульовано визначення веб-додатку музичного аудіо-стрімінгу з рекомендаційною системою, а також виявлено функціональні та нефункціональні вимоги до системи та складено варіанти використання. Наведено та розглянуто системи аудіо-стрімінгу, визначені параметри оцінювання музики. Розглянуто обраний алгоритм та системи які його використовують, встановленні плюси та мінуси системи.

РОЗДІЛ 3. ПОНЯТТЯ ВЕБ-ДОДАТКУ, АРХІТЕКТУРА ВЕБ-ДОДАТКІВ, ОБРАННЯ МОВ ПРОГРАМУВАННЯ, ІНСТРУМЕНТІВ РОЗРОБКИ. ПОБУДОВА ВЕБ-ДОДАТКУ

3.1 Поняття веб-додатку та архітектура веб-додатків

Веб-додаток – це програма, яка зберігається на віддаленому сервері, на яку йдуть запити, через браузер, від користувачів, частіше всього по HTTPs протоколу.

З цього можна зробити висновок, що веб-додаток - це програма яка створена для взаємодії з користувачем, а веб-сайт – це сторінка яка містить статичний контент.

Загалом веб-додатки складаються з двох основних компонентів:

1. Клієнтська частина, яка називається інтерфейсом, де використовується HTML, CSS, JavaScript. Тут відбувається взаємодія користувача з серверною частиною.
2. Серверна частина, контролює бізнес-логіку та відповідає на запити HTTP. Код на стороні сервера може бути написаний на більшості мов програмування C#, Java, PHP, Ruby, Python, JavaScript та інші.

Спростована діаграма взаємодії користувача з веб-додатком наведена нижче:

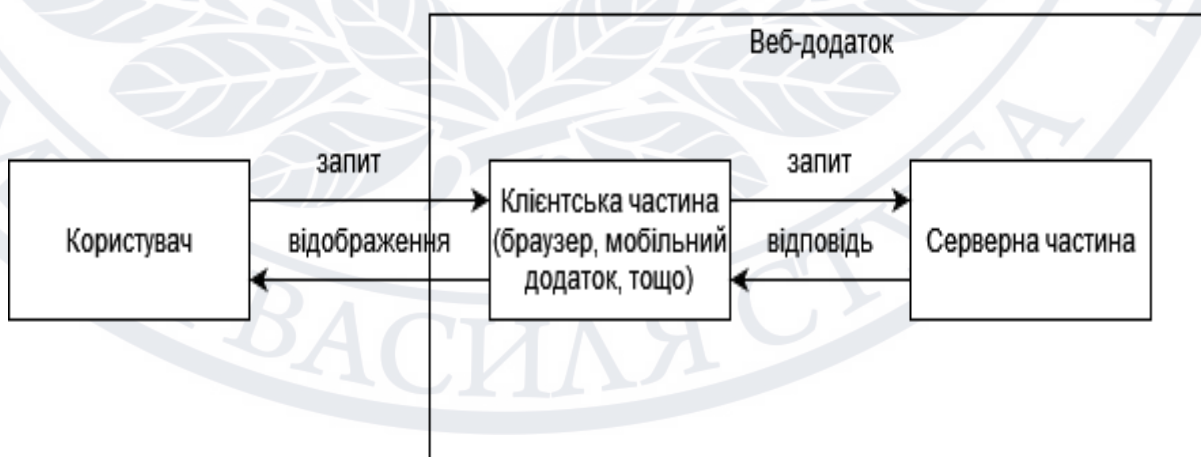


Рис. 3.1 - Діаграма взаємодії користувача з веб-додатком

Архітектура веб-додатків — це схема одночасної взаємодії між усіма компонентами веб-додатку такими як бази даних, проміжні системи, інтерфейси програми.

Сучасні веб-програми мають багаторівневу архітектуру, яка складається з презентаційного, бізнес-рівня, стійкості та бази даних. Невеликі програми мають три рівні, де в деяких випадках бізнес-рівень і стійкість діють як один рівень, тоді як складні програми мають п'ять або шість рівнів.

Рівні архітектури веб-додатків:

1. Презентаційний рівень, створений за допомогою HTML, CSS, JavaScript і його фреймворків, забезпечує зв'язок між інтерфейсом і браузером для полегшення взаємодії з користувачем.
2. Бізнес-рівень, на цьому рівні визначені бізне-логіка додатка. Цей рівень включає веб-сервер, який буде приймати запити від клієнта(браузер, додаток на смартфон) обробляти згідно бізнес-правилам та надсилати відповідь клієнту.
3. Рівень збереження використовується для збереження даних. Цей рівень відповідає за взаємодію з рівнем бази даних.
4. Рівень бази даних, також відомий як рівень служби даних. На цьому рівні частіше використовується SQL-бази даних ніж NoSQL. Приклади SQL баз – PostgreSQL, Microsoft SQL Server, MySQL, NoSQL – MongoDB, Cassandra, Redis

Приклад сучасної архітектури веб-додатків:

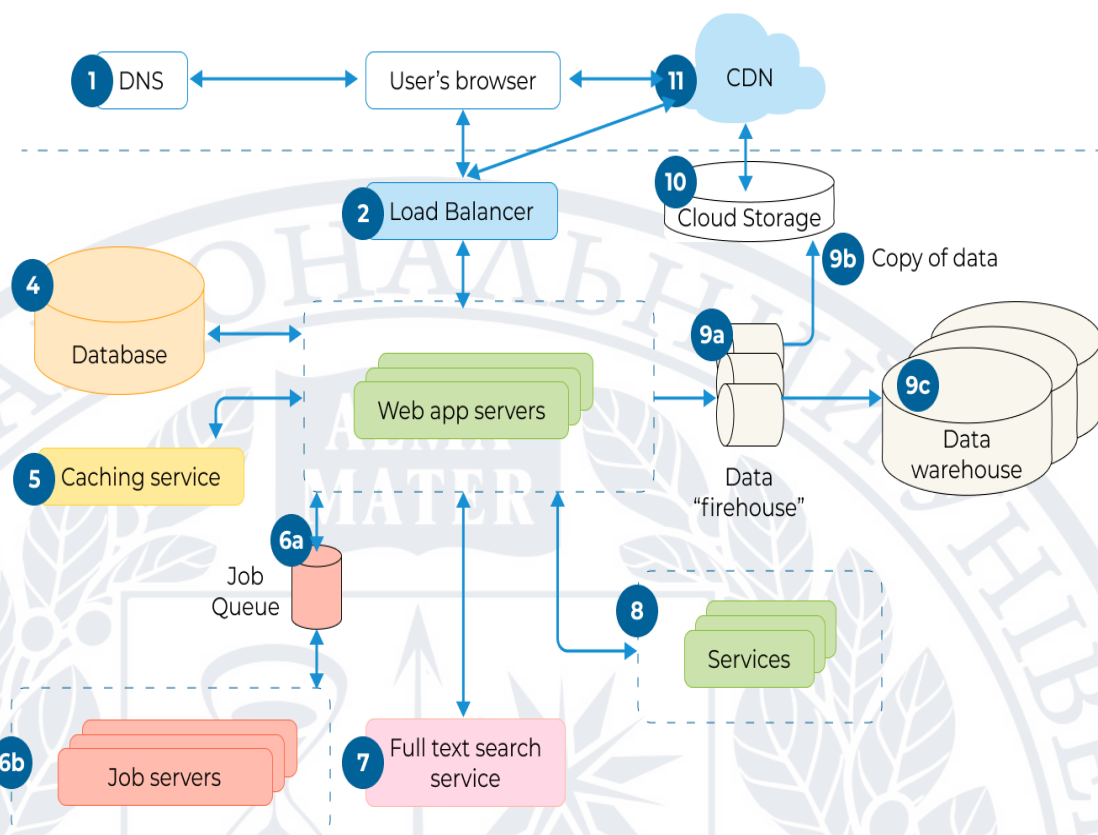


Рис 3.2 – Сучасна архітектура веб-додатку

3.2 Вибір технологій

Для клієнт частини обрано JavaScript оскільки аналогів не існує.

JavaScript – це прототипно-орієнтована, мультипарадигмова динамічна мова програмування. В більшості випадків вона використовується для створення клієнтських частин веб-додатків, також можна писати веб-сервери використовуючи фреймворк Node.js, який надає можливість взаємодіяти з пристроями введення-виведення. Також за допомогою JavaScript можна писати код об'єкто-орієнтованому стилі.

Специфікація створення скриптової мови загального призначення описана в ECMA-262.

В специфікація ECMA-262 визначено стандартну версію основної мови JavaScript.

- JavaScript — це легка інтерпретована мова програмування.
- JavaScript призначена створення веб-додатків.

- JavaScript досить гарно інтегрується з HTML та доповнює його.
- JavaScript є кросплатформенною мовою з відкритим вихідним кодом.

Для серверної частини можна обрати майже любую мову програмування, але потрібно враховувати, те що не в кожній мові є фреймворк для розробки ML (machine learning) моделей.

ML модель – це файл, який навчається розпізнаванню певних типів закономірностей. Модель навчається на основі набору даних, що надає її алгоритм, який вона може використовувати для аналізу та навчання на основі цих даних. Завершивши моделі навчання, можна брати її для прийняття рішень і виконання прогнозів за даними, які раніше не зустрічалися. Наприклад, якщо потрібно створити додаток для розпізнавання емоцій користувача по його вираженню обличчя. Для цього треба навчити модель за набором зображень особи, що з них позначено тегом певної емоції, а потім застосувати цю модель у накладенні для розпізнавання емоцій користувача.

Розберемо декілька мов в яких є ML фреймворк та за допомогою яких можна зручно розробити веб-сервер:

- Python – це найпопулярніша мова для роботи з ML. Для неї написано дуже багато ML фреймворків. Ось декілька з них: TensorFlow, Scikit-Learn, Keras та інші. Фреймворки для веб-розробки: Django, Flask, Hug та інші.
- Java – одна з популярних мов для роботи з ML. Декілька фреймворків: Java-ML, Neuroph, Weka. Фреймворки для веб-розробки: Spring.
- JavaScript – популярна мова для ML розробки. ML фреймворки: TensorFlow.js. Фреймворки для веб-розробки: Node.js

Для розробки серверної частини було обрано мову програмування Python. Оскільки існує багато фреймворків для розробки ML та веб-розробки. З плюсів

такого підходу, коли обрана одна мова для роботи з ML та веб-серверу, що легко можна інтегрувати одну систему в іншу.

Python — це популярна мова програмування на якій можна розробляти веб-додатки, програми для ML, та багато інших користувацьких програм. В ній є високо рівневі структури даних, динамічний тип, динамічне зв'язування. Вихідний код цієї мови можна знайти в інтернеті. Загалом використання мови Python можна знайти в різних програмах різного призначення.

Також ця мова є найпопулярнішою для розробки ML-додатків завдяки великому вибору бібліотек та підтримці, як великих IT-компаній (Google та інші), так і звичайних розробників, так як більшість бібліотек є з відкритим вихідним кодом.

Також для зручного розгортання веб-додатку на віддаленій машині (серверу) було обрано платформу Docker.

Docker — це програма з відкритим кодом для створення, розгортання та керування docker-контейнерами у загальній операційній системі (ОС) із екосистемою суміжних інструментів.

Docker — це і демон (процес, що працює у фоновому режимі), і клієнтська команда. Це як віртуальна машина, але вона відрізняється в важливих аспектах. По-перше, менше дублювання. З кожною додатковою віртуальною машиною, яку ви запускаєте, ви дублюєте віртуалізацію процесора та пам'яті та швидко витрачаєте ресурси під час локальної роботи. Docker чудово підходить для налаштування локального середовища розробки, оскільки він легко додає запущений процес без дублювання віртуалізованого ресурсу. По-друге, він більш модульний. Docker дозволяє легко запускати кілька версій або екземплярів однієї програми конфігурації без конфліктів портів.

На рис. 3.3 наведена внутрішня робота докера.

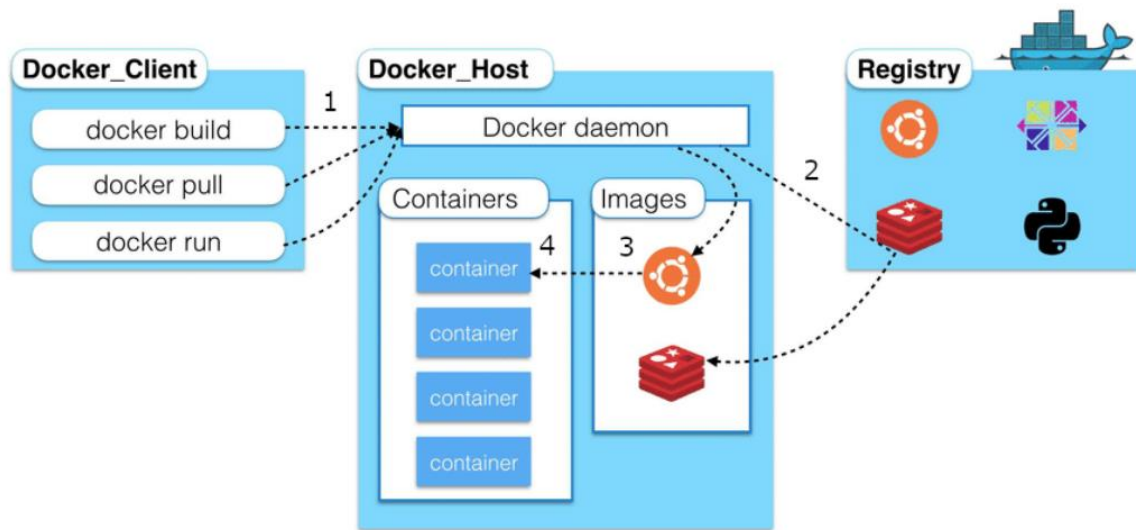


Рис.3.3 - Внутрішня робота докера

3.3 Початок роботи

Для потрібно обрати стек технологій які необхідні для розробки музичної рекомендаційної системи.

Нижче наведені технології за допомогою яких було розроблено веб-додаток.

- python \geq 3.7.0 - Рекомендації до бібліотек визначені у файлі requirements.txt.
- Scikit-learn(sklearn) - бібліотека для машинного навчання, за допомогою неї можна тренувати або створювати алгоритми класифікації, класифікації, кластеризація. Наприклад random forest.
- Flask – веб-фреймворк, який представляє модулі за допомогою який можна створювати веб-додатки при цьому забути про деталі низького рівня, таких як робота з потоками, HTTP/HTTPS протоколи.
- Pandas – бібліотека, за допомогою якої можна використовувати чисельні таблиці, часові ряди і операції пов'язані з роботою цих структур.

- NumPy — ця бібліотека підтримує великі багаторівневі або зубчасті масиви, матриці. Також є підтримка різних математичних операцій на цими даними.

Всі запити з клієнтської частини будуть відбуватись по REST API.

API – це контракт, який надає програма по якому можна звернутись до неї. Наприклад, система Mailgun надає змогу відправляти email-повідомлення. На вхід система очікує запит у вигляді JSON, далі за допомогою цього запиту програма відправляє повідомлення користувачу вказаному в запиті.

Загалом схему роботи API можна зобразити, як показано нижче на рисунку 3.4.

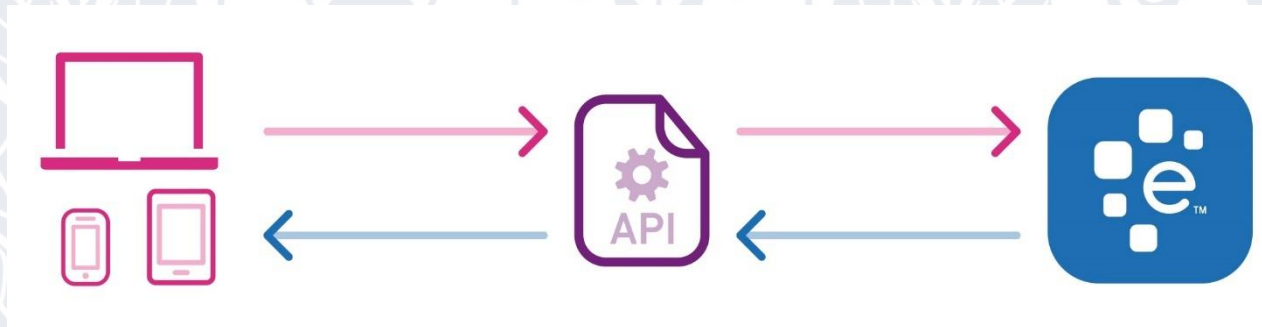


Рис. 3.4 – Схема запиту до системи по API

3.4 Структура проекту

Розробка IT-проекту з впровадження рекомендаційної системи включає наступні етапи:

- 1) Планування - процес формування вимог до функціоналу який має бути в рекомендаційній системі.
- 2) Дизайн(ui\ux) – створення макету майбутнього продукту, по цьому макету буде зроблена клієнтська частина.
- 3) Розробка ML моделі для аналізу та рекомендації музичного контенту на основі запиту користувача.
- 4) Розробка веб-серверу для обробки запиту користувача
- 5) Тестування - перевірка рекомендаційної системи на готовність

б) Підтримка – створення нового функціоналу та підтримка існуючого

3.5 Практична реалізація рекомендаційної системи обраними мовами програмування

Для початку треба знайти датасет з даними про музичні композиції. Існує декілька популярних сервісів де можна знайти такі датасети.

Розглянемо декілька з них:

1. Kaggle - це онлайн-спільнота спеціалістів із обробки даних і практиків машинного навчання. Kaggle дозволяє користувачам знаходити та публікувати набори даних, досліджувати та створювати моделі в веб-середовищі обробки даних, працювати з іншими вченими з обробки даних та інженерами з машинного навчання, а також брати участь у змаганнях для вирішення завдань науки про дані. Всі датасети є безкоштовними, але щоб їх отримати обов'язково потрібна реєстрація.
2. Google Dataset Search - це пошукова система від Google, яка допомагає дослідникам знаходити онлайн-дані, які є у вільному доступі для використання. На відміну від Kaggle, ця система не є повністю безкоштовною. Деякі результати пошуку можуть містити платні датасети.

Для пошуку датасету з музичними композиціями та їх властивостями можна обрати систему Kaggle, так як вона є безкоштовною і в ній є багато датасетів, які можна використати для розробки рекомендаційної системи.

Наприклад датасет від Spotify. Його можна знайти за посиланням: <https://www.kaggle.com/datasets/yamaerenay/spotify-dataset-19212020-600k-tracks?select=tracks.csv>

Цей датасет містить 20 колонок, 13 з яких є атрибутами пісні.

Атрибути пісні: *acousticness*, *danceability*, *duration_ms*, *energy*, *instrumentalness*, *key*, *liveness*, *loudness*, *mode*, *speechiness*, *tempo*, *timesignature*, *valence*.

Аналіз датасету. Для цього потрібно створити кореляційну таблицю. На рисунку 3.4 бачимо, що атрибути: *explicit*, *danceability*, *energy*, *loudness* є важливими для прогнозування популярності, а *acousticness*, *instrumentalness* - ні. З цього можна зробити декілька висновків.

1. *Energy* впливає на популярність пісні. Багато популярних пісень є енергійними. Оскільки кореляція тут не надто висока, пісні з низьким рівнем енергії мають певний потенціал стати більш популярними.
2. *Acousticness* не корелює з популярністю. У більшості популярних сьогодні пісень є або електронні, або електричні інструменти.

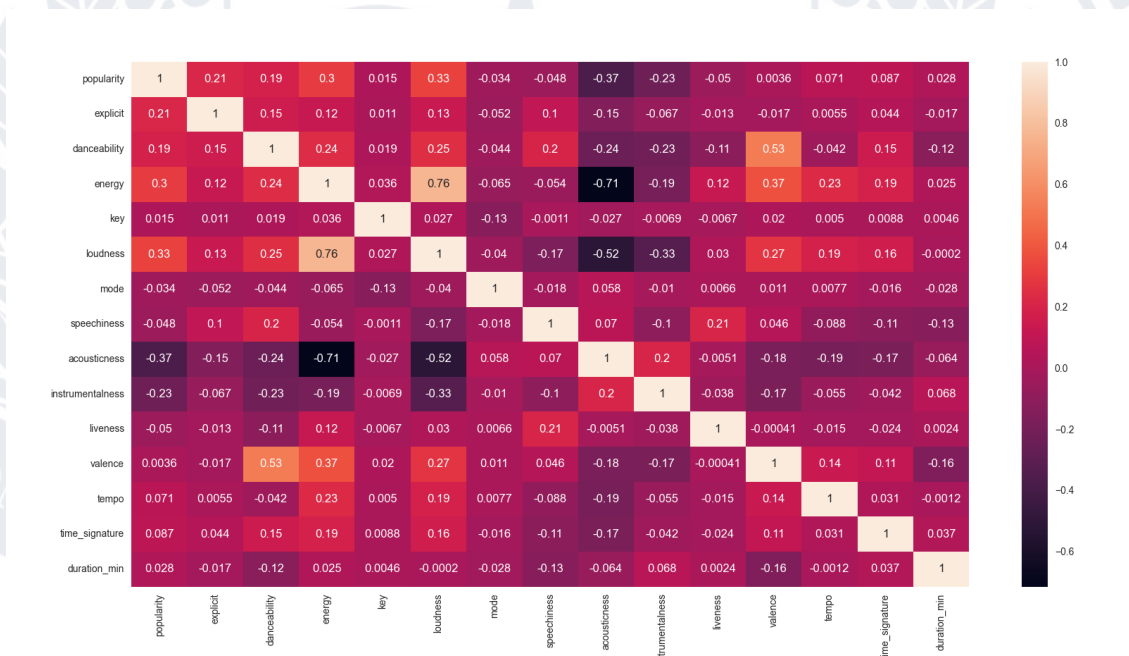


Рис. 3.5 – Кореляційна таблиця

Тепер можна знайти найпопулярніші пісні (рис. 3.6) та найпопулярніших виконавців (рис. 3.7) подивитись наскільки їх характеристики співпадають з тими, що впливають на популярність.

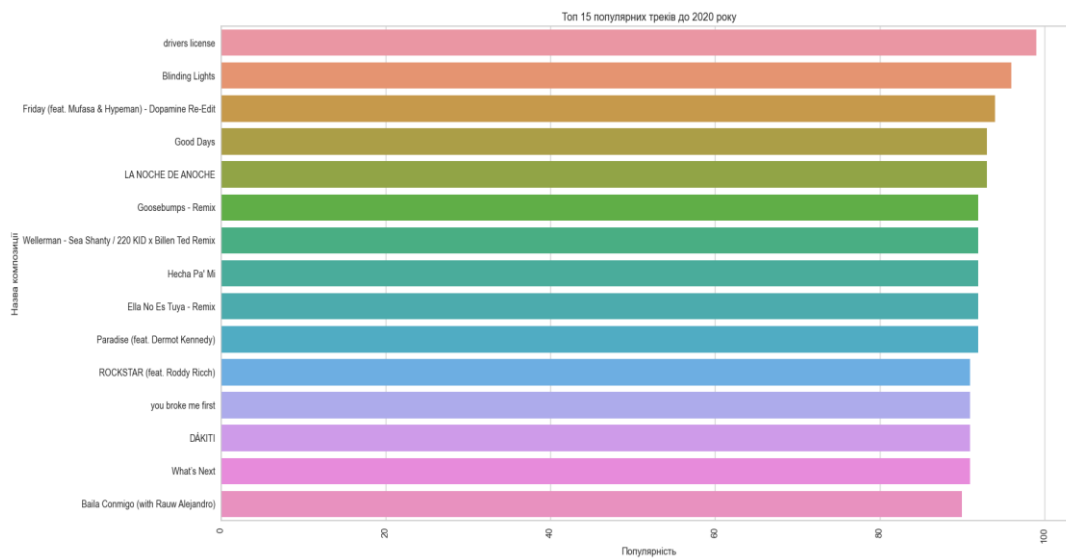


Рис. 3.6 – Топ 20 найпопулярніших пісень в датасеті

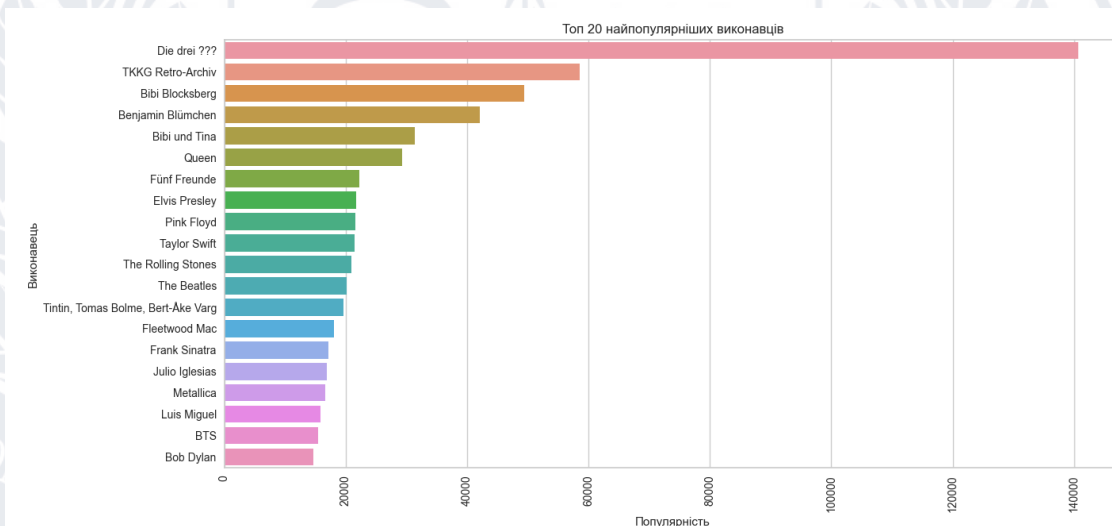
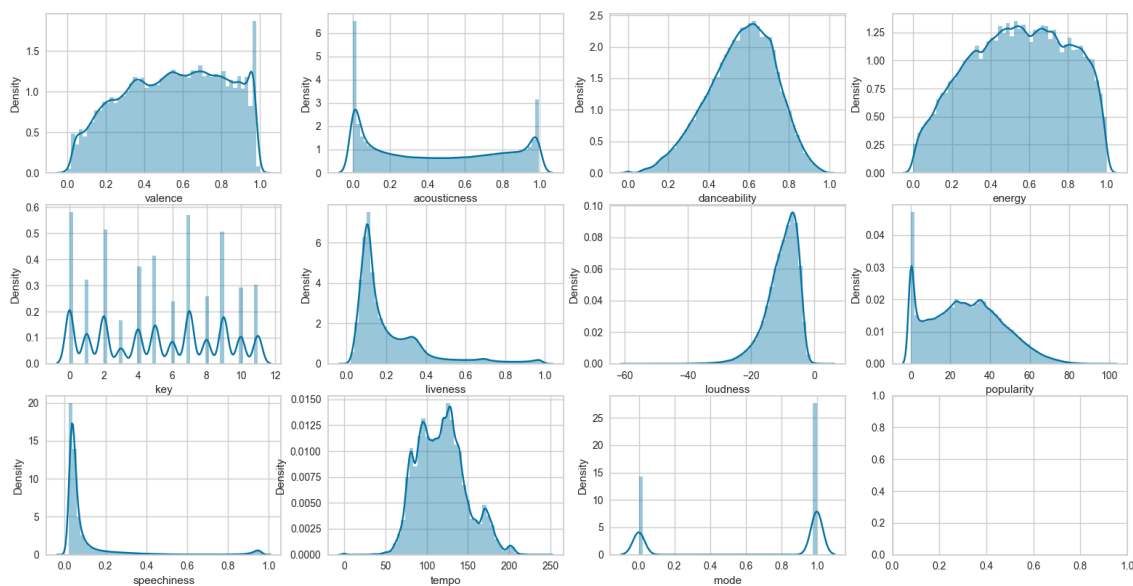


Рис. 3.7 – Топ 20 найпопулярніших виконавців в датасеті

Переглянувши датасет та знайшовши пісні та виконавців, можна зробити висновок, що дійсно популярність залежить від таких атрибутів: explicit, danceability, energy, loudness.

Далі можна зробити візуалізацію кожної характеристики візуалізацію змінності для кожного стовпця в датасеті.



Тепер можна розпочати розробку моделі ML, яка може передбачити популярність на основі доступних функцій.

На рис. 3.8 показані коефіцієнти кореляції Пірсона та взаємна інформація між функціями та залежною змінною. Ця візуалізація може бути використана при виборі ознак для ідентифікації ознак із високою кореляцією або великою взаємною інформацією із залежною змінною.

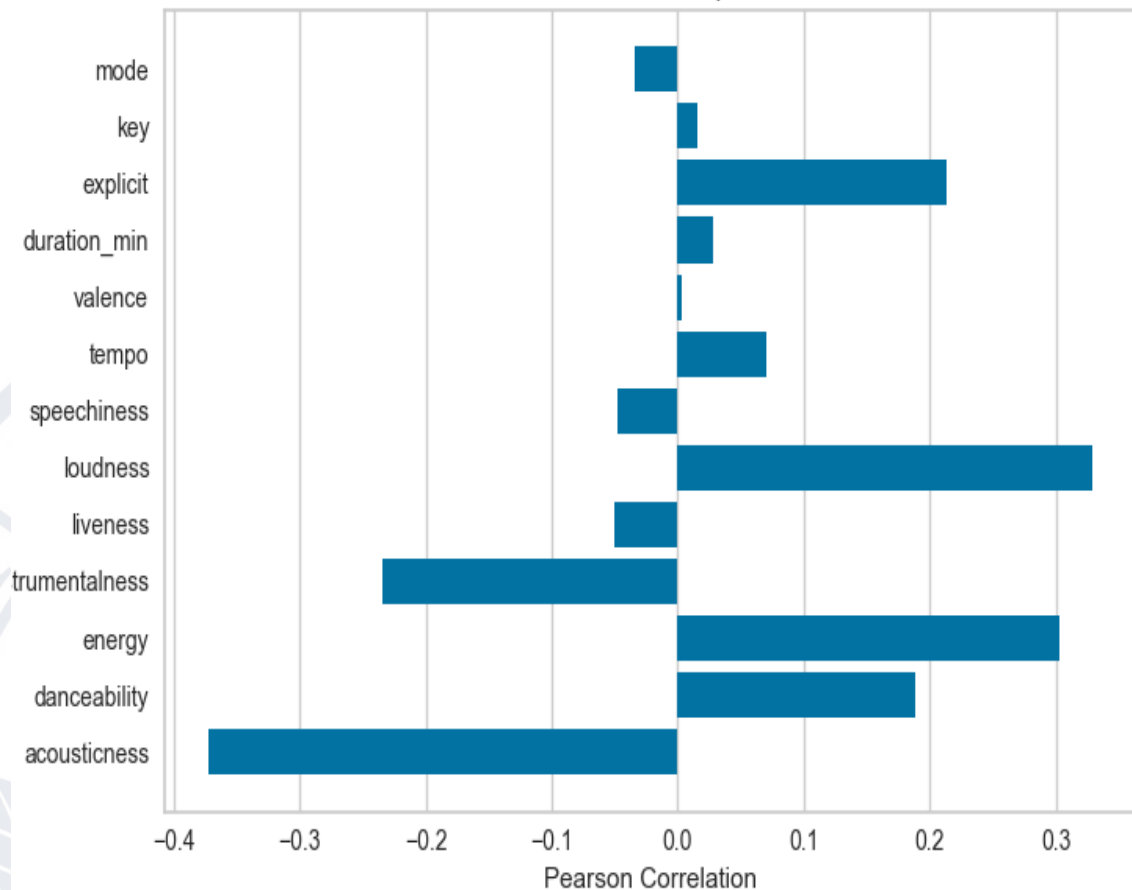


Рис. 3.8 – Коефіцієнти кореляції Пірсона

З наведеного вище графіка видно, що 5 ознак з негативною кореляцією і 8 ознак з позитивною кореляцією.

Вибір ознак, які можна виключити з моделі:

1. `id`: ідентифікатор є унікальним для кожної доріжки, тому не може допомогти моделі.
2. `name`: є 446 475 унікальних значень. Цю категоріальну функцію вставити в модель проблематично, тому її буде видалено.

Перед початком тренування моделі потрібно видалити дублікати з датасету, для цього існує декілька способів. Можна видалити вручну або за допомогою бібліотеки `pandas`.

Програмний код для видалення дублікатів:

```

#зчитування назви колонок
columns = list(pd.read_csv('music.csv', nrows = 1))
#зчитування всього датасету без колонок 'id','name'
dataFrame = pd.read_csv('music.csv', usecols=[i for i in cols if i not in ['id','name']])
#видалення дублікатів
dataFrame = dataFrame[~dataFrame.duplicated()==1]

```

Далі потрібно знайти схожий датасет для тестування або розділити існуючий.

За допомогою наведеного нижче коду можна розділити існуючий датасет

```

X_train, X_test, y_train, y_test = train_test_split(dataFrame.drop('popularity', axis=1), dataFrame
['popularity'], test_size = 0.2, random_state = 42)

```

Потім розробимо модель ML, яка повертає r2-оцінку та середню квадратичну помилку для навчання та тестових наборів даних. Результат на рисунку 3.9, 3.10

```

def Decision_tree (X_train,y_train,X_test,y_test,min_samples_split,max_leaf_nodes):
    tree = DecisionTreeRegressor(max_leaf_nodes = max_leaf_nodes , min_samples_split =
min_samples_split)
    tree.fit(X_train, y_train)
    y_train_pred = tree.predict(X_train)
    train_rmse = np.sqrt(mse(y_train, y_train_pred))
    y_test_pred = tree.predict(X_test)
    test_rmse = np.sqrt(mse(y_test, y_test_pred))
    r2_train = r2_score(y_train, y_train_pred)
    r2_test= r2_score(y_test, y_test_pred)
    mae = (abs(y_test - y_test_pred)).mean()
    return train_rmse,test_rmse,r2_train,r2_test,y_train_pred,y_test_pred,mae

```

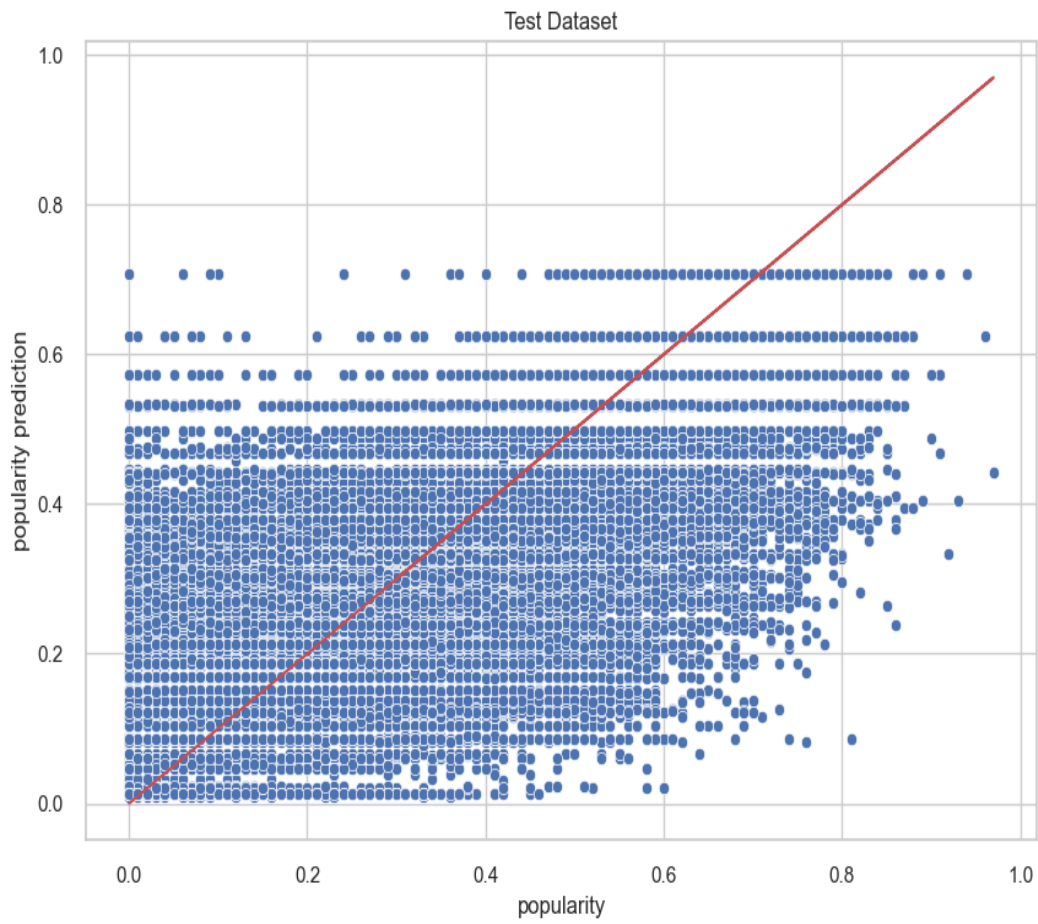


Рис. 3.9 – Середня квадратична помилка для тестових даних

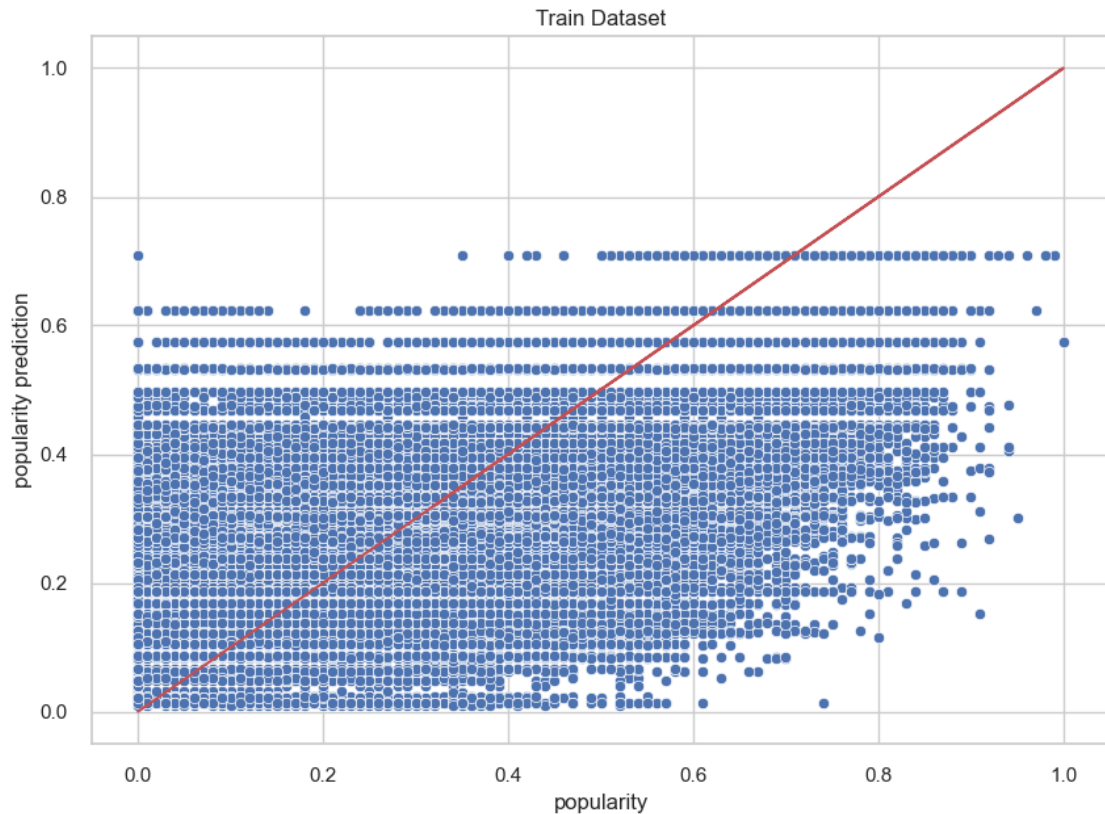


Рис. 3.10 - Середня квадратична помилка для навчальних даних

Тепер потрібно зробити пошук оптимальних гіперпараметрів для дерева рішень для прогнозування популярності пісні.

```
n_features = dataframe.shape[1]
```

```
n_samples = dataframe.shape[0]
```

```
grid = GridSearchCV(DecisionTreeRegressor(random_state=0), cv=3, n_jobs=-1, verbose=5,
    param_grid={
        'max_depth': [None,5,6,7,8,9,10,11],
        'max_features': [None, 'sqrt', 'auto', 'log2', 0.3,0.5,0.7, n_features//2, n_features//3, ],
        'min_samples_split': [2,0.3,0.5, n_samples//2, n_samples//3, n_samples//5],
        'min_samples_leaf':[1, 0.3,0.5, n_samples//2, n_samples//3, n_samples//5]}
    )
```

Далі потрібно розрахувати Мангеттенську дистанцію для всіх пісень на основі заданої.

Мангеттенська метрика d_1 між двома векторами \mathbf{p} , \mathbf{q} в n -вимірному дійсному просторі з заданою прямокутною системою координат — сума довжин проєкцій відрізка між точками на осі координат.

$$d_1(\mathbf{p}, \mathbf{q}) = \|\mathbf{p} - \mathbf{q}\|_1 = \sum_{i=1}^n |p_i - q_i|,$$

де,

$\mathbf{p} = (p_1, p_2, \dots, p_n)$ і $\mathbf{q} = (q_1, q_2, \dots, q_n)$ - вектори

```
def get_recommendations(self, song_name, count):
    distances = []
    song = self.data_[(self.data_.name.str.lower() == song_name.lower())].head(1).values[0]
    rem_data = self.data_[self.data_.name.str.lower() != song_name.lower()]
    for r_song in tqdm(rem_data.values):
        dist = 0
        for col in np.arange(len(rem_data.columns)):
            if not col in [3,7,13]:
                dist = dist + np.absolute(float(song[col]) - float(r_song[col]))
        distances.append(dist)
    rem_data['distance'] = distances
    rem_data = rem_data.sort_values('distance')
    columns = ['artists', 'name']
    return rem_data[columns][:count]
```

3.6 Результат роботи рекомендаційної системи

Для тестування роботи проєктованої системи, потрібно відкрити веб-додаток та спробувати знайти якусь музичну композицію:

Для тестування додатку потрібно ввести існуючу назву пісні, наприклад Nirvana – “Smells like teen spirit”

MUSIC RECOMMENDER

Smells like teen spirit Track Search

#	Track	Artist
1	Even Flow	Pearl Jam
2	I Hate Myself And Want To Die - 2013 Mix	Nirvana
3	Coming For You	The Offspring
4	Pretend We're Dead	L7
5	Hitchin' a Ride	Green Day
6	Knights of Cydonia	Muse
7	What Do They Know?	Mindless Self Indulgence
8	B.Y.O.B.	System Of A Down
9	Say It Ain't So	Weezer
10	Just A Girl	No Doubt
11	Learn to Fly	Foo Fighters
12	Ace of Spades	Motörhead

Рис 3.11 Результат роботи рекомендаційної системи

Отже, як можна побачити, після пошуку композиції, нам одразу видаються рекомендації, так як Nirvana – рок гурт, основні рекомендації це рок пісні, є вибірка з 20 композицій схожих на цю, та одразу видно автора чи гурт, які виконали ці композиції.

Виберемо одну з рекомендованих пісень і перейдемо за посиланням:

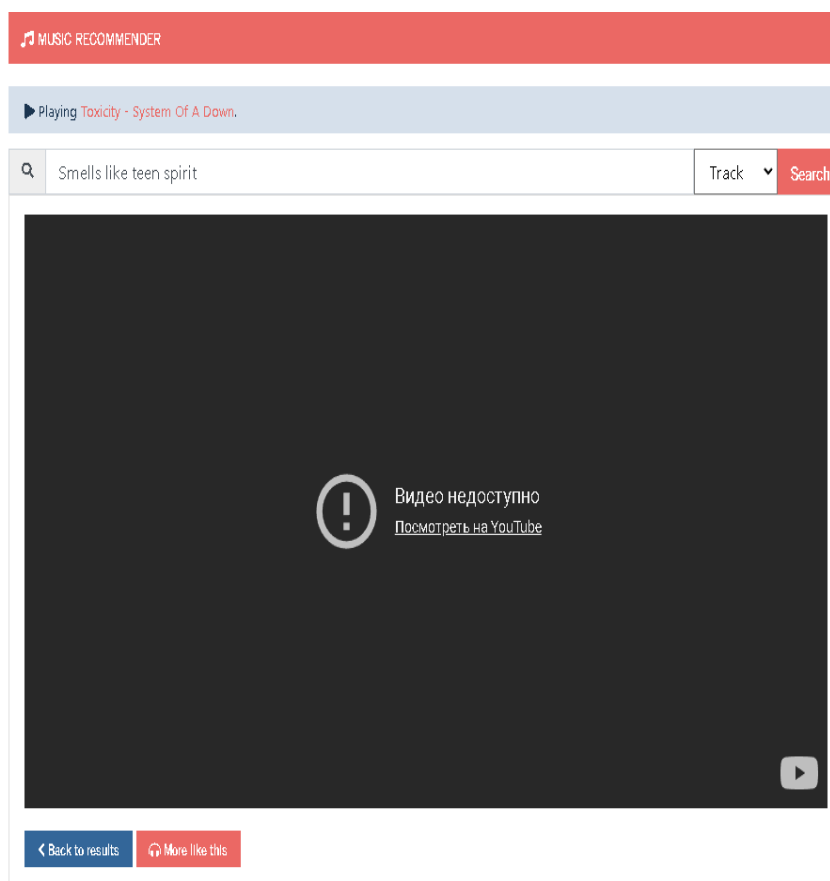


Рис 3.12 Результат переходу за посиланням

Можна побачити у пошуковій формі запит з нашою піснею, також над нею є обраний варіант з рекомендацій, нижче також видніється дві активні кнопки: «Повернутися до результатів пошуку» та «Більше таких пісень»

Також є можливість перейти до платформи YouTube.com натиснувши «Подивитися на YouTube» і прослухати обрану рекомендацію:

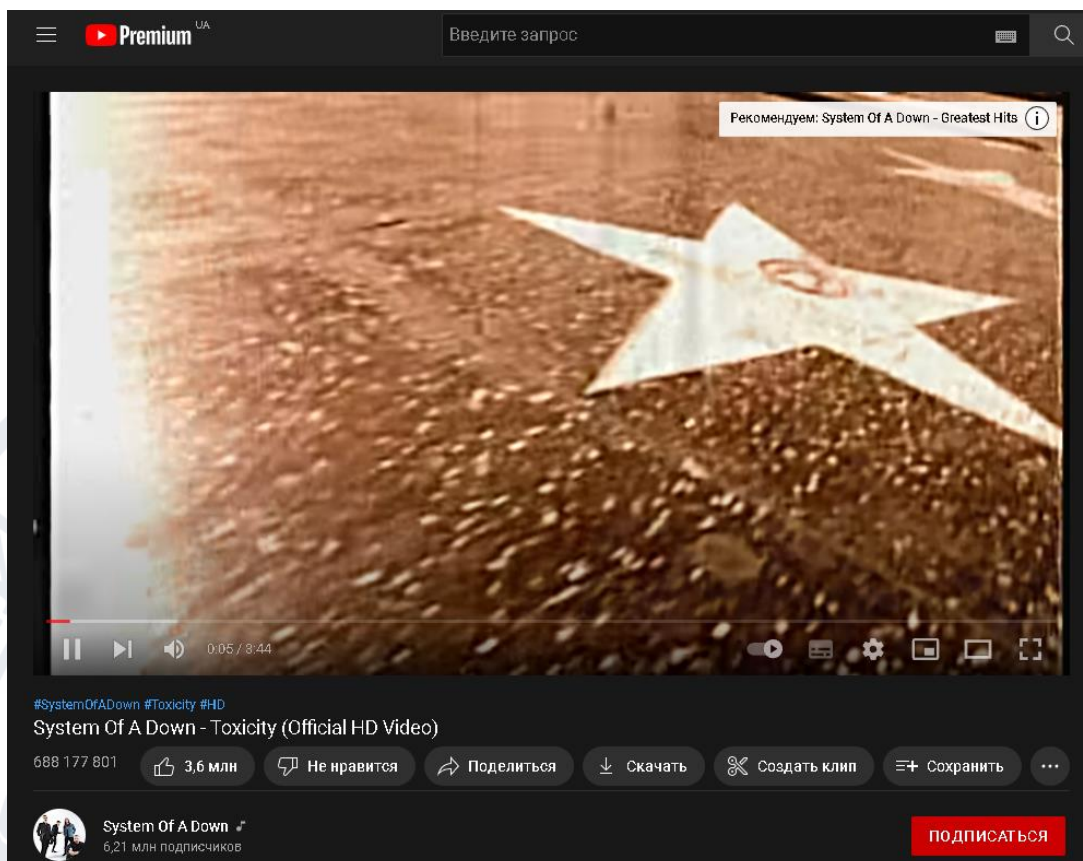


Рис. 3.13 Перехід за обраною рекомендаційною піснюю

Чи скористатися опціями які запропоновані нам системою:

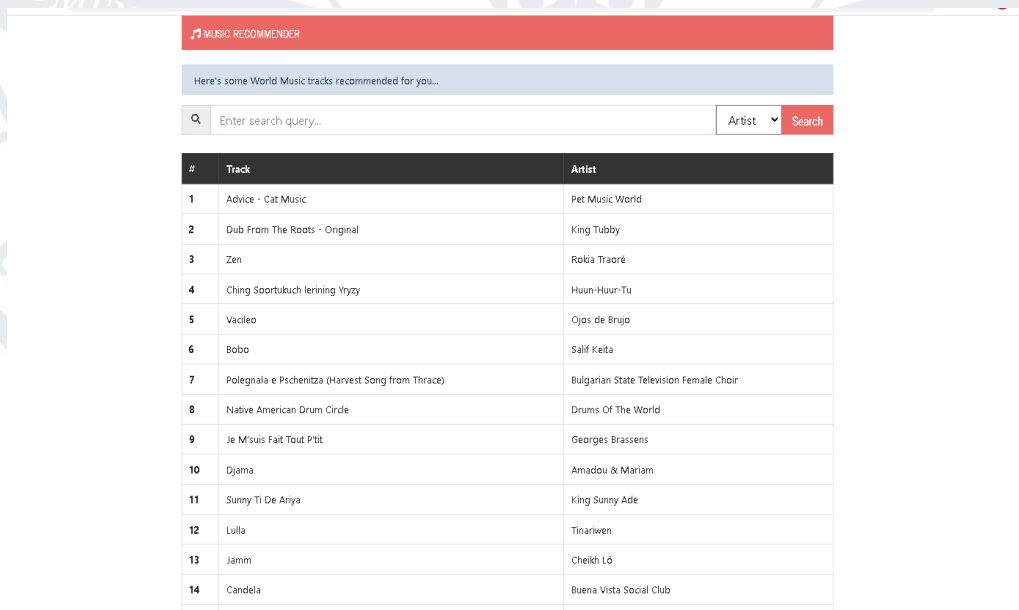


Рис. 3.14 Кнопка «Більше таких рекомендацій»

Висновки до розділу

Розроблено діаграми ризиків та процесів для проектування та програмування системи. Для реалізації даного проекту було обрано алгоритм фільтрування вмісту, цей алгоритм дає прогнози на основі схожих даних, які користувачу сподобались в минулому.

Після реалізації системи обраними мовою програмування та середовищем, також було проведено ряд тестів які підтверджують працездатність програми та відповідають всім функціональним і нефункціональним вимогам.

Оскільки вся система відповідає затвердженим критеріям, то веб-додаток можна розгорнути на сервері, наприклад в хмарній платформі AWS або Azure, щоб користувачі мали доступ до веб-додатку через браузер.

ВИСНОВОК

У магістерській роботі було розроблено веб-сайт для рекомендації музичного контенту. Розробка велась в середовищі VS Code на мові програмування Python та JavaScript з використанням датасету музичних композицій з сайту Kaggle. Даний веб-додаток полегшує знаходження музичних композицій схожих на ті що введе користувач. Також веб-додаток має функції звичайного плеєру, такі як: Start/Stop, Previous/Next, Redirect to.

У першому розділі було досліджено:

- існуючі алгоритми для рекомендації контенту(item/user-based), виявлено їх переваги та недоліки
- підходи щодо створення рекомендаційних систем на основі алгоритму колаборативної фільтрації та фільтрації вмісту;

Також у першому розділі було розглянуто поняття машинного навчання, рекомендаційної системи та обрано алгоритм за допомогою якого буде розроблено веб-додаток.

У другому розділі наведено поняття рекомендаційної системи, Оглянуто аналоги(Pandora, Spotify, YouTube Music), виявлено їх переваги та недоліки. Запропоновано модифікація алгоритму, що використовує традиційні способи колаборативної фільтрації та дає більш якісні рекомендації. Запропоновано спосіб спрощення оригінального алгоритму, що підходить для застосування у великих соціальних мережах, проведено дослідження якості запропонованого методу. Визначено функціональні та нефункціональні вимоги.

У третьому розділі наведено поняття веб-додатку, описана структура веб-додатку. Проаналізований датасет з музичними композиція. Визначено від чого залежить популярність пісні. Розроблений веб-додаток для отримання рекомендацій пісень.

Розроблений веб-додаток має такий функціонал:

- Пошук музичних композиції схожий на введenu пісню;

- Стандартний функціонал плеєру, такий як Play/Stop, Next/Previous;
- Перехід на сторінку з музичним кліпом;



СПИСОК ЛІТЕРАТУРИ

- [1] Item-based collaborative filtering recommendation algorithms / B. Sarwar, G. Karypis, J. Konstan, J. Reidl // Proceedings of the 10th international conference on World Wide Web / ACM. — 2001. — P. 285–295.
- [2] Netflix prize — <http://www.netflixprize.com/>.
- [3] Matrix factorization and neighbor based algorithms for the netflix prize problem / G. Tak'acs, I. Pil'aszy, B. N'emeth, D. Tikk // Proceedings of the 2008 ACM conference on Recommender systems / ACM. — 2008. — P. 267–274.
- [4] Konstan I., Stathopoulos V., Jose J. On social networks and collaborative recommendation // Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval / ACM. — 2009. — P. 195–202.
- [5] Cantador I., Konstan I., Jose J. Categorising social tags to improve folksonomy-based recommendations // Web Semantics: Science, Services and Agents on the World Wide Web. — 2011. — Vol. 9, no. 1. — P. 1–15.
- [6] Castelluccio M. The music genome project // Strategic Finance. — 2006. — Vol. 88, no. 6. — P. 57–58.
- [7] Groh G., Ehmig C. Recommendations in taste related domains: collaborative filtering vs. social filtering // Proceedings of the 2007 international ACM conference on Supporting group work / Citeseer. — 2007. — P. 127–136.
- [8] Koren Y. The bellkor solution to the netflix grand prize // Netflix prize documentation. — 2009.
- [9] Friedman J. Greedy function approximation: a gradient boosting machine // Annals of Statistics. — 2001. — P. 1189–1232.
- [10] Lemire D., Maclachlan A. Slope one predictors for online ratingbased collaborative filtering // Society for Industrial Mathematics. — 2005. — Vol. 5. — P. 471–480.
- [11] Evaluating collaborative filtering recommender systems / J.L. Herlocker, J.A. Konstan, L.G. Terveen, J.T. Riedl // ACM Transactions on Information Systems (TOIS). — 2004. — Vol. 22, no. 1. — P. 5–53.

[12] Slaney M., Casey M. Locality-sensitive hashing for finding nearest neighbors [lecture notes] // Signal Processing Magazine, IEEE. — 2008. — Vol. 25, no. 2. — P. 128–131. [14] Dean J., Ghemawat S. Mapreduce: simplified data processing on 50 large clusters // Commun. ACM. — 2008. — Vol. 51, no. 1. — P. 107– 113.

[13] Машинне навчання: [Електронний ресурс] // Режим доступу: https://www.sas.com/en_gb/insights/articles/analytics/machine-learning-algorithms.html

[14] Types of recommendation systems: [Електронний ресурс] // Режим доступу: <https://medium.com/mllearning-ai/what-are-the-types-of-recommendation-systems-3487cbafa7c9>

[15] Recommendation systems: [Електронний ресурс] // Режим доступу: <https://www.nvidia.com/en-us/glossary/data-science/recommendation-system/>

[16] Pandora internet radio: [Електронний ресурс] // Режим доступу: <https://www.pandorarecovery.com/pandora-internet-radio.html>

[17] Pandora internet radio: [Електронний ресурс] // Режим доступу: <https://en.wikipedia.org/wiki/Spotify>

[18] Collaborative filtering: [Електронний ресурс] // Режим доступу: <https://www.analyticsvidhya.com/blog/2022/02/introduction-to-collaborative-filtering/>

[19] Content based filtering: [Електронний ресурс] // Режим доступу: <https://www.upwork.com/resources/what-is-content-based-filtering>

[20] What is content based filtering: [Електронний ресурс] // Режим доступу: <https://medium.com/analytics-vidhya/what-is-content-based-filtering-an-applied-example-in-python-42acb4cbaa30>

[21] Spotify, Pandora or Apple [Електронний ресурс] // Режим доступу: <https://nerdalert.com/spotify-pandora-or-apple-music-which-one-is-better/>

[22] Recommendation systems a review [Електронний ресурс] // Режим доступу: https://www.researchgate.net/publication/325074466_Recommendation_Systems_a_review

- [23] Рекомендаційна система [Електронний ресурс] // Режим доступу
https://uk.wikipedia.org/wiki/Рекомендаційна_система
- [24] Docker [Електронний ресурс] // Режим доступу
<https://aws.amazon.com/docker/>
- [25] Recommendation systems an overview [Електронний ресурс] // Режим доступу:
https://www.researchgate.net/publication/220604600_Recommender_Systems_An_Overview
- [26] Python [Електронний ресурс] // Режим доступу
[https://en.wikipedia.org/wiki/Python_\(programming_language\)](https://en.wikipedia.org/wiki/Python_(programming_language))
- [27] Python docs [Електронний ресурс] // Режим доступу
<https://docs.python.org/3/>
- [28] Web-application [Електронний ресурс] // Режим доступу
<https://www.stackpath.com/edge-academy/what-is-a-web-application/>
- [29] Web application architecture [Електронний ресурс] // Режим доступу
<https://www.simform.com/blog/web-application-architecture/>
- [30] Modern Web Application Architecture [Електронний ресурс] // Режим доступу
<https://litslink.com/blog/web-application-architecture>
- [31] Machine learning [Електронний ресурс] // Режим доступу
<https://www.techtarget.com/searchenterpriseai/definition/machine-learning-ML>
- [32] What is machine learning model [Електронний ресурс] // Режим доступу
<https://learn.microsoft.com/en-us/windows/ai/windows-ml/what-is-a-machine-learning-model>
- [33] Taxicab geometry [Електронний ресурс] // Режим доступу
https://en.wikipedia.org/wiki/Taxicab_geometry
- [34] What is JavaScript [Електронний ресурс] // Режим доступу
https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript
- [35] 5 problems of recommender system [Електронний ресурс] // Режим доступу
https://readwrite.com/5_problems_of_recommender_systems/

[36] Ratings of musical parameters [Електронний ресурс] // Режим доступу https://www.researchgate.net/publication/256081030_Perceptual_Ratings_of_Musical_Parameters

[37] Languages with ML frameworks [Електронний ресурс] // Режим доступу <https://www.geeksforgeeks.org/top-5-programming-languages-and-their-libraries-for-machine-learning-in-2020/>

[38] Random forest [Електронний ресурс] // Режим доступу <https://towardsdatascience.com/understanding-random-forest-58381e0602d2>

[39] K-means clustering [Електронний ресурс] // Режим доступу <https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1>

[40] Decision tree in ML [Електронний ресурс] // Режим доступу <https://towardsdatascience.com/decision-trees-in-machine-learning-641b9c4e8052>

[41] K-Nearest Neighbor [Електронний ресурс] // Режим доступу <https://medium.com/swlh/k-nearest-neighbor-ca2593d7a3c4>

ДОДАТОК А

Скріншоти рекомендаційної системи

MUSIC RECOMMENDER

Here's some Ambient tracks recommended for you...

Enter search query... Artist Search

#	Track	Artist
1	Startoucher	Biosphere
2	A Long, Long Way	Röyksopp
3	Glósóli	Sigur Rós

MUSIC RECOMMENDER

Here's some Ambient tracks recommended for you...

classical Genre Search

#	Track	Artist
1	Ellen's Gesang III (Ave Maria), Op. 56, No. 6, D. 839, "Hymne An Die Jungfrau"	Ingrid Kertesi
2	Vaughan Williams: The Lark Ascending: Opening (Excerpt)	Ralph Vaughan Williams
3	Symphony No. 9 in D Minor, WAB 109: II. Scherzo: Bewegt, lebhaft – Trio: Schnell – Scherzo	Anton Bruckner
4	Brothers	Ennio Morricone
5	Concerto for Orchestra, Sz.116: Finale: Pesante; Presto	Fritz Reiner
6	Tzigane, M. 76 (Version for Violin & Orchestra)	Maurice Ravel
7	Concerto for Viola & Orchestra: I. Andante comodo	William Walton
8	Cello Suite No. 1 in G Major, BWV 1007: I. Prélude	Johann Sebastian Bach
9	Night on Bare Mountain	Modest Mussorgsky
10	Figment No. 2, "Remembering Mr. Ives"	Elliott Carter
11	The Fellowship Reunited (feat. Sir James Galway, Viggo Mortensen and Renée Fleming)	Howard Shore
12	Tchaikovsky: The Nutcracker, Op. 71, Act II: No. 14c, Pas de deux. Variation II "Dance of the Sugar Plum Fairy"	Pyotr Ilyich Tchaikovsky
13	Undertow	Richard Skelton
14	Toccata And Fugue In D Minor, BWV 565	Leopold Stokowski