

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Штучного інтелекту
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Дослідження та розробка гібридних методів формування рекомендацій
в ситуації холодного старту рекомендаційної системи
(тема)

Виконав:
студент 2 курсу, групи СШМ-21-1
Скриннік В.І.
(прізвище, ініціали)

Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма Системи штучного інтелекту
(повна назва спеціалізації)

Керівник проф. Чалий С.Ф.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____
(підпис)

В.О. Філатов
(прізвище, ініціали)

2023 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)
Кафедра Штучного інтелекту
(повна назва)
Рівень вищої освіти другий (магістерський)
Спеціальність 122 Комп'ютерні науки
(код і повна назва)
Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)
Освітня програма Системи штучного інтелекту (СШІ)
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«_____» _____ 20__ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Скринніку Владиславу Ігоровичу
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження та розробка гібридних методів формування рекомендацій в ситуації холодного старту рекомендаційної системи

затверджена наказом університету від 31 березня 2023 р. № 306Ст

2. Термін подання студентом роботи до екзаменаційної комісії 17 травня 2023 р.

3. Вихідні дані до роботи науково-технічні публікації, відкриті джерела в мережі Інтернет, відкриті набори даних, документація мов програмування, документація алгоритмів, документація проектування систем рекомендацій

4. Перелік питань, що потрібно опрацювати в роботі рекомендаційні системи та алгоритми підбору, ситуація холодного, поняття музичного аудіо-стрімінгу. Обрання мов програмування, інструментів розробки. Побудова веб-додатку

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) _____

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на дипломну роботу	03.04.2023	виконано
2	Аналіз предметної області та постановка задачі	10.04.2023	виконано
3	Огляд методів створення рекомендаційних систем	20.04.2023	виконано
4	Розробка моделі рекомендаційного алгоритму	21.05.2023	виконано
5	Опрацювання ситуації холодного старту	25.04.2023	виконано
6	Вдосконалення існуючого рекомендаційного алгоритму	27.04.2023	виконано
7	Проведення експериментів	30.04.2023	виконано
8	Аналіз отриманих результатів	03.05.2023	виконано
9	Написання пояснювальної записки	07.05.2023	виконано
10	Попередній захист	10.06.2023	
11	Захист перед ЕК	17.05.2023	

Дата видачі завдання 3 квітня 2023 р.

Студент _____


(підпис)

Керівник роботи _____

(підпис)

проф. Чалий С.Ф.

(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 82 с., 24 рис., 3 дод., 14 джерел.

АНАЛІЗ ДАНИХ, ВЕБ-ДОДАТОК, ГІБРИДНІ МЕТОДИ, МАШИННЕ НАВЧАННЯ, МЕТА-ДАНИ, ПРОЕКТ, ХОЛОДНИЙ СТАРТ, DATA INPUTATION, HTTP, JAVASCRIPT, PYTHON, RECCOMENDATION SYSTEMS.

Об'єкт дослідження – процес побудови рекомендацій.

Мета роботи – дослідження гібридних методів побудови рекомендацій для формування переліку товарів та послуг в ситуації холодного старту рекомендаційної системи.

Методи розробки базуються на технологіях JavaScript (візуальна частина) та Python.

У результаті виконання роботи було досліджено методи надання рекомендацій у ситуації холодного старту рекомендаційної системи.

ABSTRACT

Explanatory note: 82 p., 24 fig., 3 ann., 14 sources.

COLD START, DATA ANALYSIS, DATA INPUTATION, HTTP, HYBRID METHODS, JAVASCRIPT, MACHINE LEARNING, METADATA, PROJECT, PYTHON, RECOMMENDATION SYSTEMS, WEB APPLICATION.

The object of research is research and improvement of methods of recommender systems. The study also includes an analysis of existing methods for the formation of recommendations, the development and testing of hybrid methods, as well as an analysis of the results of their use.

The purpose of the work is research and development of hybrid methods of forming recommendations in the conditions of a cold start of the recommendation system.

Development methods are based on JavaScript (visual part) and Python technologies.

As a result of the work, the methods of providing recommendations in the situation of a cold start of the recommendation system were investigated

ЗМІСТ

Вступ.....	8
1 Рекомендаційні системи та алгоритми підбору	9
1.1 Аналіз задач побудови рекомендацій	13
1.2 Аналіз проблеми холодного старту.....	17
1.3 Аналіз методів побудови рекомендацій в ситуації холодного старту	19
1.4 Постановка задачі дослідження.....	22
1.5 Висновок до розділу	24
2 Розробка гібридного методу формування рекомендацій в ситуації холодного старту рекомендаційної системи	25
2.1 Структуризація ситуації холодного старту	25
2.2 Гібридні методи колоборативної фільтрації	28
2.3 Удосконалений гібридний метод у ситуації холодного старту.....	30
2.4 Поняття імпутації даних.....	38
2.5 Використання методів імпутації даних	39
2.6 Результати проведеного експерименту.....	46
2.7 Розробка методу імпутації даних	47
3 Експериментальна перевірка отриманих результатів	51
3.1 Аргументація вибору тематики для програмно-візуальної реалізації рекомендаційної системи, покращеної методом імпутації даних.....	51
3.2 Аналіз алгоритму та систем	52
3.3 Технології рекомендаційних систем у музичному аудіо-стрімінгу .	58
3.4 Обрання мов програмування	60
3.5 Практична реалізація рекомендаційної системи обраними мовами програмування	61
3.6 Програмна реалізація комбінованого методу імпутації.....	64
3.7 Результат роботи імпутаційного методу	66
3.8 Результат роботи рекомендаційної системи	68

Висновки	71
Перелік джерел посилання	72
Додаток А Програмний код файлу app.py	75
Додаток Б Програмний код файлу home.js	77
Додаток В Скріншоти рекомендаційної системи	81
Додаток Г Відомість кваліфікаційної роботи.....	82

ВСТУП

Системи рекомендацій – це інструменти, які можуть рекомендувати елементи, які цікавлять користувача.

Більшість соціальних мереж дозволяє користувачам публікувати різні об'єкти: відео, треки, документи. Деякі, наприклад Last.fm, надають користувачеві доступ до вже існуючого контенту. У тому й іншому випадку кількість подібних об'єктів настільки велика, що користувачеві складно знайти серед них інформацію, яка його цікавить шляхом звичайного перегляду. Рекомендаційні системи вирішують це завдання, передбачаючи передбачливість об'єкта для конкретного користувача, ґрунтуючи свій прогноз на даних, зазначених користувачем явно, або зібраних з історії його взаємодії із соціальною мережею. Від цього підвищується привабливість як самої мережі, так і її вмісту.

Для складання рекомендацій зазвичай використовуються методи колаборативної фільтрації. Коли потрібно передбачити, наскільки користувачеві сподобається новий об'єкт, для цього використовуються оцінки схожих користувачів. Ці методи добре відомі та застосовуються у багатьох проєктах: Netflix, Amazon, Last.fm. За останні роки якість алгоритмів цього значно збільшилася.

Об'єкт дослідження: музичний контент.

Предмет дослідження: алгоритми для пошуку і рекомендації.

Метою магістерської кваліфікаційної роботи є створення програми яка дозволить використати алгоритми для пошуку і надання рекомендацій на основі обраного контенту.

1 РЕКОМЕНДАЦІЙНИ СИСТЕМИ ТА АЛГОРИТМИ ПІДБОРУ

Існує профіль користувача, і профіль користувача може бути створений з використанням неявної та явної інформації від користувача. Неявний спосіб включає історію перегляду, демографічну інформацію тощо. Явна інформація включає запити користувачів оцінити товари та заповнення анкет. Системи рекомендацій класифікуються як колоборативна фільтрація (CF) і фільтрація на основі вмісту.

Колоборативна фільтрація – це техніка, за якої елементи рекомендуються користувачеві на основі схожості користувача з групою користувачів. Спільну фільтрацію можна класифікувати як підхід сусідства та модель латентного фактору. У підході на основі сусідства інтерес користувача визначається за допомогою групи користувачів, з якою він пов'язаний. У моделі латентного фактору інтереси користувачів визначаються як фактори, які можуть представляти інтерес для користувача. Елементи в сховищі постійно оновлюються.

Для досягнення цього результату застосовуються різні методи штучного інтелекту та методи машинного навчання.

Прикладами систем рекомендацій можуть бути google.com, amazon.com, Netflix та інші популярні портали електронної комерції, музики та відео, доступні в Інтернеті. Оскільки ці системи містять мільйони елементів, вони не можуть належним чином функціонувати без хорошої системи рекомендацій.

У додатку системи рекомендацій є два класи сутностей, які ми будемо називати користувачами та елементами. Формальне визначення системи рекомендацій таке:

- C: Набір усіх користувачів;
- S: Набір усіх можливих елементів, які можна рекомендувати, наприклад, відео, пісні та книги;

– U : функція корисності, яка вимірює корисність певного елемента s S для користувача c C , тобто $U: C \times S$;

– R , де R – цілком упорядкована множина.

Простір S можливих елементів може бути дуже великим, коливаючись у сотнях або навіть тисячах і тисячах елементів у деяких програмах, таких як рекомендація книг або компакт-дисків. Подібним чином територія людини також може бути дуже великою – у деяких випадках мільйони.

У машині рекомендацій наскільки корисний предмет визначається його рейтингом. Рейтинг – це показник, який подобається користувачеві. Оскільки його дає користувач, то надійність цього рейтингу має додаткову цінність для розуміння вибору користувача. Рейтинг можна брати різними способами. деякі звичайні форми оцінювання – це запити на основі шкал від 1 до 5, як у випадку програм у Google Play. Він також може запитувати на основі шкали від 1 до 10, яка використовується в багатьох методах оцінювання. Багато агентств обслуговування клієнтів використовують цю шкалу оцінювання. Незалежно від того, яку шкалу оцінювання ви збираєтеся використовувати, одна й описує те, що вам подобається, а інші описують ступінь неприязні.

Профіль користувача можна створити, зберігаючи його характеристики, такі як вік, стать, область, електронна пошта, мобільний телефон та інші речі. Профіль товару може бути створений на основі особливостей товару. Як і у випадку з книгою, це мова, яку автор Йона коштував видавцеві тощо. У випадку з телебаченням це споживання електроенергії за ознакою бренду та багато функцій, які можна використовувати для створення профілю. Те, як ми створюємо профіль, має важливий вплив на систему рекомендацій. Рейтинги складаються для підмножини даних, а не для всіх даних. Рейтингова матриця створюється між користувачем і товаром, і це стає серцем системи рекомендацій. Спосіб аналізу рейтингової матриці визначає систему рекомендацій. Різні домени

використовують різні методи для отримання даних із матриці елементів користувача.

Системи рекомендують різні елементи для користувача на основі будь-яких методів машинного навчання або штучного інтелекту, які вони використовували в рейтинговій матриці. Хороші системи рекомендацій покращуються завдяки відгукам користувачів. Хороша система рекомендацій також дає хороші рекомендації, навіть незважаючи на дуже мало оцінок. Точність хороших систем рекомендацій зростає, оскільки історія користувачів збільшується в системі. Коли новий елемент надходить до репозиторію, його не можна рекомендувати, оскільки про цей елемент немає явної чи неявної інформації. У літературі це розглядається як проблема холодного запуску. Проблема холодного запуску може бути проблемою холодного запуску нового користувача, коли є новий користувач і немає інформації про користувача, або проблемою холодного запуску нового елемента, коли є новий елемент і немає інформації про елементи.

Рекомендаційні системи призначені для пошуку об'єктів, які сподобаються користувачеві або будуть корисні для нього. У типових системах є список користувачів $U = (u_1, u_2, \dots, u_m)$ та предметів $I = (i_1, i_2, \dots, i_n)$. Під час взаємодії з системою користувачі знайомляться з об'єктами, формуючи матрицю рейтингів R , де $r_{ua,i}$ – рейтинг предмета $i \in I$ у користувача $ua \in U$ [12]. Як правило, матриця рейтингів дуже розріджена, т.к. кількість різних предметів у системі велике і вже відомі ua предмети $I_{ua} \subseteq I$ становлять малу частку від загальної кількості. Завдання рекомендаційної системи зазвичай формулюється як обчислення передбачення та рекомендації. Користувач, для якого вони обчислюються, далі називатиметься активним.

Пророцтво – чисельне значення $r_{ua,i}$, що виражає передбачену перевагу предмета $i \in I_{ua}$ для активного користувача ua . Передбачене

значення лежить у заздалегідь визначеному інтервалі рейтингу, наприклад від 1 до 5.

Рекомендація – список із N предметів $Ir \subset I$, найбільш бажаних для активного користувача, причому в нього входять лише незнайомі користувачеві елементи $Ir \cap I = \emptyset$. Завдання в такій постановці також називають рекомендацією Top-N.

Способи формування матриці рейтингів бувають явні та неявні. Перші припускають, що користувач, ознайомившись із предметом, виставить йому оцінку за наявною шкалою (так працює, наприклад, Netflix). У таких випадках R складена з оцінок у чистому вигляді. Однак цей підхід застосовується не скрізь. Часто для складання матриці доводиться аналізувати історію взаємодії користувачів із предметами, у цьому випадку має місце неявне формування рейтингу. У випадку з треками, величина $r_{u,i}$ може обчислюватися як кількість прослуховувань.

Формально завдання знаходження рекомендованого об'єкта можна представити наступним чином:

$$\forall u \in U, s'_u = \arg \max_{s \in S} h(u, s). \quad (1.1)$$

де U – множина користувачів;

S – безліч об'єктів, які можуть бути рекомендовані користувачеві;

h – функція, що визначає наскільки деякий об'єкт s задовольняє деякого користувача u .

Таким чином, необхідно вибрати такий об'єкт $s' \in S$, при якому значення задоволеності для кожного користувача $u \in U$ максимального.

Існує три основні підходи до рекомендацій:

- фільтрування вмісту (item-based);
- колаборативна фільтрація (user-based);
- гібридний підхід.

1.1 Аналіз задач побудови рекомендацій

1.1.1 Загальний опис побудови рекомендацій

Проблема холодного старту в рекомендаційних системах може значно ускладнити процес надання релевантних рекомендацій новим користувачам або новим елементам каталогу. Основна причина цієї проблеми полягає в тому, що система не має достатньої кількості інформації про користувача або елемент, щоб зробити точні рекомендації. Це може призвести до того, що нові користувачі не отримують релевантних рекомендацій, що може вплинути на їх задоволеність використанням системи та зменшити їх бажання повернутися на платформу. Нові елементи також можуть не отримувати достатньої уваги в системі, що може призвести до зниження попиту на них та їх помітності серед користувачів.

Для подолання проблеми холодного старту в рекомендаційних системах, можна використовувати різні підходи. Один з таких підходів – це використання колаборативного фільтрування на основі контенту. Цей підхід базується на тому, що якщо користувач зробив вибір певного елемента, то система може запропонувати інші елементи зі схожим контентом, що сподівається, що буде релевантними для користувача. Наприклад, якщо користувач обрав певну книгу, то система може запропонувати інші книги зі схожою тематикою або жанром.

Рекомендаційні системи є важливим інструментом для покращення користувацького досвіду в багатьох сферах, включаючи електронну комерцію, соціальні мережі, медіа, культуру та багато іншого. Одним з головних завдань рекомендаційних систем є знаходження найбільш відповідних або релевантних елементів для користувача, основуючись на їхніх діях, інтересах, поведінці та інших факторах.

Побудова рекомендацій включає в себе такі етапи, як збір та аналіз даних, побудову моделей рекомендацій, відбір та ранжування

рекомендацій (рисунок 1.1). Залежно від характеру завдання та типу даних, можуть використовуватись різноманітні методи та алгоритми, такі як колаборативна фільтрація, контент-базована фільтрація, глибинне навчання та інші.

У процесі побудови рекомендацій необхідно вирішувати багато важливих завдань, таких як підвищення точності та релевантності рекомендацій, зменшення ризику спаму та підробок, забезпечення ефективності та масштабованості системи, захист персональної інформації користувачів тощо.



Рисунок 1.1 – Процес побудови рекомендацій

Завдання побудови рекомендацій можуть відрізнятися залежно від специфіки використовуваних даних та контексту застосування. Наприклад, в електронній комерції однією з основних задач є підвищення конверсії та збільшення обсягу продажів. Для досягнення цих цілей можуть бути використані такі методи як колаборативна фільтрація, контент-базована фільтрація, гібридні методи та інші.

У соціальних мережах, завдання побудови рекомендацій можуть включати знаходження потенційних друзів, рекомендації відповідних груп та спільнот, підвищення зацікавленості користувача контентом та інші.

У культурі та медіа, рекомендаційні системи можуть бути використані для підвищення зацікавленості глядачів та читачів, забезпечення різноманітності та підвищення якості контенту, а також підвищення лояльності аудиторії.

Одним з важливих аспектів різноманітності завдань побудови рекомендацій є збір та обробка даних. Для досягнення якісних результатів необхідно забезпечити якість даних, а також уникнути недостатньої кількості даних для побудови моделі.

Для забезпечення якісної роботи рекомендаційної системи, необхідно звернути особливу увагу на збір та обробку даних. Недостатня кількість даних або неправильний їх формат можуть значно погіршити якість рекомендацій.

Збір даних може бути здійснений різними способами, включаючи збір даних з веб-сторінок, використання даних соціальних мереж, даних з транзакцій та інші. Після збору даних необхідно їх обробити, щоб використовувати їх для побудови моделі рекомендацій.

Одним з ключових завдань обробки даних є очистка та препроцесинг. Це означає видалення непотрібної або дубльованої інформації, нормалізацію та векторизацію даних, щоб зробити їх придатними для подальшої обробки.

Для досягнення якісних результатів та забезпечення точності рекомендаційної системи необхідно використовувати різноманітні методи машинного навчання та статистичного аналізу даних. До таких методів можуть відноситися алгоритми колаборативної фільтрації, контент-базовані методи та гібридні рекомендаційні системи.

Колаборативна фільтрація – це метод, який базується на знаходженні подібності між користувачами або товарів. Для цього використовується матриця оцінок, яка відображає оцінки користувачів для товарів. На основі цієї матриці можна визначити подібність між користувачами або товари.

Контент-базовий метод використовує інформацію про властивості товарів та інтереси користувачів для побудови рекомендацій. Для цього необхідно зібрати та обробити інформацію про властивості товарів, наприклад, жанр фільму або автора книги, та про інтереси користувачів, наприклад, їх вподобання щодо певного жанру.

Одним з недоліків контент-базового методу є обмеженість відомостей про користувачів та товари, що можуть бути враховані. Якщо користувачі не мають достатньої інформації про певний товар, наприклад, через те, що він був недоступний на ринку протягом тривалого часу, то рекомендаційна система не зможе врахувати цей товар при побудові рекомендацій для користувача.

Гібридні рекомендаційні системи поєднують різні методи побудови рекомендацій з метою покращення точності та різноманітності рекомендацій. Наприклад, можуть бути поєднані колаборативна фільтрація та контент-базовий метод, щоб врахувати як інтереси користувачів, так і характеристики товарів.

Окрім цього, для досягнення максимальної точності рекомендаційної системи, необхідно використовувати різноманітні метрики оцінки якості рекомендацій, такі як середня абсолютна помилка (MAE), середня квадратична помилка (RMSE) та інші. Ці метрики дозволяють оцінювати точність рекомендаційної системи та виявляти помилки, які необхідно виправити [13].

Узагальнюючи, рекомендаційні системи є важливим інструментом в сучасному світі, де інформаційна надмірність стає все більшою проблемою. Для забезпечення якісної роботи рекомендаційної системи, необхідно звернути особливу увагу на збір та обробку даних, використання різноманітних методів машинного навчання та статистичного аналізу даних, а також використання різноманітних метрик оцінки як точності, так і ефективності рекомендаційної системи.

1.2 Аналіз проблеми холодного старту

Холодний старт – це ситуація, коли рекомендаційна система не може рекомендувати продукти користувачеві в зв'язку з тим, що недостатньо інформації про користувача та його взаємодії з системою (рисунок 1.2).

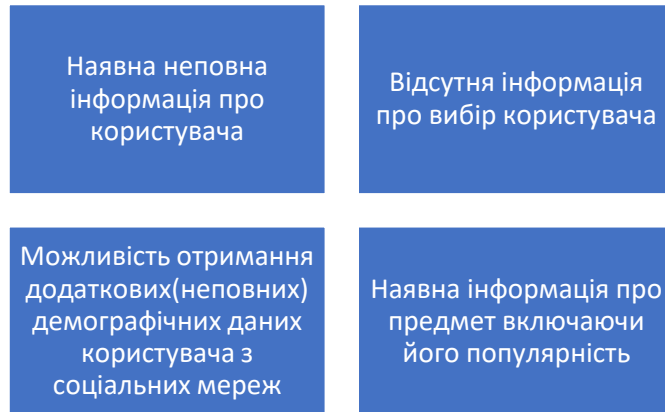


Рисунок 1.2 – Характеристика холодного старту

Ми можемо мати три різні підпроблеми в області проблем холодного запуску. Перша проблема виникає, коли ми не маємо жодної інформації про нового користувача, який входить у систему. це відбувається лише тоді, коли користувач приєднується до системи вперше. Прикладом може бути, якщо ви приєднуєтесь до Amazon або Flipkart вперше, коли виникає така проблема. Ця проблема відома як проблема холодного запуску нового користувача. Друга проблема виникає, коли ми вводимо в систему новий елемент. цей предмет дуже оригінальний у своєму роді. Система рекомендацій не може знайти жодних оцінок, пов'язаних із цим товаром. Система спільної фільтрації, якій потрібна матриця оцінки елементів користувача, щоб надати рекомендацію, не може запуснитися, і проблема відома як проблема холодного запуску. Третя проблема виникла, коли ми вперше запустили систему. У цьому випадку ми не маємо ні інформації про користувача, ні інформації про предмети. Іншими словами, ми не маємо

жодної матриці оцінок за елементами користувача, яка потрібна для належної роботи спільних систем рекомендацій. Ця проблема відома як проблема системи холодного запуску. У проблемах холодного запуску можна застосувати відомі рішення на основі вмісту для вирішення проблеми. Інші рішення використовують комбінацію різних методів машинного навчання.

Якщо говорити більш конкретно про основні проблеми, пов'язані з холодним стартом, включають:

Недостатньо інформації про користувача: у випадку нового користувача, система не має достатньо даних про його інтереси, поведінку, відгуки тощо. Це зменшує точність рекомендацій та знижує задоволеність користувача.

Обмежені можливості фільтрації: у випадку нового продукту, система не може використовувати інформацію про попередні взаємодії з користувачами, оскільки немає жодного відгуку або дій користувача щодо цього продукту.

Проблеми з розміткою: у випадку нового користувача або нового продукту, система може мати проблеми зі збором та аналізом даних, оскільки не завжди можна віднести дії користувачів до певної категорії.

Проблема розвитку: у випадку малої кількості користувачів або продуктів, система може мати проблеми зі зростанням та розвитком, оскільки не вистачає даних для покращення алгоритмів рекомендацій.

Холодний старт є надзвичайно важливою проблемою для рекомендаційних систем, оскільки він може позначитися на якості рекомендацій та задоволеності користувача. Однак, є кілька можливих шляхів для розв'язання проблем холодного старту [4].

По-перше, можна використовувати різноманітні методи аналізу та обробки даних, щоб зібрати більше інформації про користувачів. Наприклад, можна використовувати дані з соціальних мереж або попередніх

взаємодій користувача зі схожими продуктами, щоб дізнатися про їх інтереси та вподобання.

По-друге, можна використовувати методи активного навчання, які дозволяють зібрати більше даних про користувачів. Наприклад, можна використовувати експерименти, де користувачі запрошуються відповісти на запитання про свої вподобання та інтереси.

По-третє, можна використовувати генеративні моделі, які дозволяють генерувати синтетичні дані про користувачів та продукти на основі вже наявних даних. Це допомагає збільшити обсяг даних та забезпечити більш точні рекомендації для нових користувачів або продуктів.

По-четверте, можна використовувати гібридні моделі, які комбінують кілька методів рекомендацій в одній системі. Наприклад, можна використовувати колаборативний та контентний підхід для рекомендацій.

Загалом, холодний старт є важкою проблемою для рекомендаційних систем, але застосування вищезгаданих методів може допомогти зменшити його вплив на якість рекомендацій та задоволеність користувачів. Важливо розуміти, що кожен підхід має свої переваги та недоліки, і вибір підходу для конкретної рекомендаційної системи залежить від різних чинників, таких як тип продукту, масштаб системи та доступність даних. Однак, якщо використовувати комбінацію декількох методів та постійно вдосконалювати рекомендаційну систему, можна досягти більшої точності та ефективності в роботі з користувачами.

1.3 Аналіз методів побудови рекомендацій в ситуації холодного старту

Холодний старт є однією з найважливіших проблем в області рекомендаційних систем. Це стосується ситуації, коли система звертається до нового користувача, для якого немає жодної інформації про його вподобання та інтереси, або до нового продукту, про який немає достатньої

кількості даних. У таких випадках стандартні алгоритми рекомендаційних систем не можуть впоратися з поставленою задачею, оскільки не мають достатньої кількості даних для побудови зв'язків між користувачами та продуктами.

Для розв'язання проблеми холодного старту, дослідники запропонували кілька методів побудови рекомендацій, які дозволяють зменшити вплив цієї проблеми на ефективність рекомендаційних систем.

Першим методом є використання контентних факторів. Контентні фактори включають інформацію про характеристики продуктів, такі як автор, рік видання, жанр, опис тощо, а також профіль користувача, який містить інформацію про його вік, стать, місцезнаходження, звички та інтереси. Використання контентних факторів дозволяє знайти схожі продукти за їхніми характеристиками та зв'язати їх з новим користувачем на основі його профілю. Цей метод дозволяє створювати рекомендації для нових користувачів та продуктів, що не мають достатньої кількості історичних даних [1].

Другим методом є використання гібридних рекомендаційних систем, які комбінують різні методи рекомендацій в одній системі. Наприклад, можна використовувати контентні фактори разом з колаборативним фільтруванням, яке базується на історичних даних про взаємодії користувачів з продуктами. Гібридні системи дозволяють знизити вплив проблеми холодного старту, коли в системі недостатньо історичних даних для рекомендацій.

Третім методом є використання ранжування рекомендацій. Ранжування рекомендацій дозволяє присвоювати вагу різним факторам, які впливають на рівень релевантності рекомендації. Наприклад, якщо два продукти мають схожі контентні характеристики, то рекомендація може бути ранжована в залежності від рейтингу, який користувачі присвоюють цим продуктам.

Четвертим методом є використання аналізу зв'язків між продуктами та користувачами. Цей метод використовує аналіз графа взаємодій міжкористувачами та продуктами, щоб знайти спільні зв'язки та встановити взаємозв'язки між ними. Наприклад, якщо користувачі, які мають подібний профіль, взаємодіють з одними й тими ж продуктами, то можна припустити, що ці продукти є взаємозамінними. Цей метод дозволяє знаходити неочевидні зв'язки та забезпечує більш точні рекомендації.

П'ятим методом є використання інтерактивних рекомендаційних систем, які дозволяють користувачам активно взаємодіяти з системою та надавати зворотний зв'язок.

Цей метод передбачає, що рекомендації не створюються тільки на основі вхідних даних, але й на основі дій користувачів, які взаємодіють з системою. Інтерактивні рекомендаційні системи забезпечують користувачам можливість вибирати, чи вони хочуть отримувати рекомендації, а також надавати системі зворотний зв'язок про рекомендації, які їм були надані.

Один з простих прикладів інтерактивних рекомендаційних систем – це система, що пропонує підписатися на різні категорії новинних розсилок, коли користувач реєструється на новинний сайт. Користувач може обрати, які новинні категорії він хоче отримувати і як часто він хоче отримувати ці розсилки.

Інтерактивні рекомендаційні системи можуть бути використані і в інших галузях, таких як онлайн-магазини, де користувачі можуть брати участь у опитуваннях, розглядати товари та надавати зворотний зв'язок про продукти, які їм сподобалися, або не сподобалися.

У підсумку, існує багато різних методів створення рекомендацій, і краще використовувати їх у поєднанні. Завдяки поєднанню різних методів, можна створити більш точну та персоналізовану систему рекомендацій, яка допоможе користувачам знайти те, що їм дійсно цікаво і корисно.

1.4 Постановка задачі дослідження

Багато складних систем можна добре описати мережами, де вузли представляють індивідів, а ребра позначають відносини між ними. Останнім часом персоналізовані рекомендації в складних мережах привернули все більшу увагу фізиків. Персоналізована рекомендація спрямована на пошук об'єктів (наприклад, книг, веб-сторінок, музики тощо), які, найімовірніше, збиратимуть користувачі. Наприклад, класичний пошук інформації можна розглядати як рекомендацію документів із заданими словами, а процес передбачення посилань можна розглядати як проблему рекомендації в односторонніх мережах. Центральну проблему персоналізованих рекомендацій можна розділити на дві частини: одна – це оцінка подібності на основі історичних записів дій користувачів; інший – це використання додаткової інформації (наприклад, атрибутів об'єктів) для ефективного фільтрування нерелевантних об'єктів. Для формального завдання, оскільки обчислення та зберігання подібності всіх пар користувачів є дорогим, ми зазвичай розглядаємо лише перших k найбільш схожих користувачів. Для останнього завдання дуже точні описи об'єктів можуть бути корисними для фільтрації нерелевантних об'єктів, однак це обмежується словником атрибутів, і, з іншого боку, атрибути описують глобальні властивості об'єктів, які менш корисні для генерувати персоналізовані рекомендації.

Система соціального тегування дозволяє користувачам вільно призначати теги для анотації своїх колекцій, не вимагає від користувачів спеціальних навичок для участі, розширює семантичні зв'язки між користувачами та об'єктами, і, таким чином, привернула велику увагу наукової спільноти. Подібно до функцій тегування, ключові слова та номери PACS аналізуються, щоб краще охарактеризувати структуру співавторства та мереж цитування [10].

Системи соціальних тегів вже знайшли широке застосування в Recommender Systems. Тобто користувачі можуть ефективно знаходити

цільові об'єкти за допомогою тегів. Зокрема, ми розглядаємо частоту використання тегів як особисті переваги користувачів, а семантичні зв'язки між тегами та об'єктами – як глобальну інформацію. Експериментальні результати показують, що даний алгоритм може значно підвищити точність рекомендацій. Подальше емпіричне дослідження показує, що запропонований алгоритм особливо ефективний для об'єктів, зібраних кількома користувачами, що нагадує нам добре відому проблему холодного запуску. Оскільки інформації про нові об'єкти мало, соціальні теги можуть ефективно будувати зв'язки між існуючими та новими об'єктами. Таким чином, введення тегів може надзвичайно допомогти користувачам знаходити нові (або менш популярні), але цікаві об'єкти, і таким чином підвищити загальну точність. Крім того, ми використовуємо методи на основі ентропії та відстані Хеммінга для вимірювання внутрішньої та міжрізноманітності шаблонів використання тегів відповідно.

Постановка задачі дослідження полягає у вдосконаленні стандартних алгоритмів рекомендацій за допомогою гібридних методів формування рекомендацій в ситуації холодного старту рекомендаційної системи.

Основною метою дослідження є створення ефективної та точної рекомендаційної системи, яка зможе надавати персоналізовані рекомендації користувачам навіть в ситуації, коли в системі недостатньо даних про користувача.

Для досягнення цієї мети необхідно вирішити наступні завдання:

- проаналізувати та описати існуючі стандартні алгоритми рекомендацій та їхні переваги та недоліки;
- дослідити методи гібридної рекомендації та їхні можливості в покращенні ефективності та точності рекомендацій;
- розробити нові гібридні методи рекомендацій, які дозволять покращити точність та ефективність рекомендацій в ситуації холодного старту;

– провести експериментальне дослідження запропонованих гібридних методів та порівняти їхню ефективність та точність з існуючими стандартними алгоритмами рекомендацій;

– розробити прототип рекомендаційної системи, яка використовує запропоновані гібридні методи та перевірити її ефективність та точність на реальних даних.

Реалізуючи існуючі стандартні алгоритми рекомендацій, буде досліджено такі алгоритми як колаборативна фільтрація, контент-базована рекомендація та гібридні комбінації цих алгоритмів. Для дослідження гібридних методів будуть застосовані такі підходи як гібридна колаборативна фільтрація, гібридна контент-базована рекомендація та комбінації цих двох підходів.

Для розробки прототипу рекомендаційної системи будуть використані мова програмування Python, а також різноманітні бібліотеки та фреймворки для роботи з даними та машинним навчанням, такі як Pandas, Scikit-learn, TensorFlow, та Keras.

1.5 Висновок до розділу

В даному розділі було розглянуто основні алгоритми та методи, що використовуються в рекомендаційних системах. Фільтрування вмісту, колаборативна фільтрація та гібридні методи фільтрації є основними підходами до підбору рекомендацій. Також були детально описані особливості обробки великих обсягів даних, методи зведення розмірності та аналіз кореляції між даними. Використання цих методів та алгоритмів дозволяє підвищити точність та ефективність рекомендаційної системи та зробити її більш персоналізованою для кожного користувача.

2 РОЗРОБКА ГІБРИДНОГО МЕТОДУ ФОРМУВАННЯ РЕКОМЕНДАЦІЙ В СИТУАЦІЇ ХОЛОДНОГО СТАРТУ РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ

2.1 Структуризація ситуації холодного старту

Ситуація холодного старту є викликом для рекомендаційних систем, які мають на момент запуску обмежену кількість даних про користувачів та їх взаємодії з системою. У такій ситуації, система не може забезпечити достатньої кількості персоналізованих рекомендацій, що може призвести до негативного впливу на користувачів та зниження якості обслуговування.

Моделі користувача, які використовуються на момент холодного старту, зазвичай є базовими та не містять достатньої кількості даних для побудови робочої моделі. Це може призвести до неточностей та помилок у рекомендаціях, що, в свою чергу, може призвести до незадоволення користувачів та зменшення їх лояльності до системи [6].

Одним з можливих рішень цієї проблеми є використання гібридних методів формування рекомендацій, які комбінують у собі різні підходи та алгоритми для отримання кращої якості рекомендацій. Наприклад, можна використовувати методи колаборативного та контент-базованого підходів, які дозволяють враховувати як інформацію про користувачів та їх взаємодії з системою, так і про деталі контенту, який система пропонує.

Крім того, важливо провести попереднє дослідження та аналіз даних, що є на момент холодного старту, для отримання якомога більшої кількості інформації про користувачів та їх взаємодії з системою. Це можна зробити шляхом використання методів екстракції знань з даних та аналізу даних, таких як кластеризація, асоціативне правило та інші. Застосування цих методів дозволить виявити патерни та закономірності у поведінці користувачів, що можуть бути використані для покращення якості рекомендацій на стадії холодного старту.

Також, важливо використовувати методи активного навчання та залучення користувачів до процесу формування рекомендацій. Наприклад, можна використовувати методи збору даних від користувачів, такі як запити на зворотний зв'язок та опитування, для збору додаткової інформації про їх вподобання та інтереси. Це дозволить збільшити обсяг даних та покращити якість рекомендацій на стадії холодного старту.

Крім того, можна використовувати методи активного навчання, такі як кероване навчання та підсилене навчання, для покращення якості рекомендацій на стадії холодного старту. Кероване навчання дозволяє системі використовувати експертні знання та попередні досвіди для покращення рекомендацій. Підсилене навчання дозволяє системі самостійно вчитися та покращувати свої рекомендації на основі взаємодії з користувачами.

Узагалі, використання гібридних методів, проведення попереднього аналізу даних, використання методів активного навчання та залучення користувачів до процесу формування рекомендацій – це лише декілька можливих шляхів для покращення якості рекомендацій на стадії холодного старту. Важливо також забезпечити якість даних та моніторинг їх якості, щоб уникнути помилок та забезпечити точність та достовірність рекомендацій.

Крім того, важливо враховувати індивідуальні потреби та вподобання користувачів при формуванні рекомендацій, оскільки кожен користувач є унікальним. Наприклад, система може використовувати інформацію про розташування користувача та рекомендувати місцеві події або ресторани, що можуть відповідати його географічному положенню та інтересам.

Для покращення якості рекомендацій на стадії холодного старту, можна також використовувати механізми зворотного зв'язку від користувачів. Наприклад, система може запитувати користувачів про те, чи сподобалась їм певна рекомендація, чи є вони задоволені рівнем персоналізації, чи бажають вони отримувати більше рекомендацій з

конкретної категорії тощо. Такий зворотний зв'язок може допомогти системі покращити свої рекомендації та враховувати індивідуальні потреби користувачів.

Отже, для покращення якості рекомендацій на стадії холодного старту необхідно використовувати комплексний підхід, що включає в себе проведення попереднього аналізу даних, використання гібридних методів, методів активного навчання, залучення користувачів та механізми зворотного зв'язку. Крім того, важливо забезпечити якість даних та моніторинг їх якості, щоб уникнути помилок та забезпечити точність та достовірність рекомендацій.

Для досягнення кращих результатів на стадії холодного старту, можна також використовувати нові технології та інноваційні підходи, такі як машинне навчання та штучний інтелект. Застосування цих технологій може значно покращити якість рекомендацій та зробити їх більш персоналізованими.

Нарешті, для забезпечення успіху системи рекомендацій на стадії холодного старту, необхідно постійно вдосконалювати її та адаптувати до змін у потребах користувачів. Це може бути досягнуто шляхом проведення регулярного моніторингу та аналізу даних, відстеження нових тенденцій та впровадження інновацій, а також залучення користувачів до процесу формування рекомендацій шляхом збору додаткової інформації про їхні вподобання та поведінку.

Стадія холодного старту є дуже важливою для системи рекомендацій, оскільки саме на цій стадії користувачі формують свої перші враження від сервісу. Тому, якщо система зможе надати користувачам якісні та персоналізовані рекомендації вже з перших кроків, то шанси на успіх значно збільшаться. Для цього необхідно використовувати різноманітні методи та підходи, такі як збір даних про користувачів, аналіз їх поведінки та використання інноваційних технологій, таких як машинне навчання та штучний інтелект. Також, важливо забезпечувати постійне вдосконалення

та адаптацію системи до змін у потребах користувачів, що допоможе забезпечити її успішний розвиток на довгостроковій перспективі.

Таким чином ситуація холодного старту характеризується властивостями наведеними на рисунку 2.1.



Рисунок 2.1 – Характеристика ситуації холодного старту

2.2 Гібридні методи колоборативної фільтрації

Гібридні рекомендаційні системи поєднують два або більше рекомендаційних підходів, щоб використовувати найкращі аспекти кожного з них. Існує сім основних методів гібридизації, включаючи зважене поєднання різних підходів, перемикування між різними рекомендаціями залежно від поточної ситуації, представлення результатів різних підходів одночасно, поєднання функцій різних підходів, каскадування результатів, збільшення вхідних даних одного підходу за допомогою результату іншої та використання іншої моделі для вирішення проблеми. Щоб уникнути обмежень спільної фільтрації та покращити її продуктивність, фільтрацію на основі пам'яті або на основі моделі можна комбінувати з кожним із

наведених вище підходів. Більшість досліджень використовують комбінацію спільної фільтрації та фільтрації на основі вмісту, щоб вирішити проблеми масштабованості та незгодженості інтересів споживачів у спільній фільтрації.

У зв'язку з зростаючим обсягом даних та складністю інтелектуальних технологій гібридні рекомендаційні системи стають все більш популярними. Ці системи використовують два або більше підходи до рекомендацій, щоб використовувати найкращі аспекти кожного з них.

Існує сім основних методів гібридизації: зважене поєднання різних підходів, перемикання між різними рекомендаціями залежно від поточної ситуації, одночасне уявлення результатів різних підходів, поєднання функцій різних підходів, каскадування результатів, збільшення вкладу одного підходу з результатами іншого та з використанням іншої моделі для вирішення проблеми [1].

Одним із найпоширеніших методів гібридизації є збалансоване поєднання різних підходів. Цей метод полягає в ранжируванні різних підходів до рекомендацій відповідно до їх ефективності, а потім в об'єднанні їх у виважену суму на основі їх рангу. Переключення між різними рекомендаціями в залежності від поточної ситуації означає використання різних підходів залежно від контексту користувача, наприклад, його розташування, часу доби або пристрою, на якому він використовує систему.

За допомогою гібридних рекомендаційних систем можна підвищити якість рекомендацій, оскільки вони дозволяють комбінувати різні підходи та методи для отримання більш точних результатів. Наприклад, поєднання спільної фільтрації та фільтрації на основі вмісту може вирішити проблему конфлікту інтересів споживачів під час спільної фільтрації [4].

Як правило, гібридні рекомендаційні системи будуються з використанням алгоритмів машинного навчання та статистичних методів, таких як класифікація, кластеризація та підсумовування розмірностей. Вони

також можуть використовувати інші методи, такі як аналіз тексту, обробка зображень та аудіоаналіз.

Однак при побудові гібридної рекомендаційної системи важливо враховувати декілька факторів, таких як обсяг даних, якість даних, кількість доступних ресурсів та обмеження часу обчислень. Крім того, необхідно пам'ятати про конфіденційність даних і дотримуватися етичних принципів при зборі та використанні даних.

2.3 Удосконалений гібридний метод у ситуації холодного старту

Системи рекомендацій є важливою частиною сучасного електронного бізнесу. Однак, при старті нової системи рекомендацій, коли немає достатньої інформації про поведінку користувачів, можуть виникати проблеми з якістю рекомендацій. Це відомо як проблема «холодного старту». Щоб покращити якість рекомендацій на стадії холодного старту, можна використовувати удосконалені методи, такі як гібридний метод з додатковою перевіркою умов або фільтрацією.

Гібридний метод рекомендацій – це метод, що поєднує декілька алгоритмів рекомендацій в одній системі. Удосконалений гібридний метод включає додаткову перевірку умов або фільтрацію до вже існуючого методу. Наприклад, фільтрування користувачів у системі холодного старту додатковими умовами, такими як перевірка по айпі, може забезпечити кращу якість рекомендацій на стадії холодного старту.

Фільтрація користувачів на стадії холодного старту може бути здійснена за допомогою різних умов, наприклад, перевірка на наявність попереднього досвіду використання систем рекомендацій, аналіз геолокації або перевірка за айпі-адресою. Фільтрація за айпі може допомогти уникнути багатьох проблем, таких як боти, спам і інші види шахрайства, які можуть впливати на якість рекомендацій. Фільтрування за айпі-адресою може

допомогти виявити користувачів, які вже зареєструвалися в системі рекомендацій, і пропонувати їм відповідні рекомендації [10].

Також можна використовувати додаткові дані про користувачів, такі як їх соціальні мережі, рейтинги та відгуки, щоб покращити якість рекомендацій на стадії холодного старту. Застосування гібридного методу з додатковою перевіркою умов або фільтрацією може покращити точність рекомендацій на стадії холодного старту і допомогти залучити нових користувачів до системи (рисунок 2.2).

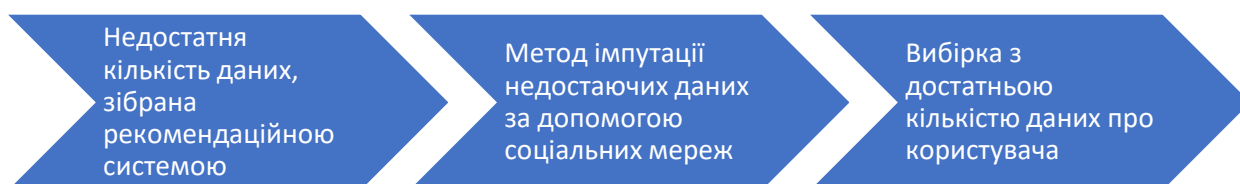


Рисунок 2.2 – Структура удосконаленого гібридного методу

Враховуючи те, що етап додавання даних користувача є важливим етапом у розвитку багатьох сучасних сервісів і платформ, хотілося б надати більш детальний опис кожного з чотирьох етапів, які складають цей етап.

Етап 1 – відбір даних користувача з соціальних мереж. Цей етап зазвичай починається зі збору даних із профілю користувача в певній соціальній мережі, наприклад Facebook, Instagram, LinkedIn, Twitter тощо. Дані, які можуть використовуватися на цьому етапі, можуть включати основну інформацію про користувача, наприклад його ім'я, вік, стать, місцезнаходження, мови, якими вони розмовляють, освіта, робота, інтереси та хобі. Ці дані збираються з різних соціальних мереж і зберігаються в базі даних для подальшої обробки.

Етап 2 – імпутація даних. З огляду на те, що дані, які були зібрані з соціальних мереж, можуть містити багато відсутніх значень, другим етапом

доповнення даних користувача є імпутація даних. Цей етап полягає в заповненні відсутніх значень даними, які з високою ймовірністю належать до розглянутої групи користувачів.

Загалом імпутація даних є важливою складовою процесу доповнення даних користувача, оскільки від цього залежить точність і правильність подальшої обробки даних.

Третій етап поповнення даних користувачів – відбір схожих користувачів. Цей етап передбачає аналіз характеристик користувача та порівняння їх з іншими користувачами, щоб визначити тих, хто має найбільш схожі характеристики. Цей крок допомагає визначити групу користувачів, які можуть мати подібні інтереси та поведінку, і доповнити дані про відповідного користувача на основі інформації про цю групу користувачів.

Для визначення схожості користувачів можна використовувати різні методи, наприклад кластеризацію, метод головних компонентів, дискримінантний аналіз та багато інших. Вибір того чи іншого методу залежить від характеру даних і завдань, які вирішуються.

На четвертому етапі додавання матриці вибору користувача доповнюються додатковими даними, які були отримані на попередніх етапах. Це може включати додавання нових функцій і значень до матриці вибору користувача, що дозволяє отримати більш повну та правильну інформацію про користувача.

Загалом процес розширення даних користувача є складним і багатоетапним процесом, який включає збір даних, імпутацію, аналіз і відбір схожих користувачів, а також розширення матриць вибору користувачів. Правильний підхід до додавання даних дозволяє отримати більш повну та правильну інформацію про користувача, що дозволяє надавати більш ефективно та точно персоналізоване обслуговування користувачів.

Крім того, процес доповнення даних користувачів можна використовувати для вдосконалення різних бізнес-процесів, таких як маркетингові кампанії, рекомендації продуктів, прогнозування попиту та інші.

Наприклад, якщо компанія має детальну інформацію про своїх користувачів, вона може створювати персоналізовані рекомендації для кожного користувача, збільшуючи шанси на успішний продаж своєї продукції. Крім того, компанії можуть використовувати дані користувачів для прогнозування попиту на свої товари та послуги, що дозволяє їм підготувати потрібну кількість продукту за певний період часу та уникнути зайвих витрат на зберігання непроданих товарів.

Загалом, доповнення даних користувачів є важливим етапом у роботі з даними, який дозволяє підвищити якість і коректність даних, які використовуються для прийняття рішень та підготовки аналітичних звітів. Для досягнення максимальної ефективності процес додавання даних користувача повинен бути розроблений і реалізований з урахуванням специфіки даних і потреб бізнесу.

Існує кілька методів імпутації даних, таких як метод середнього значення, метод медіани, метод K-Nearest Neighbors (KNN) і методи машинного навчання. Кожен із цих методів має свої переваги та недоліки, а вибір того чи іншого методу залежить від характеру даних і завдань, які вирішуються.

Наприклад, методом середнього значення можна заповнити пропущені значення числових ознак. Для кожного атрибута, який має пропущені значення, обчислюється середнє значення всіх інших користувачів із подібними характеристиками, і це значення використовується для заповнення пропущених значень. Цей метод досить простий і ефективний для числових даних, але він не підходить для категоріальних даних, таких як стать або місцезнаходження.

Залежно від конкретного завдання додавання даних користувача можуть використовуватися різні математичні формули та алгоритми. Нижче наведено кілька поширених формул, які можна використовувати в процесі додавання даних користувача.

Імпутація даних з використанням медіанного значення:

$$\hat{x}_i = \text{median}(\{x_j | j \in N_i\}) \quad (2.1)$$

де x_i – імпутоване значення;

x_j – значення j -го атрибута користувача;

N_i – набір користувачів, які мають значення i -го атрибут.

Відбір подібних користувачів за допомогою косинусної подібності:

$$\text{sim}(u_i, u_j) = \frac{u_i \cdot u_j}{|u_i| \cdot |u_j|} \quad (2.2)$$

де u_i і u_j – вектори ознак користувачів.

Алгоритм k -середніх для кластеризації користувачів:

- ініціалізація k центрів кластера випадковим чином;
- призначте кожного користувача до найближчого кластера;
- розрахувати нові координати центрів кластерів як середнє арифметичне всіх користувачів у кожному кластері;
- повторюйте попередні два кроки, доки кластери не стабілізуються.

Вигляд KNN алгоритму у математичній інтерпритації можна зобразити наступним чином:

$$\{y\} = \operatorname{operatorname} * \{ \operatorname{argmax} \}_{\{y_i \in \mathbf{Y}\}} \sum_{\{j=1\}}^{\{k\}} w_j [y_i = y_j] \quad (2.3)$$

Ці формули та алгоритми можна доповнювати та змінювати залежно від конкретної проблеми та типу даних, що використовуються.

Метод KNN можна використовувати для заповнення пропущених значень у категоріальних даних. Для кожного відсутнього значення знайдіть K користувачів із найближчими характеристиками та використовуйте їхні значення, щоб заповнити відсутнє значення. Цей метод використовує інформацію про найближчих сусідів, що дозволяє точніше заповнити пропущені значення та зменшити вплив шуму.

Етап 3 – відбір схожих користувачів. Після того, як дані будуть зібрані та вписані, наступним кроком буде вибір користувачів, які мають схожі характеристики. На цьому етапі можна використовувати різні алгоритми машинного навчання, щоб допомогти знайти користувачів зі схожими інтересами, знайомствами, роботою тощо. Це допоможе підвищити точність рекомендацій і забезпечить більш персоналізований досвід для користувачів.

Етап 4 – додавання матриць за вибором користувача. На останньому кроці дані про користувачів, які були вибрані на попередньому кроці, додаються до матриці вибору користувачів. Ця матриця містить інформацію про користувачів та їхні характеристики, які були зібрані та оброблені під час попередніх кроків. Додавання нових даних до матриці допоможе підвищити якість рекомендацій і забезпечить точнішу взаємодію з користувачем. Математична інтерпретація фінального вигляду матриці з імпутованими даними може бути описана наступним чином:

Таблиця N_m з відомими даними $x = x_{ij}$ для $i = 1 \dots n$ та $j = 1 \dots m$. Також, нехай ми маємо новий набір даних $x = x_{ij}$ для $i = 1 \dots k$ та $j = 1 \dots m$ де k - кількість нових записів. Для додавання нових даних в таблицю, ми можемо використати наступну математичну формулу:

$$\mathbf{X}_{\text{new}} = \begin{pmatrix} \mathbf{X} \\ \mathbf{X}' \end{pmatrix} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1m} \\ x_{21} & x_{22} & \cdots & x_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nm} \\ x'_{1,1} & x'_{1,2} & \cdots & x'_{1,m} \\ x'_{2,1} & x'_{2,2} & \cdots & x'_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ x'_{k,1} & x'_{k,2} & \cdots & x'_{k,m} \end{pmatrix} \quad (2.4)$$

де x_{new} – це оновлена таблиця, яка включає в себе як відомі, так і нові дані;

\mathbf{X} – відомі дані;

\mathbf{X}' – нові дані;

x_{ij} та x'_{ij} – це елементи таблиці.

Взагалі етап доповнення даних користувача є важливою складовою багатьох сучасних сервісів і платформ. Ці кроки, які я описав вище, допомагають збирати та обробляти дані користувачів, щоб забезпечити більш персоналізовану та точну взаємодію з користувачем. Цей процес безперервний, оскільки дані користувача змінюються з часом, тому фаза збільшення даних користувача має бути завжди напоготові та готовою до збору та обробки нових даних.

Для реалізації удосконаленого гібридного методу у системі рекомендацій можна використовувати алгоритми машинного навчання, такі як алгоритм колаборативного фільтрування, алгоритм контент-базованої фільтрації, алгоритми з урахуванням контексту, такі як факторизація матриці, але з додатковою фільтрацією. Крім того, можна використовувати методи розпізнавання образів та обробки природних мов, якщо вони застосовні до конкретної системи рекомендацій [2].

Гібридний метод з додатковою перевіркою умов або фільтрацією є одним з найбільш ефективних підходів до створення систем рекомендацій. Проте, при його застосуванні потрібно враховувати деякі важливі аспекти.

Перш за все, необхідно забезпечити залучення нових користувачів до системи, і використані умови або фільтри не повинні ставити перешкоди цьому процесу [7]. Крім того, забезпечення захисту даних користувачів, зокрема при використанні додаткових даних про них з соціальних мереж, є надзвичайно важливим аспектом.

Удосконалені методи рекомендацій можуть значно поліпшити якість рекомендацій на стадії холодного старту. Це може сприяти залученню нових користувачів до системи та збільшенню її популярності. Крім того, застосування таких методів може покращити задоволеність користувачів та зменшити кількість відмов від використання системи.

Для успішної реалізації гібридного методу з додатковою перевіркою умов або фільтрацією в системі рекомендацій, необхідно мати достатній обсяг даних та використовувати потужні алгоритми машинного навчання. Крім того, важливо проводити регулярні оновлення системи та аналізувати результати її роботи, щоб виявляти та виправляти помилки [5].

Застосування удосконалених методів рекомендацій у різних сферах може мати значний вплив на покращення досвіду користувачів і збільшення прибутку бізнесу. Наприклад, в онлайн-торгівлі, застосування гібридних методів рекомендацій може підвищити кількість покупок за рахунок залучення нових клієнтів та підвищення лояльності вже існуючих. У музичній та кінематографічній сферах, удосконалені методи рекомендацій можуть підвищити інтерес користувачів до нових виконавців та фільмів, збільшити кількість прослуховувань і переглядів, та забезпечити збільшення доходів для бізнесу.

Однак, необхідно зазначити, що застосування таких методів рекомендацій може виникати етичні проблеми. Наприклад, якщо система рекомендацій використовує додаткові дані користувачів з соціальних мереж, це може порушувати приватність користувачів і породжувати питання про використання їх даних без їх згоди. Також важливо пам'ятати про можливість зловживання системами рекомендацій, які можуть бути

використані для маніпулювання поведінкою користувачів та впливу на їх вибори.

Науковці та інженери продовжують працювати над удосконаленням методів рекомендацій. Наприклад, на даний момент активно використовуються глибинні нейронні мережі, що дозволяє здійснювати більш точні та персоналізовані рекомендації. Крім того, деякі дослідження присвячені використанню методів з розподільним представленням даних (embedding), що дозволяє зменшити розмірність даних та прискорити обчислення.

Також важливим аспектом в розробці систем рекомендацій є врахування етичних та правових аспектів. Наприклад, необхідно забезпечувати конфіденційність даних користувачів та не допускати дискримінації за будь-якими ознаками, такими як стать, раса або вік. Крім того, необхідно враховувати особливості культур та менталітету користувачів, щоб уникнути неприємних ситуацій та конфліктів.

У цілому, системи рекомендацій мають великий потенціал для використання в різних сферах, що дозволяє покращити якість обслуговування та задоволеність клієнтів.

2.4 Поняття імпутації даних

Імпутація даних є важливим етапом в обробці даних, що дозволяє заповнити пропущені значення в датасеті для подальшого використання в аналізі та моделюванні. Однак, вибір критерію якості заповнення пропущених значень може бути складним завданням. Недоімпутовані дані можуть призвести до неточності аналізу або моделювання, тоді як переімпутовані дані можуть призвести до заниження якості моделі. Один з підходів до вибору оптимального критерію якості імпутації полягає в використанні зовнішніх критеріїв. Ці критерії оцінюють якість імпутованих

даних шляхом порівняння прогнозів на тестовій вибірці, отриманих з використанням імпутованих даних та з оригінальними даними.

2.5 Використання методів імпутації даних

Проте, використання зовнішніх критеріїв може бути недостатньо ефективним, оскільки вони не враховують внутрішню структуру даних та можуть не бути достатньо чутливими до невеликих змін у датасеті. Тому, щоб забезпечити оптимальну якість імпутації, можна використовувати методи оптимального планування експерименту. Ці методи дозволяють ефективно визначити оптимальне розбиття даних на навчальну та тестову вибірки з метою максимізації критерію якості. Одним з таких методів є метод Taguchi, який базується на матриці експерименту. Цей метод дозволяє визначити оптимальну комбінацію змінних, яка максимізує критерій якості імпутації.

Іншим методом є метод Vox-Behnken, який також використовує матрицю магистраль експерименту, але відрізняється від методу Taguchi тим, що він не вимагає повного перебору комбінацій змінних, що зменшує обчислювальну складність. Також можна використовувати методи оптимального планування експерименту для визначення оптимального алгоритму імпутації. Наприклад, метод Джекмана-Сімпсона дозволяє визначити оптимальну комбінацію алгоритмів імпутації та їх параметрів, яка максимізує критерій якості.

Оскільки вибір оптимального критерію імпутації та оптимальної комбінації змінних та алгоритмів є складним завданням, можна використовувати автоматичне машинне навчання для побудови моделі, яка буде самостійно вирішувати ці завдання. Наприклад, можна використовувати методи з глибинним навчанням, такі як автокодувальники, щоб автоматично визначати оптимальну комбінацію змінних та алгоритмів імпутації, яка максимізує критерій якості. У кінцевому підсумку,

використання методів оптимального планування експерименту для вибору оптимального критерію якості та алгоритмів імпутації може значно поліпшити якість імпутації даних та підвищити ефективність аналізу та моделювання.

Методи Taguchi, Vox-Behnken та метод Джекмана-Сімпсона є методами оптимального планування експерименту, які використовуються для визначення оптимальної комбінації факторів та їх рівнів, яка максимізує критерій якості.

Метод Taguchi базується на теорії сигналу та шуму, де сигнал описує вплив факторів на відповідь, а шум описує випадкові зміни у відповіді, які не пов'язані з факторами (рисунок 2.3). Метод Taguchi використовує матрицю планування, що складається з рівнів факторів, які тестуються у відповіді. Метод дозволяє використовувати факторний аналіз для визначення впливу кожного фактора на відповідь, а також використовувати метод S/N для визначення кращої комбінації рівнів факторів.

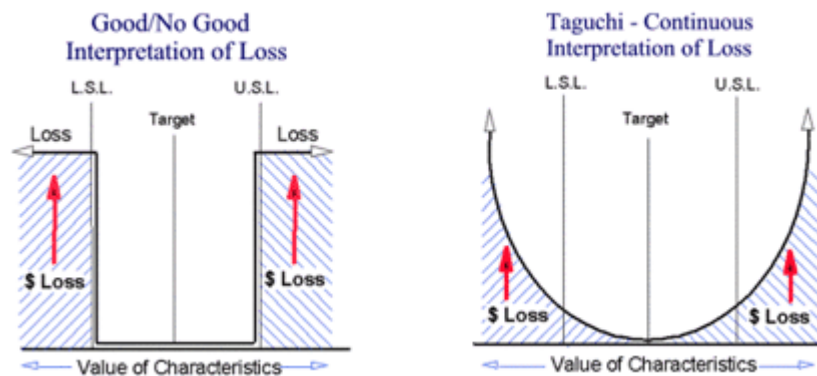


Рисунок 2.3 – Візуальна інтерпретація методу Taguchi

Наприклад, розглянемо таблицю 2.1 (таблиця планування для 3 факторів з 3 рівнями):

Таблиця 2.1 – Реалізація таблиці роботи алгоритму Taguchi L9 (планування для 3 факторів з 3 рівнями)

Експеримент	Фактор 1	Фактор 2	Фактор 3
1	1	1	1
2	1	2	2
3	1	3	3
4	2	1	2
5	2	2	3
6	2	3	1
7	3	1	3

Критерій якості може бути визначений як сума квадратів відхилень від цілі (наприклад, мінімізація відхилень від середнього значення).

Метод Джекмана-Сімпсона базується на ідеї побудови моделі відгуку відносно кожного параметру окремо (рисунок 2.4). Для цього використовуються різні регресійні моделі, такі як лінійна, квадратична або кубічна регресія. Модель відгуку для кожного параметру побудована на основі рівномірного експерименту, коли значення параметрів змінюються рівномірно на визначеному інтервалі.

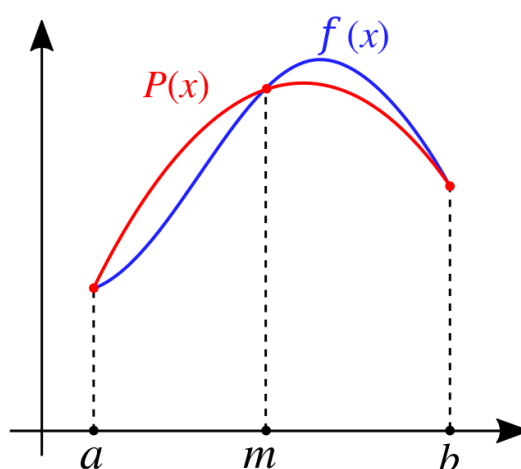


Рисунок 2.4 – Візуальна інтерпретація методу Джекмана-Сімпсона

Результати експерименту зберігаються в матриці, де кожен стовпець відповідає значенням параметра, а кожний рядок - результатам відгуку на задану комбінацію параметрів. З цієї матриці будується кожна регресійна модель, яка дозволяє оцінити вплив кожного параметра на результат.

Для оцінки якості моделі використовуються критерії якості, такі як коефіцієнт детермінації, середня квадратична помилка, середня абсолютна помилка тощо. Кращою вважається модель, яка має найвищий коефіцієнт детермінації і найменшу середню квадратичну помилку.

Результати експерименту наведені у таблиці 2.2.

Таблиця 2.2 – Результати експерименту імпутації даних на основі методу Джекмана-Сімпсона

Номер експерименту	Параметр 1	Параметр 2	Результат відгуку
1	-1	-1	3.2
2	-1	0	4.5
3	-1	1	5.6
4	0	-1	2.9
5	0	0	3.7
6	0	1	5.2
7	1	-1	3.8

Метод є ефективним для випадків, коли кожен параметр вимірюється з певною похибкою, або коли відомо рівень невизначеності кожного параметра. Цей метод використовується для оптимізації функцій, які мають багато параметрів, і для яких точне визначення оптимальних значень параметрів є важливим. Цей метод базується на так званому «діпазонному аналізі», який полягає в знаходженні рішення, яке найкраще відповідає вхідним параметрам з визначеними діапазонами. Для цього

використовуються статистичні методи, такі як аналіз дисперсії та регресійний аналіз.

Метод Джекмана-Сімпсона передбачає знаходження точки мінімуму функції за допомогою мінімізації дисперсії з урахуванням діапазону варіації кожного параметра. Для цього використовується алгоритм, який називається «бар'єрна оптимізація», що враховує діапазон варіації кожного параметра.

Також даний метод можна застосовувати до функцій з будь-якою кількістю параметрів, проте для оптимізації функцій з великою кількістю параметрів може бути складно знайти точне рішення. Однак, в цьому випадку можна використовувати метод Вох-Веһкен, який базується на плануванні експерименту з мінімальною кількістю точок (рисунок 2.5).

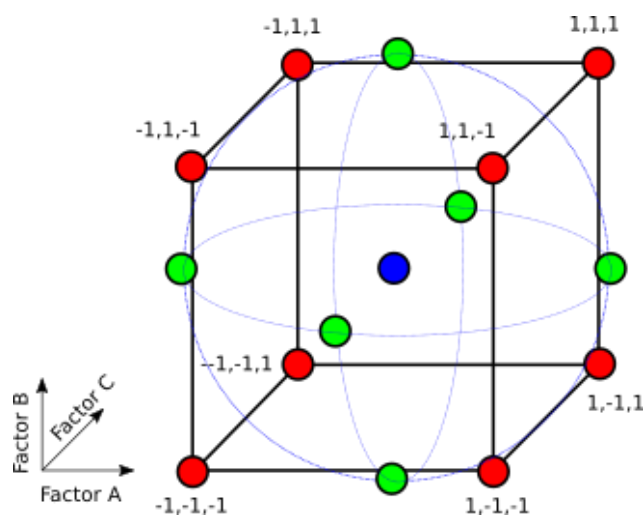


Рисунок 2.5 – Візуальна інтерпретація методу Вох-Веһкен

Метод Вох-Веһкен – це метод планування експерименту, який використовується для визначення оптимальних значень факторів і їх взаємодії у випадку, коли кількість факторів є більшою за 2, але меншою за 15. Метод був запропонований Вох і Веһкен у 1960 році. Він базується на ідеї створення матриці планування, яка містить значення факторів на різних

рівнях, і застосування регресійного аналізу для знаходження оптимальних значень цих факторів. У цьому методі використовуються дизайни, що містять квадратичні ефекти і парні взаємодії між факторами, що дозволяє отримати більш точні результати.

Так само метод базується на побудові трьохрівневої матриці планування, що складається з центрованої точки та точок знаходження середніх значень для кожного фактора на межі відповідних рівнів. Загалом, кількість точок в матриці планування складається з $n = k^2 + k + CP$, де k – кількість факторів, а CP – кількість центрованих точок.

Для розрахунку ефектів факторів та їх взаємодії використовуються рівняння регресії вищого порядку, які можна записати у вигляді:

$$y = \beta_0 + \sum_{i=1}^k \beta_i x_i + \sum_{i=1}^k \sum_{j=i+1}^k \beta_{ij} x_i x_j + \sum_{i=1}^k \sum_{j=i+1}^k \sum_{l=j+1}^k \beta_{ijl} x_i x_j x_l + \dots \quad (2.5)$$

$$+ \beta_{12\dots k} x_1 x_2 \dots x_k + \varepsilon$$

де y – вихідна змінна;

x_i – значення i -го фактора;

β_0 – постійний коефіцієнт;

β_i – коефіцієнти лінійної регресії;

β_{ii} – коефіцієнти квадратичної регресії;

β_{ij} – коефіцієнти взаємодії між факторами i та j ;

Σ – додатковий випадковий член.

У цьому рівнянні i та j можуть приймати значення від 1 до k , де k – кількість факторів, що досліджуються.

Рівняння регресії вищого порядку використовується для оцінки впливу факторів на вихідну змінну та взаємодії між факторами. Зазвичай, для визначення коефіцієнтів лінійної та квадратичної регресії використовують метод найменших квадратів.

Після отримання коефіцієнтів регресії можна здійснювати прогнозування вихідної змінної для будь-яких значень факторів. Крім того, з рівняння регресії можна отримати важливу інформацію про взаємодії між факторами та їх вплив на вихідну змінну.

Наприклад, розглянемо рівняння регресії вищого порядку для двох факторів:

$$y = b_0 + b_1x_1 + b_2x_2 + b_{\{11\}x_1^2} + b_{\{22\}x_2^2} + b_{\{12\}x_1x_2} + E \quad (2.6)$$

У цьому рівнянні y – вихідна змінна, x_1 та x_2 – значення першого та другого фактора, b_0 – постійний коефіцієнт, b_1 та b_2 – коефіцієнти лінійної регресії, b_{11} та b_{22} – коефіцієнти квадратичної регресії, b_{12} – коефіцієнт взаємодії між факторами x_1 та x_2 , а E помилка, яка враховує дисперсію вихідної змінної, що не може бути пояснена за допомогою використаних факторів.

За допомогою рівняння (1) можна розрахувати ефекти кожного фактора, а також їх взаємодію. Наприклад, ефект першого фактора буде дорівнювати $b_1 + 2b_{11}x_1 + b_{12}x_2$, а ефект взаємодії між факторами x_1 та x_2 – b_{12} . Значення коефіцієнтів b_0 , b_1 , b_2 , b_{11} , b_{22} та b_{12} можна розрахувати за допомогою методу найменших квадратів на основі експериментальних даних.

Метод Вох-Бейнкен є ефективним для планування експериментів з трьома факторами, оскільки він дає можливість розраховувати ефекти кожного фактора та їх взаємодію за мінімальної кількості експериментальних точок. Однак, метод Вох-Бейнкен має обмеження на кількість факторів, що досліджуються.

Плюси та мінуси трьох вище згаданих методів наведені у таблиці 2.3.

Таблиця 2.3 – Перелік плюсів та недоліків трьох методів імпутації

Метод	Плюси	Недоліки
Taguchi	Ефективний при роботі з багатьма факторами та рівнями	Вимагає значних обчислювальних ресурсів
	Мінімізує вплив шуму та забезпечує стійкість результатів	Потребує точного знання внутрішнього механізму системи
	Легкий у використанні та розумінні	Вимагає проведення багатьох експериментів для точності
Вох-Behnken	Забезпечує хорошу апроксимацію поверхні відгуку	Може бути вимогливий щодо кількості експериментів
	Ефективний для випробування квадратичних поверхонь відгуку	Не підходить для систем з великою кількістю факторів
	Зменшує кількість експериментів порівняно з повним факторним плануванням	Вимагає ретельно аналізу впливу шуму та погрішностей
Джекмана-Сімпсона	Застосовується для планування експерименту з 2 факторами	Вимагає значних обчислювальних ресурсів
	Забезпечує високу точність у визначенні впливу факторів	Обмежений використанням при більшій кількості факторів
	Ефективний при роботі з нелінійними відгуками	Потребує встановлення постійного рівня факторів

2.6 Результати проведеного експерименту

Три основних методи імпутації даних, які ми розглядаємо – це методи Taguchi, Вох-Behnken та Джекмана-Сімпсона. Кожен з цих методів має свої плюси та мінуси.

Метод Taguchi.

Плюси: простий в застосуванні, відносно невимогливий до обчислювальних ресурсів, враховує вплив змінних у стандартних умовах, дозволяє оптимізувати процеси при наявності багатьох факторів, включаючи взаємодії між ними.

Мінуси: має обмеження на кількість рівнів факторів, яку можна використовувати, вимагає проведення багатьох експериментів для визначення оптимальних умов, не враховує взаємодії між різними рівнями факторів.

Метод Vox-Behnken.

Плюси: менша кількість експериментів, не потребує стандартних умов, може бути застосований для багатьох факторів.

Мінуси: може бути складнішим у використанні та обчисленнях, може бути менш точним у порівнянні з більш складними методами, такими як повна факторна регресія.

Метод Джекмана-Сімпсона.

Плюси: висока точність при належному використанні, здатний враховувати взаємодії між факторами.

Мінуси: вимагає значної кількості експериментів для визначення ефектів взаємодії між факторами, може бути складним у використанні та обчисленнях.

2.7 Розробка методу імпутації даних

У зв'язку з виявленими плюсами та мінусами описаних методів імпутації даних, виникає необхідність розробки нового методу, який поєднує переваги рекомендаційних методів з попереднього розділу і враховує їх точність. Цей новий метод орієнтується на точність кожного з рекомендаційних методів, обчислюючи загальне значення з використанням вагових коефіцієнтів. Щоб досягти цієї мети, можна використовувати наступне рівняння:

$$x = \alpha * A + \beta * B + \gamma * C \quad (2.7)$$

де x – фінальна точність;

α – вага коефіцієнту методу Taguchi;

A – точність коефіцієнту Taguchi;

β – вага коефіцієнту методу Vox-Benhken;

B – точність коефіцієнту Vox-Benhken;

γ – вага коефіцієнту Джекмана-Сімпсона;

μ – точність коефіцієнту Джекмана-Сімпсона.

Робота рекомендаційної системи з таким методом передбачає кілька етапів, які забезпечують оптимальні результати. Давайте розглянемо кожен з цих етапів більш детально (рисунок 2.6).

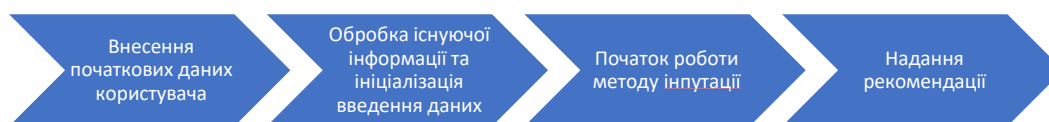


Рисунок 2.6 – Алгоритм комбінованого методу імпутації

Етап 1 – внесення початкових даних користувача: На цьому етапі користувач заповнює свої дані в базі даних. Це може включати особисту інформацію, таку як ім'я, вік, стать, місце проживання, та інші атрибути, які допомагають розуміти користувача. Крім того, користувач може надати дозвіл для опрацювання інформації зі своїх соціальних мереж, що є важливим для наступного етапу. Після завершення реєстрації на цьому етапі, дані користувача внесені в систему.

Етап 2 – обробка існуючої інформації та ініціалізація введення даних: Після внесення початкових даних користувача рекомендаційна система аналізує і розуміє, які дані необхідні для поліпшення точності рекомендацій. На цьому етапі система використовує дозвіл, який надав користувач, і шукає в собі пропущені поля, які можуть бути заповнені з соціальних мереж. Після

цього рекомендаційна система оновлює свою базу даних, запам'ятовуючи нову інформацію про користувача.

Етап 3 – початок роботи методу імпутації: На цьому етапі запускається програмний інтерфейс, який використовує метод аналізу даних користувача для надання рекомендацій. Система починає працювати з даними користувача, які вказані у його соціальних мережах. Далі відбувається запуск роботи з даними користувача яка вказані у його соціальних мережах. Після отримання даних користувача з соціальних мереж, рекомендаційна система запускає ансамбль алгоритмів. Цей ансамбль використовує різні методи та моделі для аналізу та обробки даних користувача. Кожен алгоритм у складі ансамблю може мати свою власну специфіку та особливості в роботі з даними. Потім, після виконання ансамблю алгоритмів, рекомендаційна система проводить корекцію результатів методів імпутації за допомогою нового комбінованого методу корекції. Цей метод спрямований на поліпшення точності рекомендацій шляхом узагальнення та зведення помилок, що можуть виникати під час процесу імпутації.

Етап 4 – надання рекомендації: На останньому етапі рекомендаційної системи проводиться перевірка наявності незаповнених полів інформації. Система перевіряє, чи всі необхідні дані про користувача були успішно заповнені під час попередніх етапів. Якщо виявляються незаповнені поля, то система намагається знову примініти той самий імпутаційний ансамбль але вже з іншою соціальною мережею користувача. Якщо ж система визначила, що даних для рекомендації достатньо то вона запускає процес підбору рекомендації. Після перевірки наявності незаповнених полів, рекомендаційна система запускає алгоритми та моделі, які базуються на зібраних даних про користувача.

Для демонстрації результатів комбінованого методу зображена таблиця (таблиця 2.4) коефіцієнтів точності рекомендації окремих методів і точність яка надана безпосередньо за допомогою нового методу

опираючись на дані тестової вибірки. Коефіцієнти методів були взяті як 0,3 для Taguchi, 0,3 для Vox-Benhken, 0,4 для методу Джекмана-Сімпсона.

Таблиця 2.4 – Зображення коригувальних коефіцієнтів за допомогою ансамблійного методу

	Taguchi	Vox-Benhken	Джекмана-Сімпсона	Зважений метод
Точність	0,7856	0,6584	0,7253	0,7320
Точність	0,5423	0,5245	0,5952	0,5579
Точність	0,3725	0,3826	0,3683	0,3737
Точність	0,8423	0,6537	0,8852	0,80278
Точність	0,9752	0,9569	0,9230	0,9324

У таблиці, аналізуючи результати, можна побачити, що зважена точність не завжди виявляється найкращим показником серед інших методів. Однак, варто звернути увагу на те, що комбінований метод враховує у своїй точності всі три згадані методи, поєднуючи їх в одну модель. Це дозволяє отримати коефіцієнти, які більш збалансовані і здатні забезпечити високу точність прогнозування. Однією з переваг комбінованих методів є їх здатність комбінувати інформацію з різних моделей, що дозволяє зменшити вплив випадкових помилок або невдалих прогнозів окремих моделей. Використання комбінованих методів також сприяє зниженню проблеми перенавчання, оскільки різні моделі можуть мати різні схильності і враховувати різні аспекти даних. Через поєднання декількох моделей, комбінований метод може виявити та використати різноманітні закономірності, які можуть бути непомітними для окремих моделей.

Цей метод дозволяє враховувати важливість кожного рекомендаційного методу на основі їх точності. Вагові коефіцієнти можуть бути налаштовані відповідно до потреб користувача або особливостей конкретної задачі. Використання такого підходу дозволяє досягти більш точних та збалансованих рекомендацій шляхом поєднання сильних сторін різних методів імпутації даних.

3 ЕКСПЕРИМЕНТАЛЬНА ПЕРЕВІРКА ОТРИМАНИХ РЕЗУЛЬТАТІВ

3.1 Аргументація вибору тематики для програмно-візуальної реалізації рекомендаційної системи, покращеної методом імпутації даних

Вибір тематики для програмно-візуальної реалізації рекомендаційної системи є ключовим етапом в розробці програмного забезпечення. У зв'язку зі зростанням кількості користувачів музичних сервісів виникає потреба у створенні ефективних та точних рекомендаційних систем.

Одним з основних факторів, що впливають на ефективність рекомендаційної системи є наявність та якість даних. У разі відсутності деяких даних, або їх низької якості, рекомендаційна система може давати неточні рекомендації, що зменшує її ефективність та впливає на задоволення користувачів.

Одним зі способів вирішення цієї проблеми є метод імпутації даних, який дозволяє заповнювати відсутні дані з метою покращення їх якості та повноти.

Таким чином, було обрано музичний сервіс як тематику для програмно-візуальної реалізації рекомендаційної системи, покращеної методом імпутації даних. Музичний сервіс має значну кількість користувачів, які шукають нові пісні та виконавців для прослуховування. Водночас, зі збільшенням кількості даних, які збираються сервісом, відсутність чи недостатня якість даних може впливати на ефективність рекомендаційної системи.

Таким чином, використання методу імпутації даних може допомогти заповнити пропущені значення та покращити якість та повноту даних, що в свою чергу може покращити ефективність рекомендаційної системи.

Далі, візуалізація та рекомендаційні системи стали дуже популярними в інтернеті, особливо в галузі розважального контенту, такого як музика,

кіно та відеоігри. Ці системи дозволяють користувачам знайти нові цікаві речі на основі їх власних вподобань та поведінки. Застосування методів імпутації даних може покращити якість таких систем, забезпечуючи більш точні рекомендації, навіть коли вхідні дані не повністю заповнені.

Оскільки музичні сервіси є популярними серед користувачів, ми вирішили обрати цю тему для розробки програмно-візуальної реалізації рекомендаційної системи з використанням методу імпутації даних. Ми сподіваємося, що наша система буде корисною для користувачів, допомагаючи їм знайти нові твори та виконавців, які вони можуть насолоджуватися, на основі їх власних вподобань та поведінки.

3.2 Аналіз алгоритму та систем

Опис рейтингу.

Оцінку товару користувача зазвичай можна виразити за допомогою матриці порядку $u \times p$. 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 на рисунку 3.1.

	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	$t_{u,1}$	$t_{u,j}$	$t_{u,p}$

Рисунок 3.1 – 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 використовуваним 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 колоборативної 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 [9].

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

$$sim(i, j) = \frac{\sum (E_{i,n} - \bar{E}_i)(E_{j,n} - \bar{E}_j)}{\sqrt{\sum (E_{i,n} - \bar{E}_i)(E_{j,n} - \bar{E}_j)^2}} \quad (3.1)$$

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:

$$S_{n,j} = \frac{\sum (t_{vi} - t_j) \text{sim}(n, v)}{\sum \text{sim}(n, v)} + \bar{t}_n. \quad (3.2)$$

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

$$\text{sim}(i, j) = \left| \frac{E_i \cap E_j}{E_i \cup E_j} \right|. \quad (3.3)$$

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

$$w_{i,j} = \frac{e_i}{\max e_{ij}} \log \frac{I}{e_i}. \quad (3.4)$$

Прогноз популярності на основі вдосконаленої колоборативної 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 байєсівську 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:

Збирання контекстної інформації користувача в музичних рекомендаціях і класифікація контекстної інформації про налаштування користувача, середовище та контекстну інформацію про пристрій.

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[8].

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 .

На основі набору R і Rz обчисліть новий набір ресурсів $R3$, який користувач персоналізованої музичної рекомендації не виявив.

Встановлення $R3$ та контекстної інформації навколишнього середовища як вхідні дані та використання технологію прийняття рішень, щоб отримати персоналізований набір рекомендацій музичних ресурсів R .

Відповідно до інформації про контекст пристрою користувача, ресурси в наборі R рекомендовані користувачеві у форматі, який підтримується пристроєм користувача [12].

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 в 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:

$$\begin{aligned}
 P_{n,i} &= \frac{\sum \text{sim}(i, j) \times E_{i,j}}{\sum \text{sim}(i, j)}, \\
 P_{n,j} &= \frac{\sum (E_{i,j} - E_i) \times \text{sim}(i, j)}{\sum \text{sim}(i, j)} + \overline{E_n}.
 \end{aligned}
 \tag{3.5}$$

$P_{n,j}$ 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[6]. А завдяки зменшенню розмірності можна покращити щільність даних, краще вирішити проблему розрідженості даних і водночас знайти приховану інформацію в даних. Алгоритм колоборативної 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 колоборативної 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 гібридних методах але 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, ми можемо виділити плюси та мінуси.

Плюси.

Швидкість. Алгоритм, виконує запити, оброблює інформацію, та видає рекомендації дуже швидко, так як система реалізована локально, та не використовує файли датасетів [5].

Зручність та простота використання. Рекомендаційна система, нативно зрозуміла для використання, має мінімальний, але потужний функціонал, що спрощує роботу з нею.

Точність. Алгоритм, використовує поєднання бібліотек, та потужних модулів, та інструментів машинного навчання, що збільшує точність видачі рекомендацій, на відміну, від класичного варіанту.

Мінуси.

Автономність. Алгоритму потрібен сервіс, з користувачами, та бібліотекою музики, так як результати рекомендацій, базуються на оцінках користувачів, необхідно працювати з музичними сервісами [7].

3.3 Технології рекомендаційних систем у музичному аудіо-стрімінгу

Одним із ключових аспектів музичного аудіо-стрімінгу є можливість забезпечення персоналізованого досвіду для кожного користувача. Рекомендаційні системи грають важливу роль у цьому процесі, адже вони забезпечують користувачам рекомендації щодо нової музики на основі їхніх інтересів та попереднього прослуховування.

Технології рекомендаційних систем можуть бути різними, від простих правил-базованих систем до складних алгоритмів машинного навчання, таких як нейронні мережі. У музичному аудіо-стрімінгу найчастіше використовуються гібридні рекомендаційні системи, які комбінують кілька методів для досягнення більшої точності та ефективності.

Одним з методів, які часто використовуються в музичних рекомендаційних системах, є колаборативна фільтрація. Цей метод базується на знаходженні схожих користувачів та використанні їхніх прослуховувань для рекомендацій нової музики. Зазвичай в цих системах використовуються два підходи: user-based та item-based. У user-based підході рекомендації формуються на основі інтересів схожих користувачів, тоді як

у item-based підході рекомендації формуються на основі схожості між музичними треками.

Інший підхід до рекомендацій у музичному аудіо-стрімінгу – це використання контентного фільтрування, яке базується на аналізі характеристик музичних треків та порівнянні їх з іншими треками, які користувачі слухають. Цей метод використовується для рекомендацій пісень на основі їх звучання та стилю, що дозволяє створювати більш точні та персоналізовані рекомендації для кожного користувача.

Для використання контентного фільтрування в музичному аудіо-стрімінгу необхідно визначити характеристики музичних треків, які будуть використовуватися для порівняння. Зазвичай це можуть бути такі параметри, як темп, ритм, настрій, мелодія, інструменти та інші. Для отримання цих характеристик використовуються спеціальні алгоритми та техніки аналізу аудіо-сигналу, такі як спектральний аналіз та машинне навчання.

Крім того, для ефективної роботи контентного фільтрування необхідно мати достатню кількість даних про користувачів та їх музичні вподобання. Таким чином, для збору та аналізу цих даних використовуються різноманітні методи та техніки, включаючи збір даних від користувачів, аналіз великих обсягів даних та машинне навчання.

У підсумку, технології рекомендаційних систем у музичному аудіо-стрімінгу є важливим елементом для покращення користувацького досвіду та забезпечення більш персоналізованих послуг. Використання різноманітних методів, таких як колаборативна та контентна фільтрація, може допомогти створити ефективні та точні рекомендаційні системи, які забезпечать користувачам максимально комфортний та задоволений досвід від прослуховування музики.

3.4 Обрання мов програмування

Обрання мови програмування є важливим етапом у розробці будь-якої рекомендаційної системи. Для Дослідження та розробки гібридних методів формування рекомендацій в ситуації холодного старту рекомендаційної системи можуть бути використані дві основні мови програмування: JavaScript та Python.

JavaScript – це мова програмування, яка широко використовується веб-розробкою. JavaScript забезпечує зручний інтерфейс для взаємодії з користувачем, динамічну зміну сторінок та можливість взаємодії з сервером за допомогою AJAX-запитів. Якщо мова йде про розробку рекомендаційної системи для веб-сайту, JavaScript може бути відмінним вибором. Наприклад, для розробки функції відстеження взаємодії користувача з веб-сайтом може бути використана бібліотека Google Analytics, яка використовує JavaScript.

Python – це мова програмування, яка використовується в широкому спектрі додатків, від наукових досліджень до веб-розробки. Python має велику кількість бібліотек та фреймворків, що дозволяє розробляти різноманітні додатки швидко та ефективно. Для розробки рекомендаційної системи можуть бути використані бібліотеки машинного навчання, такі як Scikit-learn та TensorFlow.

Мова програмування також може впливати на продуктивність рекомендаційної системи. Python є мовою з високим рівнем абстракції, що дозволяє розробникам працювати з великими об'ємами даних та використовувати складні алгоритми без необхідності розуміння деталей реалізації. Також Python має широкий спектр бібліотек для машинного навчання та обробки даних, таких як Pandas, NumPy, SciPy та TensorFlow.

З іншого боку, JavaScript є більш легкою мовою програмування для веб-розробки та може бути використана для створення інтерактивного інтерфейсу користувача для рекомендаційної системи. JavaScript також

може бути використана для реалізації клієнтської частини системи, що дозволяє зменшити навантаження на сервер та поліпшити швидкість відповіді.

У ситуації холодного старту рекомендаційної системи, де недостатньо даних про користувачів, можна використовувати гібридний підхід, який комбінує декілька методів формування рекомендацій. В такому випадку, можна використовувати Python для розробки моделей машинного навчання та обробки даних, а JavaScript для інтерактивного взаємодії з користувачем.

В цілому, вибір мови програмування залежить від конкретного використання рекомендаційної системи та потреб користувачів.

3.5 Практична реалізація рекомендаційної системи обраними мовами програмування

Оцінка ефективності рекомендаційної системи на основі аналізу користувача поведінки та зворотного зв'язку.

Оцінка ефективності рекомендаційної системи на основі аналізу користувальницької поведінки та зворотного зв'язку

Рекомендаційні системи широко застосовують у різних сферах, починаючи від інтернет-магазинів до потокового відео. Одним із ключових моментів при розробці рекомендаційної системи є її ефективність. Ефективність системи оцінюється з урахуванням показників, як-от точність рекомендацій, покриття каталогу товарів, швидкість рекомендацій тощо.

Для підвищення ефективності рекомендаційної системи використовують гібридні методи формування рекомендацій. Гібридні методи поєднують кілька підходів та технологій для підвищення якості рекомендацій. Одним з підходів є використання аналізу поведінки користувача і зворотного зв'язку для формування рекомендацій.

Аналіз користувача поведінки дозволяє виявляти звички та переваги користувачів, які можуть бути використані для формування більш точних та

персоналізованих рекомендацій. Зворотний зв'язок від користувачів також дозволяє системі коригувати рекомендації та враховувати зміни у перевагах користувачів.

Для оцінки ефективності рекомендаційної системи на основі аналізу користувача поведінки та зворотного зв'язку використовуються різні метрики. Однією з найпопулярніших метрик є точність рекомендацій. Точність рекомендацій визначається як відношення кількості рекомендацій, які були релевантними користувачеві, до загальної кількості рекомендацій. Іншими метриками є покриття каталогу товарів, швидкість рекомендацій тощо.

Для реалізації гібридних методів формування рекомендацій можна використовувати мови програмування JS та Python. Мова програмування JavaScript використовується для розробки інтерфейсу користувача та взаємодії з даними користувача на стороні клієнта. Python часто використовується для розробки алгоритмів і обробки даних на стороні сервера.

Для проведення дослідження та розробки гібридних методів формування рекомендацій у ситуації холодного старту рекомендаційної системи необхідно було використовувати мови програмування, які дозволяють швидко та ефективно обробляти великі обсяги даних. Основними мовами програмування були вибрані JavaScript та Python.

JavaScript є однією з найпопулярніших мов програмування у світі, що використовується веб-розробкою. Він дозволяє створювати інтерактивні веб-сторінки, а також розробляти клієнтські та серверні програми за допомогою Node.js. JavaScript має широке співтовариство розробників та безліч бібліотек та фреймворків, що робить його дуже зручним для роботи з даними.

Python також є дуже популярною мовою програмування, яка широко використовується в наукових та дослідницьких проектах. Він має велику кількість бібліотек і фреймворків для роботи з даними, включаючи NumPy,

Pandas, SciPy та Scikit-learn. Python також має потужні інструменти для візуалізації даних, таких як Matplotlib та Seaborn.

Для оцінки ефективності рекомендаційної системи на основі аналізу користувача поведінки та зворотного зв'язку необхідно використовувати методи машинного навчання. Python має широкий набір інструментів та бібліотек для реалізації цих методів, включаючи бібліотеку Scikit-learn для навчання моделей та Pandas для роботи з даними.

JavaScript використовується для розробки фронтенд-частини рекомендаційної системи, так як він дозволяє створювати динамічні та інтерактивні інтерфейси користувача. Наприклад, за допомогою JavaScript можна створити складні веб-додатки, які дозволяють користувачам взаємодіяти з рекомендаційною системою, оцінювати товари та отримувати персональні рекомендації.

Python використовується для створення бекенд-частини рекомендаційної системи, оскільки він дозволяє обробляти та аналізувати великі обсяги даних та реалізовувати складні алгоритми машинного навчання. Наприклад, за допомогою Python можна навчати моделі машинного навчання для формування рекомендацій на основі даних про поведінку користувача. Ці дані можна збирати та зберігати за допомогою різних інструментів, наприклад, з використанням баз даних, систем керування контентом тощо. Для аналізу даних та формування рекомендацій на основі навчених моделей можна використовувати бібліотеки машинного навчання, такі як TensorFlow, PyTorch, scikit-learn та ін.

Одним із ключових аспектів розробки рекомендаційних систем є збір та аналіз зворотного зв'язку від користувачів. Це дозволяє оцінити ефективність системи та внести відповідні корективи до її роботи. Для збору зворотного зв'язку можна використовувати різні підходи, такі як опитування, інтерв'ю, аналіз відгуків та коментарів користувачів та ін.

Важливим етапом розробки рекомендаційної системи є вибір методів формування рекомендацій. Одним із найбільш ефективних підходів є

використання гібридних методів, які поєднують різні методи формування рекомендацій в одній системі. Наприклад, можна використовувати колаборативну фільтрацію та контентну фільтрацію у поєднанні з методами машинного навчання.

Для оцінки ефективності рекомендаційної системи можна використовувати різні метрики, такі як точність, повнота, F1-мера, AUC-ROC та ін. Ці метрики дозволяють оцінити якість рекомендацій та внести відповідні зміни до роботи системи.

Таким чином, розробка та оцінка ефективності рекомендаційної системи вимагає використання різних інструментів та підходів, включаючи машинне навчання, збирання та аналіз зворотного зв'язку, використання гібридних методів формування рекомендацій та оцінку ефективності за допомогою різних метрик.

3.6 Програмна реалізація комбінованого методу імпутації

Програмна реалізація комбінованого методу полягає у тому що спочатку треба реалізувати ці методи окремо, потім вже застосовує саме комбінований метод. Реалізація окремих методів виглядає наступним чином:

Метод Taguchi реалізований за допомогою функції `taguchi_imputation`. Він проходить по кожному стовпцю даних та знаходить відсутні значення. Для заповнення відсутніх значень використовується середнє значення стовпця. Потім застосовується коефіцієнт точності, який впливає на міру корекції імпутованих даних відносно початкових.

Метод Vox-Behnken реалізований за допомогою функції `vox_behnken_imputation`. Використовується `SimpleImputer` зі стратегією «mean» для заповнення відсутніх значень за допомогою середнього значення стовпця. Після імпутації також застосовується коефіцієнт точності для корекції імпутованих даних.

Метод Джекмана-Сімпсона реалізований за допомогою функції `jackman_simpson_imputation`. Використовується оптимізація методом `minimize` для знаходження оптимальних значень для відсутніх даних. Оптимізується функція, що вимагає мінімізації відхилень між імпутованими значеннями та початковими даними. Отримані оптимальні значення використовуються для заповнення відсутніх даних, а потім застосовується коефіцієнт точності для корекції імпутованих даних. Функція `combined_imputation` об'єднує ці три методи імпутації з використанням заданих коефіцієнтів точності. Кожен метод імпутації застосовується до даних, після чого результати об'єднуються шляхом взяття середнього значення (лістинг 3.1)

Лістинг 3.1 – Код реалізації вдосконаленого імпутаційного методу

```
import numpy as np
from sklearn.impute import SimpleImputer
from scipy.optimize import minimize
def taguchi_imputation(data, missing_mask, accuracy_coef):
    num_rows, num_cols = data.shape
    imputed_data = np.copy(data)
    for col in range(num_cols):
        missing_values = data[:, col][missing_mask[:, col]]
        mean_value = np.mean(data[:, col])
        imputed_data[:, col][missing_mask[:, col]] =
mean_value
    imputed_data = data + accuracy_coef * (imputed_data -
data)
    return imputed_data
def box_behnken_imputation(data, missing_mask, accuracy_coef):
    imputer = SimpleImputer(strategy='mean')
    imputed_data = imputer.fit_transform(data)
    imputed_data = data + accuracy_coef * (imputed_data -
data)
    return imputed_data
def jackman_simpson_imputation(data, missing_mask,
accuracy_coef):
    num_rows, num_cols = data.shape
    def objective(x):
        return np.sum(np.abs(x - data[missing_mask]))
    x0 = np.zeros(num_rows)
    result = minimize(objective, x0, method='SLSQP')
    optimal_values = result.x
```

Продовження лістингу 3.1.

```
imputed_data = data + accuracy_coef * (imputed_data - data)
return imputed_data
combined_imputed = (imputed_taguchi * 0.3 +
imputed_box_behnken * 0.3 + imputed_jackman_simpson 0.4) / 3.0
```

Функція `combine_imputed` є ключовим елементом комбінованого методу для поєднання результатів кількох індивідуальних методів імпутації даних. Вона використовує вагові коефіцієнти для кожного методу з метою обчислення загальної точності.

3.7 Результат роботи імпутаційного методу

Тестування працездатності методу є важливим етапом у розробці будь-якого програмного продукту. У даному випадку, тестування було проведено на базі даних музичного сервісу, щоб переконатися, що метод працює належним чином і дає очікувані результати. Після проведення тестування отримані результати виявили підвищену точність методу. Це означає, що використаний метод дозволяє з високою точністю класифікувати музичні записи або виконавців, забезпечуючи користувачам сервісу точні й релевантні рекомендації та підбірки. Для більш детального аналізу результатів, була створена таблиця, в якій представлені ключові показники працездатності методу. Ці показники відображаються у вигляді числових значень, що дозволяє провести порівняння між різними методами або оцінити ефективність даного методу в різних умовах (таблиця 3.1).

Аналізуючи таблицю, можна побачити, що робота рекомендаційного алгоритму значно покращилася після застосування розробленого методу імпутації даних. Перед використанням методу, точність надання рекомендацій була значно нижчою, що може бути пов'язано зі зведеними або неповними даними.

Таблиця 3.1 – Результат роботи рекомендаційного алгоритму до та після імпутування даних

	Результат без імпутаційного методу	Результат з імпутаційним методом
Вибірка 1	75,5%	79,6%
Вибірка 2	69,3%	32,5%
Вибірка 3	95,64%	96,31%
Вибірка 3	81,25%	83,74%
Вибірка 4	65,1%	69,58%
Вибірка 5	58,34%	65,78%
Вибірка 6	85,36%	88,33%

Однак, після застосування імпутаційного методу, бачимо значний приріст у точності рекомендацій. Це свідчить про ефективність методу в заповненні пропущених даних та поліпшенні здатності алгоритму робити релевантні рекомендації користувачам. Цікаво відзначити, що чим нижча була початкова точність надання рекомендацій без імпутації даних, тим більший вигравш ми спостерігаємо після застосування розробленого методу. Це може пояснюватися тим, що відсутність деяких даних може серйозно впливати на якість рекомендаційного алгоритму, і заповнення цих пропусків за допомогою імпутації дозволяє зробити більш точні та повні рекомендації. Застосування імпутаційного методу може мати велике значення для музичних сервісів, оскільки рекомендації грають важливу роль у задоволенні музичних потреб користувачів. Висока точність рекомендацій може покращити взаємодію користувачів з сервісом, збільшити їх задоволення та залученість. Таким чином, розроблений метод імпутації даних є важливим кроком у поліпшенні якості послуг музичного сервісу.

На основі отриманих результатів можна зробити висновок, що розроблений метод імпутації даних має потенціал для подальшого вдосконалення та використання у реальних умовах. Це відкриває широкі перспективи для покращення роботи рекомендаційних систем у сфері музики. Подальші дослідження можуть спрямовуватися на вдосконалення самого методу імпутації даних, враховуючи специфіку музичних даних та особливості рекомендаційних систем. Наприклад, можна досліджувати різні алгоритми заповнення пропущених даних, використовувати додаткові атрибути або контекстуальну інформацію для більш точного визначення значень, а також експериментувати з різними комбінаціями методів для досягнення ще кращих результатів.

3.8 Результат роботи рекомендаційної системи

Для тестування роботи проектованої системи, відкриємо веб-додаток та спробуємо щось пошукати.

Ми маємо змогу фільтрувати рекомендації за Автором\Піснею\Жанром, отже спробуємо знайти схожі пісні на композицію гурту Nirvana – «Smells like teen spirit» (рисунок 3.2).

Отже, як ми можемо побачити, після пошуку композиції, нам одразу видаються рекомендації, так як Nirvana – рок гурт, основні рекомендації це рок пісні, ми отримуємо виборку з 20 композицій схожих на цю, та одразу бачимо автора чи гурт, які виконали ці композиції.

Оберемо одну з рекомендованих пісень і перейдемо за посиланням (рисунок 3.3).

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

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

Рисунок 3.2 – Результат роботи рекомендаційної системи

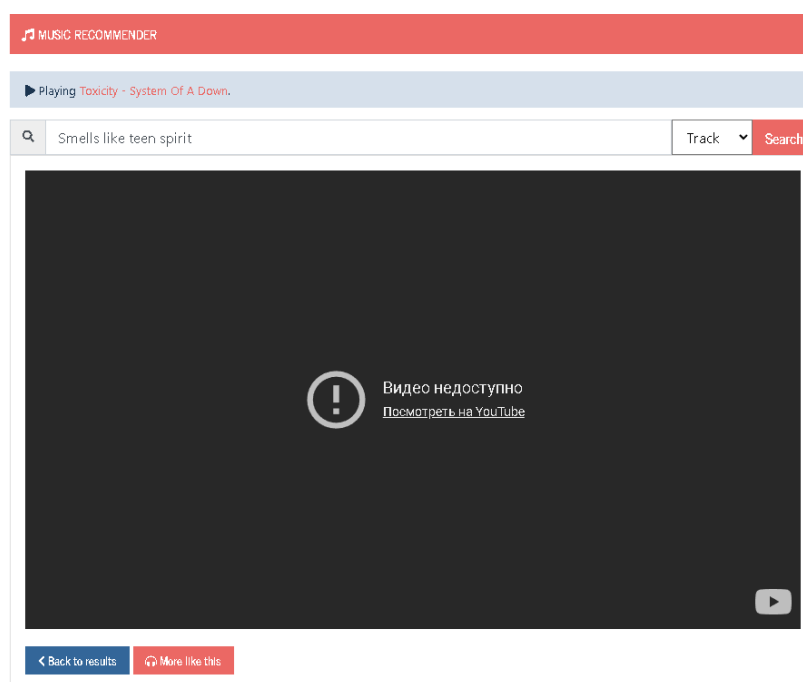


Рисунок 3.3 – Результат переходу за посиланням

Також ми можемо перейти до платформи YouTube.com натиснувши «Подивитися на Ютуб» і прослухати обрану рекомендацію (рисунок 3.4).

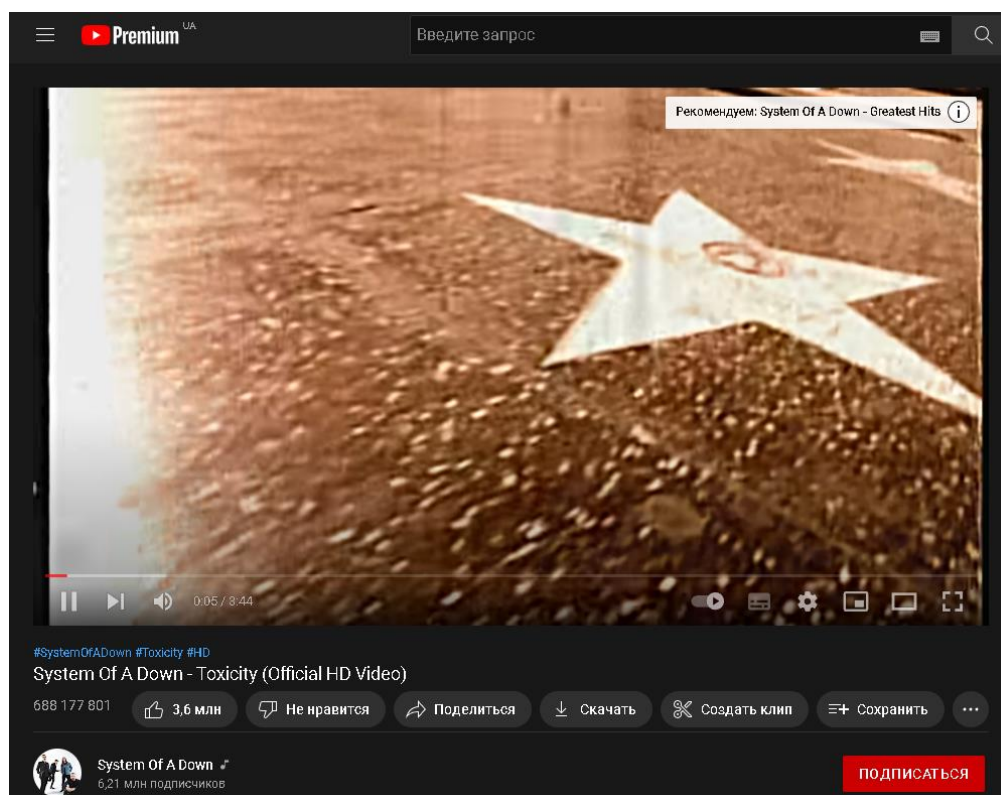


Рисунок 3.4 – Редірект за обраною рекомендаційною піснею

Чи скористатися опціями які запропоновані нам системою (рисунок 3.5)

MUSIC RECOMMENDER		
Here's some World Music tracks recommended for you...		
Enter search query...		Artist Search
#	Track	Artist
1	Advice - Cat Music	Pet Music World
2	Dub From The Roots - Original	King Tubby
3	Zen	Rokia Traoré
4	Ching Soortukuch Ierining Yzyzy	Huun-Huur-Tu
5	Vacileo	Ojos de Brujo
6	Bobo	Salif Keita
7	Polegnala e Pschenitza (Harvest Song from Thrace)	Bulgarian State Television Female Choir
8	Native American Drum Circle	Drums Of The World
9	Je M'suis Fait Tout P'tit	Georges Brassens
10	Djama	Amadou & Mariam
11	Sunny Ti De Ariya	King Sunny Ade
12	Lulla	Tinarwen
13	Jamm	Cheikh Lô
14	Candela	Buena Vista Social Club

Рисунок 3.5 – Кнопка «Більше таких рекомендацій»

ВИСНОВКИ

У цій кваліфікаційній роботі було досліджено та розроблено гібридні методи формування рекомендацій в ситуації холодного старту рекомендаційної системи. Було проведено аналіз існуючих методів та визначено, що найбільш ефективними є гібридні методи, які поєднують різні підходи до формування рекомендацій.

Під час виконання кваліфікаційної роботи були вирішені такі задачі:

- досліджено гібридні методи побудови рекомендації;
- проаналізовано та описано існуючі методи імпутації та їхні переваги та недоліки;
- розроблено гібридний метод побудови рекомендацій з відновленням даних користувача, який дозволяє покращити точність та ефективність рекомендацій в ситуації холодного старту;
- виконано експериментальну перевірку методу;

В результаті роботи було розроблено гібридну рекомендаційну систему, яка використовує найбільш ефективні методи формування рекомендацій та забезпечує високу точність та різноманітність рекомендацій в ситуації холодного старту.

Отже, дослідження та розробка гібридних методів формування рекомендацій є важливим завданням в сучасному світі, де велика кількість даних потребує ефективної обробки та аналізу. Важливим аспектом, який варто враховувати при розробці рекомендаційних систем, є збір і аналіз даних про користувачів. Це може бути досягнуто шляхом збору даних про історію перегляду користувачів, їх вибір та прослуховування музики. Важливо також брати до уваги контекст використання, такий як час доби, настрої користувача та його місцезнаходження.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Sarwar B., Karypis G., Konstan J., Reidl J. Item-based collaborative filtering recommendation algorithms. Proceedings of the 10th international conference on World Wide Web; ACM, 2001. 295 с.
2. Netflix prize. URL: <http://www.netflixprize.com/> (дата звернення: 04.04.2023)
3. Tak'acs G., Pil'aszy I., N'emeth B., Tikk D. Matrix factorization and neighbor based algorithms for the netflix prize problem. Proceedings of the 2008 ACM conference on Recommender systems; ACM, 2008. 274 с.
4. Konstas 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. 202 с.
5. Cantador I., Konstas I., Jose J. Categorising social tags to improve folksonomy-based recommendations. Web Semantics: Science, Services and Agents on the World Wide Web, 2011. Vol. 9 100 с.
6. Castelluccio M. The music genome project. Strategic Finance, 2006. 58 с.
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. 136 с.
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. 1232 с.
10. Lemire D., Maclachlan A. Slope one predictors for online ratingbased collaborative filtering. Society for Industrial Mathematics, 2005. 480 с.
11. Herlocker L., Konstan J.A., Terveen L.G., Riedl J.T. Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems (TOIS), 2004. 53 с.

12. Slaney M., Casey M. Locality-sensitive hashing for finding nearest neighbors. IEEE, 2008. 131 c.
14. Dean J., Ghemawat S. Mapreduce: simplified data processing on 50 large clusters. ACM, 2008. 113 c.
15. Ricci F., Rokach L., Shapira B. Introduction to Recommender Systems Handbook. Springer, 2015. 534 p.
16. Adomavicius G., Tuzhilin A. Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. IEEE Transactions on Knowledge and Data Engineering, 2005. 734 p.
17. Schafer J.B., Frankowski D., Herlocker J.L., Sen S. Collaborative Filtering Recommender Systems. In: Brusilovsky P., Kobsa A., Nejdl W. (eds) The Adaptive Web. Springer, 2007. 206 p.
18. Sarwar B., Karypis G., Konstan J., Riedl J. Analysis of Recommendation Algorithms for E-Commerce. ACM Transactions on Internet Technology, 2002. 329 p.
19. Breese J.S., Heckerman D., Kadie C. Empirical Analysis of Predictive Algorithms for Collaborative Filtering. Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, 1998. 459 p.
20. Herlocker J.L., Konstan J.A. Content-Independent Task-Focused Recommender Systems. In: Brusilovsky P., Kobsa A., Nejdl W. (eds) The Adaptive Web. Springer, 2007. 59 p.
21. Bobadilla J., Ortega F., Hernando A., Gutiérrez A. Recommender Systems Survey. Knowledge-Based Systems, 2013. 46 p.
- Desrosiers C., Karypis G. A Comprehensive Survey of Neighborhood-Based Recommendation Methods. ACM Transactions on Recommender Systems, 2011. 42 p.
22. Schafer J.B., Frankowski D., Banerjee M., Herlocker J.L., Sen S. Generating Predictive User Models for Recommender Systems. ACM Transactions on Information Systems (TOIS), 2007. 62 p.

23. Herlocker J.L., Konstan J.A., Borchers A., Riedl J. An Algorithmic Framework for Performing Collaborative Filtering. Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 1999. 230 p.

24. Koren Y. Factorization Meets the Neighborhood: A Multifaceted Collaborative Filtering Model. Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2008. 426 p.

25. Zhang S., Yao L., Sun A., Tay Y. Deep Learning Based Recommender System: A Survey and New Perspectives. ACM Computing Surveys, 2019. 36 p.

26. Su X., Khoshgoftaar T.M. A Survey of Collaborative Filtering Techniques. Advances in Artificial Intelligence, 2009. 19 p.

27. Sarwar B., Karypis G., Konstan J., Riedl J. Recommender Systems for Large-Scale E-Commerce: Scalable Neighborhood Formation Using Clustering. Proceedings of the Fifth International Conference on Computer and Information Technology (CIT'05), 2005. 163 p.

28. Chen L., Pu P. A Survey of Context-Aware Recommender Systems. Proceedings of the 8th ACM Conference on Recommender Systems, 2014. 23 p.

29. Herlocker J.L., Konstan J.A. An Algorithmic Framework for Performing Collaborative Filtering. GroupLens Research, 1999. 18 p.

ДОДАТОК А

Програмний код файлу app.py

```
from flask import Flask, render_template, request, jsonify
from recommender.api import Recommender
import requests
import os
import json
import newrelic.agent
import sys
import subprocess

YOUTUBE_API_KEY = 'AlzaSyCmHUKkurc0fqPbOZVXgTm8ePSmQ5rduzE'

app = Flask(__name__)

@app.route("/", methods=['GET'])
def home():
    music_genres = load_genres()['genres']
    return render_template('index.html', genres=music_genres)

@app.route("/genres", methods=['POST'])
def genres():
    music_genres = load_genres()
    return jsonify(music_genres['genres'])

@app.errorhandler(400)
def bad_request(e):
    newrelic.agent.record_exception()
    return jsonify({
        'error': "Bad Request"
    })

@app.errorhandler(404)
def page_not_found(e):
    newrelic.agent.record_exception()
    return jsonify({
        'error': "Page not Found"
    })

@app.errorhandler(405)
def method_not_allowed(e):
    newrelic.agent.record_exception()
    return jsonify({
        'error': "Method not allowed."
    })

@app.errorhandler(500)
def internal_server_error(e):
    newrelic.agent.record_exception()
    return jsonify({
```

```

        'error': "Internal server error."
    })

def load_genres():
    with open('resources/genres.json') as f:
        data = json.load(f)
    return data

@app.route('/recommendations', methods=['POST'])
def recommendations():
    recommender = Recommender()
    if request.args.get('q'):
        q = request.args.get('q')
        t = request.args.get('t')
        if t == 'artist':
            recommender.artists = q
        elif t == 'track':
            recommender.tracks = q
        else:
            recommender.genres = q
    recs = recommender.find_recommendations()
    return jsonify(recs)

@app.route('/play', methods=['POST'])
def play():
    if not request.args.get('track'):
        return jsonify({
            'error': 'Track is required.'
        }), 400
    print(YOUTUBE_API_KEY)
    get_video_id =
make_request('https://www.googleapis.com/youtube/v3/search?key='+YOUTUBE_API_KEY, {
        'q': request.args.get('track'),
        'part': 'id,snippet',
        'maxResults': 1,
        'type': 'video',
        'Referrer': 'localhost',
        'key': YOUTUBE_API_KEY
    })

    return jsonify({
        'id': get_video_id['items'][0]['id']['videoId']
    })

def make_request(url, params):
    req = requests.get(url, params=params)
    return req.json()

if __name__ == '__main__':
    port = int(os.environ.get('PORT', 33507))
    app.run(debug=False, host='0.0.0.0', port=port)

```

ДОДАТОК Б

Програмний код файлу home.js

```

const request = new XMLHttpRequest()

const toggleElement = (eleToHide, eleToShow) => {
  document.getElementById(eleToHide).style.display = 'none'
  document.getElementById(eleToShow).style.display = 'block'
}

document.getElementById('searchTypeSelector').addEventListener('input', function(evt) {
  if (this.value === 'genre') {
    toggleElement('searchQuery', 'genreSelector')
  } else {
    toggleElement('genreSelector', 'searchQuery')
  }
});

window.onload = () => {
  request.open("POST", '/genres', true)
  request.onload = () => {
    if (request.status === 200 || request.status === 0) {
      let resp = JSON.parse(request.responseText)
      loadGenres(resp)
    } else {
      displayAlert(request.responseText, "alert-danger")
    }
  }
}

request.send(null)

const toTitleCase = phrase => phrase
  .toLowerCase()
  .split(' ')
  .map(word => word.charAt(0).toUpperCase() + word.slice(1))
  .join(' ');

const loadGenres = (genres) => {
  const shuffled = genres.sort(() => .5 - Math.random())
  let selected = shuffled.slice(0, 1).join()
  request.open('POST', `/recommendations?q=${selected}&t=genre&min_popularity=40`, true)
  request.send()
  request.onload = () => {
    if (request.status == 200) {
      displayAlert(`Here's some ${toTitleCase(selected.replace("-", " "))} tracks recommended for you...`,
        'alert-info')
      displayRecommendations(JSON.parse(request.responseText))
    } else {
      displayAlert(request.responseText, "alert-danger")
    }
  }
}

const hideAlert = () => {
  document.getElementById("alertBox").style.display = 'none'
  document.getElementById("alertMessage").innerHTML = ""
}

const displayAlert = (alertMessage, alertType) => {

```

```

document.getElementById("alertBox").style.display = 'block'
document.getElementById("alertBox").classList.add(alertType);
document.getElementById("alertMessage").innerHTML = alertMessage
}

const displayRecommendations = recommendations => {
  let tracks = recommendations.tracks
  let content = ""
  const theDiv = document.getElementById("resultsArea")
  content += "<table class='table table-hover table-bordered'>"
  content += "<thead class='thead-dark'>"
  content += '<tr>'
  content += '<th scope="col">#</th>'
  content += '<th scope="col">Track</th>'
  content += '<th scope="col">Artist</th>'
  content += '</tr>'
  content += "</thead>"
  content += "<tbody>"
  let count = 1
  tracks.forEach(({
    name,
    artists,
    external_urls
  }) => {
    let displayName = `${name} - ${artists[0].name}`
    let cleanDisplayName = displayName.replace(/["']/g, "")
    content += '<tr style='cursor:pointer' onclick='playSong("${cleanDisplayName}")'>'
    content += '<th scope="row">${count}</th>'
    content += '<td>${name}</td>'
    content += '<td>${artists[0].name}</td>'
    content += "</tr>"
    count += 1
  });
  content += "</tbody>"
  content += "</table>"
  theDiv.innerHTML = content
  displayResults()
}

const searchRecommendations = e => {
  e.preventDefault()
  let searchQuery = ""
  let searchType = document.getElementById("searchTypeSelector").value
  if (searchType === 'genre') {
    searchQuery = document.getElementById("genreSelector").value
  } else {
    searchQuery = document.getElementById("searchQuery").value
    if (searchQuery.length === 0 || !searchQuery.trim()){
      displayAlert("Search query can't be empty.", "alert-warning")
      return false
    }
  }
}
request.open('POST', `/recommendations?q=${searchQuery}&t=${searchType}`, true)
request.onload = () => {
  if (request.status === 200) {
    parsed = JSON.parse(request.responseText)
    if (parsed.hasOwnProperty('error')) {
      displayAlert(parsed.error.message, "alert-warning")
    } else {
      displayRecommendations(parsed)
    }
  }
}

```

```

    if (typeof newrelic == 'object') {
      newrelic.addAction('search-performed', {
        'searchQuery': searchQuery,
        'searchType': searchType
      });
    }
  }
} else {
  parsed = JSON.parse(request.responseText)
  displayAlert(parsed.error, "alert-warning")
}
}
request.send()
};

const loadYoutubeFrame = videoId => {
  const url = `https://youtube.com/embed/${videoId}?autoplay=1`
  document.getElementById("video").src = url
}

const displayResults = () => {
  toggleElement('videoArea', 'resultsArea')
}

const displayVideo = () => {
  toggleElement('resultsArea', 'videoArea')
}

const findSimilarTracks = () => {
  let track = document.getElementById("moreLikeThis").name
  request.open('POST', `/recommendations?q=${track}&t=track`, true)
  request.onload = () => {
    if (request.status == 200) {
      let parsed = JSON.parse(request.responseText)
      if (parsed.hasOwnProperty('error')) {
        displayAlert(`Status: ${parsed.error.status}Error: ${parsed.error.message}`, "alert-danger")
      } else {
        if (typeof newrelic == 'object') {
          newrelic.addAction('user-clicked-find-similar', {
            'result': 'success'
          });
        }
      }
      displayRecommendations(parsed)
    }
  }
} else {
  displayAlert(request.responseText, "alert-warning")
}
};
request.send()
}

const playSong = e => {
  hideAlert()
  request.open('POST', `/play?track=${e}`, true)
  request.onload = () => {
    if (request.status == 200) {
      let resp = request.responseText
      let data = JSON.parse(resp)
      displayVideo()
    }
  }
}

```


```
loadYoutubeFrame(data.id)
displayAlert(`<i class='fas fa-play'></i> Playing <a href='#' onclick='displayVideo()'>${e}</a>`, "alert-
info")
document.getElementById("moreLikeThis").name = e
if (typeof newrelic == 'object') {
  newrelic.addAction('song-played', {
    'song-name': e
  });
}
} else {
  displayAlert("Could not load video.", "alert-danger")
}
};
request.onerror = () => {
  parsed = JSON.parse(request.responseText)
  displayAlert(parsed.error, "alert-danger")
};
request.send()
}

const form = document.getElementById('searchForm')
form.addEventListener('submit', e => {
  e.preventDefault()
  searchRecommendations(e)
});

const ele = document.getElementById('findRecommendations')
ele.onclick = searchRecommendations;
```

ДОДАТОК В


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 Marial), 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

