«Музика-бот»
(шифр)

«РОЗРОБКА РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ З ПІДБОРУ КЛАСИЧНОЇ МУЗИКИ ЗА ВИКОНАВЦЕМ»

Галузь:
Прикладна лінгвістика

2020/2021

АНОТАЦІЯ

В даній роботі описано принципи побудови систем рекомендацій, їх порівняння, алгоритм та методи. Завданням було створити рекомендаційну систему на основі корпусу коментарів про виконавців музичних творів.

Для аналізу коментарів щодо виконавців було обрано Другий концерт С. Рахманінова для фортепіано з оркестром та корпус записів з виконанням різних піаністів у YouTube.

Актуальність теми полягає в тому, що коментарі користувачів є невід'ємною частиною рекомендацій.

Мета роботи: ознайомитися з основними рекомендаційними системами музичних творів, розробити алгоритм рекомендації та описати головні методи для роботи з даними.

Об'єкт роботи — розробка програмного засобу для рекомендацій виконавців музичних творів з використанням мови Python у середовищі програмування PyCharm, що забезпечує простоту використання функціональних можливостей та легкість програмної частини.

Предметом роботи є дослідження роботи з рекомендаційними системами — за допомогою засобів мови програмування Python.

Для реалізації визначеної мети є наступні задачі:

- вивчення потрібної літератури на тему курсової роботи;
- формулювання основних понять, що стосуються заданої теми;
- практична реалізація проекту.

Як теоретичну базу для проекту було використано літературу, статті та сайти, рекомендаційні системи музичних творів, записи в YouTube. Практичною базою є у середовище програмування PyCharm.

Робота складається зі вступу, двох розділів, висновків, списку літератури, який містить 19 джерел. У вступі описується актуальність задачі розробки рекомендаційних систем та використання статистичних методів для вирішення даної задачі. У першому розділі описано основні принципи побудування

порівняльних систем, їх методику, проблеми. Описано існуючі рекомендаційних систем музичних творів. У другому розділі обирається алгоритм рекомендацій, описано головні методи для рекомендацій виконавців.

Роботу було апробовано на XXVIII Міжнародної науково-практичної конференції «Інформаційні технології: наука, техніка, технологія, освіта, здоров'я» (MicroCAD-2020) (м. Харків, 28-30 жовтня 2020 р.). – Харків : НТУ «ХПІ», 2020.

Ключові слова: рекомендаційна система, корпус, коментарі, виконавці музичних творів, порівняльні системи, методи рекомендації.

ЗМІСТ

ВСТУП

Рекомендаційні системи з’явилися на сучасному ринку IT як механізм для заміни статичному списку рекомендацій при пошуку або покупках на веб-сайтах. Рекомендаційна система — підклас системи фільтрації інформації, яка будує рейтинговий перелік об’єктів (фільми, музика, книги, новини, веб-сайти), яким користувач може надати перевагу. Для цього використовується інформація з профілю користувача.

В наш час будь-які рекомендації на основі коментарів користувачів є практичною задачею. Раніше всі відомості щодо предмету рекомендації зберігались на паперах, ще було ризиком через їх велику кількість та можливість втрати. Зі стрімким розвитком електронних програм та інформаційних систем ця задача значно полегшилася. Коментарі для рекомендації стали доступними на форумах, в соціальних мережах, в YouTube та інших додатках.

1 ПРИНЦИПИ ПОБУДОВИ РЕКОМЕНДАЦІНИХ СИСТЕМ

1.1Підходи до побудови рекомендаційних систем

При розробці рекомендаційних систем зазвичай розробники стикаються з рядом проблем прогнозування: розрідженість даних, холодний старт, бульбашки фільтрів, синонімія [1], шахрайство, розмаїття, білі ворони [2], про які більш детально описано в підрозділі 1.3.

Для позбавлення від цих проблем використовуються такі підходи, як колаборативна, контентна та гібридна (поєднання колаборативної та контентної) фільтрації.

1.1.1 Колаборативна фільтрація

Колаборативна фільтрація — це один з методів прогнозу в рекомендаційних системах, який використовує відомі переваги (оцінки) групи користувачів для прогнозування невідомих переваг (оцінок) іншого користувача. Основна ідея колаборативної фільтрації: схожим користувачам зазвичай подаються схожі об’єкти. За допомогою цього алгоритму будується певна таблиця користувачів, які групуються за схожістю, та прогнозуються результати для інших користувачів. Наприклад, маємо декількох користувачів порталу з музикою. Всіх користувачів можна поділити на групи за їх смаками (одним подобається джаз, іншим — рок). За цією інформацію в середині кожної групи можна виділити найпопулярніші хіти, які користувачі слухають більше всього. Отже, кожному учаснику певної групи будуть рекомендовані популярні композиції, які ним не були ще прослухані [3].

Переваги: швидка робота алгоритмів (K - based та ін.) — мала кількість ітерацій; прості в реалізації.

Недоліки: холодний старт; немає, чого рекомендувати новим або нетиповим користувачам (білі ворони); розріджені матриці оцінок (іноді неможливо зробити прогноз); шахрайство [4].

1.1.2 Контентна фільтрація

Фільтрація вмісту базується на моделі об’єкту, оцінки якого будуть прогнозуватися. Для кожного об’єкту буде побудовано математичну модель з використанням конкретних характеристик товару (параметри моделі). Рекомендації будуть базуватися на порівнянні характеристик поточного товару та власне характеристик користувача (інформація, яка міститься в профілі користувача) [3]. Для прикладу, візьмемо сайт з онлайн-музикою. Нехай маємо користувача, який прослухав музичні твори наступних композиторів: Й. С. Бах, Г. Ф. Гендель, А. Вівальді, Д. Скарлатті та Д. Букстехуде. Висувається гіпотеза, що цьому користувачу подобаються барокова музика, тому логічно будуть створені рекомендації певних токат, прелюдій, концертів та сюїт.

При фільтрації вмісту створюються профілі користувачів і об’єктів:

- Профілі користувачів можуть містити демографічну інформацію або відповіді на певний набір питань.
- Профілі об’єктів можуть містити назви жанрів, імена акторів, імена виконавців, тощо. Або якусь іншу інформацію в залежності від типу об’єкта.

Цей підхід застосований у проекті Music Genome Project: музичний аналітик оцінює кожну композицію за сотнями різних музичних характеристик, які можна використати для виявлення музичних уподобань користувача [5].

Переваги: більш точний результат; немає проблеми холодного старту, оскільки рекомендації базуються на моделі об’єкта, а не на попередніх оцінках користувачів.

Недоліки: «затратне» створення моделі (її побудова досить складна), невисока швидкодія алгоритмів (багато обчислень); втрата точності при скороченні параметрів моделі.

1.1.3 Гібридна фільтрація

Даний тип алгоритмів поєднує в собі підходи колаборативної та content-based фільтрації. Цей підхід найбільш популярний при розробці рекомендаційних систем для комерційних сайтів, так як його було створено щоб

подолати проблеми колаборативної фільтрації, а також покращити якість прогнозування оцінок конкретної моделі.

Переваги: велика швидкодія; кращі результати.

Недоліки: дуже дорога розробка рекомендаційної системи, оскільки реалізація цього типу алгоритмів дуже складна; важко підтримувати, оскільки навіть незначні зміни в роботі призводять до змін роботи ал- горитму.

Одним з нових алгоритмів, який майже не має звичайних проблем для заснованих на сусідстві підходів, виявився гібридний SVD алгоритм (метод сингулярного розкладу), який було створено саме для покращення результатів звичайних алгоритмів [3].

1.2Методика створення рекомендаційних систем

У процесі роботи рекомендаційні системи збирають дані про користувачів. Для опису здобуття знань у базах даних, дослідження даних, обробки зразків даних, очищення і збору даних застосовується термін «*інтелектуальний аналіз даних»* (ІАД, Data Mining).

Data Mining – це процес виявлення у необроблених даних раніше невідомих нетривіальних, практично корисних і доступних інтерпретацій знань, необхідних для прийняття рішень у різних сферах діяльності [6]. Добування даних також глибинний аналіз даних – процес напівавтоматичного аналізу великих баз даних з метою пошуку корисних фактів. Метою аналізу є виявлення правил та закономірностей, наприклад, статистичних подій [7]. В основу технології Data Mining покладено концепцію шаблонів (patterns), що є закономірностями, які властиві вибіркам даних і можуть бути подані у формі, зрозумілій людині. До основних задач Data Mining відносяться: класифікація (Classification), кластеризація (Clustering), асоціація (Associations), послідовність (Sequence) або послідовна асоціація (sequential association), прогнозування (Forecasting), визначення відхилень (Deviation Detection), оцінювання (Estimation), аналіз зв'язків (Link Analysis), візуалізація (Visualization, Graph Mining) та підбивання підсумків (Summarization) [6].

Здобуття знань для створення певних рекомендацій здійснюється за допомогою явних та неявних методів збору даних.

Явний збір даних

При явному зборі даних користувач добровільно надає системі необхідні для роботи дані.

Приклади явного збору даних:

- користувач оцінює запропонований об’єкт за диференційованою шкалою;
- користувач ранжує групу об’єктів від найкращого до найгіршого;
- користувач обирає кращий з двох запропонованих об’єктів;
- користувачу пропонують створити список його улюблених об’єктів.

Неявний збір даних

Методика неявного збору даних являє собою своєрідне «шпигунство» за користувачем.

Приклади неявного збору даних:

- спостереження за тим, що користувач оглядає в інтернет-магазинах або базах даних іншого типу;
- ведення записів про поведінку користувача онлайн;
- відстеження вмісту комп’ютера користувача [5].

1.3 Проблеми рекомендаційних систем

Сучасні рекомендаційні системи мають ряд стандартних проблем та недоліків, дослідження яких та розробка методів їх подолання є актуальною науково-практичною задачею.

1. Проблема холодного старту (Cold-start Problem, CSP) виникає тоді, коли в системі з’являються нові елементи — або нові користувачі (User Cold-Start), історія вподобань яких порожня, або нові об’єкти (Item Cold-Start), у яких ще немає оцінок та/або набору ознак. У багатьох реальних системах CSP може набувати характеру циклічної проблеми для вже відомих користувачів або об’єктів. Наприклад, якщо частина користувачів змінює свої інтереси. Дана проблема отримала назву проблеми постійного холодного старту (Continuous

Cold - start Problem, CoCoS). Як і CSP, проблема CoCoS може виникати з користувачами (User Continuous Cold - start Problem) та з об'єктами (Item Continuous Cold - start Problem). User Continuous Cold - start Problem виникає для користувачів, що змінюють свої вподобання, або рідко з'являються у системі та рідко оцінюють нові об'єкти. Item Continuous Cold-start Problem виникає при наявності об'єктів, властивості яких можуть змінитися з часом. Для вирішення проблеми Cold - start Problem, як правило, застосовують наступні підходи:

- гібридизація РС з поєднанням контентної та колаборативної фільтрації.
- використання контексту, в якому створюються та надаються рекомендації (демографічні дані, час та дата, тощо) [8].

2. Проблема бульбашки фільтрів. Класичні РС пропонують користувачам об'єкти, виходячи лише з їх попередніх вподобань. Отже, користувач потрапляє у інформаційне середовище, в якому спостерігає лише обмежену кількість однотипних об'єктів. Наслідки, викликані бульбашкою фільтрів:

1) Користувач не одержує альтернативну інформацію, яка може бути йому корисною (напр., види об'єктів, про які він зовсім не знає, але які ефективніше вирішать задачі його пошуку).

2) У користувача формується викривлена точка зору на інформаційне середовище, так як він не бачить картини в цілому (напр., при рекомендації новин).

3) Користувач може втратити інтерес до списку рекомендацій, так як йому весь час пропонують однотипні об'єкти (напр., втратить інтерес до прослуховування онлайн радіо з однотипним набором пісень).

Оскільки усі РС як основну метрику якості своєї роботи використовують точність прогнозування вподобань користувачів, а формальна постановка задачі прогнозу оцінок виглядає наступним чином: $d(R,V) \rightarrow \min$, (1) де $R = (r_1, r_2, \dots, r_n)$ – вектор, що містить список прогнозованих оцінок користувача, впорядкований за спаданням величини оцінок, $V = (v_1, v_2, \dots, v_n)$ – вектор, що містить справжні оцінки користувача, невідомі системі на етапі формування

списку рекомендацій, то для всіх РС проблема бульбашки фільтрів є актуальною, так як якісна рекомендаційна система повинна створювати рекомендації максимально схожі на попередні вподобання користувача.

Інформаційні бульбашки мають різні причини та походження, адже ми можемо свідомо чи несвідомо уникати певної інформації, використовувати або одні алгоритми соціальних мереж чи водночас отримувати інформацію через інші канали. Від цього залежить, наскільки наша бульбашка буде щільною. Таким чином ми випускаємо з поля зору інформацію, яка суперечить нашим вподобанням. Та не отримуємо інформацію, що потенційно може розширити наш світогляд, зробити нас більш толерантними до інших точок зору чи навіть спонукати змінити думку про щось, вказати на іншу логіку подій чи процесів.

В результаті виникає враження, що ваша точка зору панує всюди. Начебто вся стрічка новин у вашому особистому профілі у фейсбук чи іншій соціальній мережі — це віддзеркалення життя та вподобань решти країни.

Деякі науковці вважають, що наразі недостатньо даних, щоб вважати, що вони ґрунтовно впливають на вибір читачів стосовно контенту. Адже багато людей отримують інформацію з інших каналів також — телебачення, друкованих видань чи радіо. Навіть якщо вони користуються онлайн ресурсами, важко сказати, чи знаходяться вони в бульбашці інформації.

Для того, щоб уникнути бульбашки фільтрів, потрібно:

—підписатися на кілька сторінок платформ, медіа чи особисті профілі, що суперечать вашим поглядам. До цього можна додати вмикання телевізійних новин час від часу. Інші точки зору, протилежні думки, з якими ми не завжди погоджуємося, важливі, щоб про них знати.

—використовувати режим анонімного перегляду, видаляти історію пошуку, чистити кеш та видаляти або блокувати файли cookie. Це потрібно робити, щоб інформація про вас та ваші дії не зберігалася в браузері.

—перевірити налаштування акаунтів в соціальних мережах. Так само ваша інформація може збиратися і на її основі «будуватися» результати пошуку в браузерах [9].

3. Проблема розріджених даних. Розрідженість даних зазвичай виникає при оцінці користувачами обмеженої кількості доступних елементів, особливо коли каталог великий. Результатом є розріджена матриця рейтингу користувачів з недостатніми даними для ідентифікації подібних користувачів або товару, що негативно впливає на якість рекомендацій. Розрідженість даних переважає в РС, які покладаються на зворотний зв'язок з колегами для надання рекомендацій. Розрідженість даних міждоменних рекомендацій часто вирішують використанням моделі факторизації тріадового відношення user-item-domain. Також розглядають кожний рейтинг користувацького елемента як предикат інших відсутніх оцінок. Проблема великої розрідженості полягає в тому, що більшість комерційних рекомендаційних систем містить велику кількості даних (предметів), в той час як більшість користувачів не ставить оцінки всім предметам. В результаті цього матриця «предмет-користувач» виходить дуже великою і розрідженою (містить більшість 0 серед елементів), що представляє проблему при обчисленні рекомендацій.

Точність рекомендацій є здатністю РС релевантно передбачити переваги елемента для певного користувача. Підвищенню точності рекомендацій завджи приділяється велика увага. Це особливо необхідно в ситуаціях розрідженості даних. *Масштабованість* є важкодоступною характеристикою, яка пов'язана з кількістю користувачів і товару РС. При формуванні рекомендацій кількох пунктів для декількох сотень користувачів, ймовірно, система не зможе запропонувати мільйонам людей сотні товару, якщо не розрахована на високу масштабованість. *Різноманітність* є бажаною характеристикою, яка останнім часом привертає особливу увагу. Важливо мати різноманітні рекомендації, оскільки це допомагає уникнути упередженості рекомендацій. Таким може бути список рекомендацій з подібними елементами. Користувач, не зацікавлений в одному з них, ймовірно, не зацікавлений у жодному, не отримує користі з рекомендації.

4. Шахрайство. У рекомендаційних системах, де кожен може ставити оцінки, люди можуть давати позитивні оцінки своїм предметам і погані своїм

конкурентам. Також, рекомендаційні системи стали сильно впливати на продажі та прибуток, з тих пір як отримали широке застосування в комерційних сайтах. Це призводить до того, що недобросовісні постачальники намагаються шахрайським чином піднімати рейтинг своїх продуктів і знижувати рейтинг свої конкурентів.

5. Проблема синонімії полягає в тенденції схожих і однакових предметів мати різні імена. Більшість рекомендаційних систем не здатні виявити ці приховані зв'язки і тому відносяться до цих предметів як до різних. Наприклад, «музика для дітей» та «дитяча музика» відносяться до одного жанру, але система сприймає їх як різні.

6. Проблема «Білих ворон». До «білих ворон» відносяться користувачі, чия думка постійно не збігається з більшістю інших. Через унікальність смаку їм неможливо щось рекомендувати. Однак, такі люди мають проблеми з отриманням рекомендацій і в реальному житті, тому пошуки вирішення даної проблеми в даний час не ведуться.

Іншими проблемами є відсутність персоналізації, збереження конфіденційності, зменшення шуму, інтеграція джерел даних, відсутність новизни та адаптивність переваг користувача [10].

1.4 Огляд існуючих рекомендаційних систем музичних композицій

Існує декілька систем за рекомендацією музики: наприклад, last.fm, Pandora, spotify, Apple Music, Google Play Music. Маючи неповний список вподобань користувача, їх рекомендаційні системи передбачають, яка музика йому сподобається.

1) Last.fm — музична рекомендаційна система, яка для надання рекомендацій використовує комбінацію внутрішніх даних, таких як рейтинги користувачів, з зовнішніми — дані про пісню, дата виходу, ім'я автора, жанр та ін.

На основі аналізу статистики прослуховувань користувачам сайту індивідуально кожному підбираються й демонструються: рекомендовані сайтом для прослуховування музичні треки, популярні у схожих за смаками слухачів

(ступінь «подібності» при підборі можна регулювати); персональні сторінки людей із близькими смаками; анонси найближчих концертів; тощо.

На сайті діє система тегів. Кожен альбом, пісня або виконавець можуть позначатися користувачами по їхньому смаку тим або іншим довільним найменуванням. Сайт робить і зворотну процедуру, наприклад можливе прослуховування музики з бази сайту (за допомогою безкоштовної програми клієнта або вбудованого безпосередньо на сторінки флеш-плеєра) раніше вже позначеної користувачами яким-небудь тегом (або декількома тегами).

На сайті last.fm існують співтовариства користувачів, об'єднаних за довільною ознакою. Географічного, смаковому, об'єднуючого користувачів яких або сторонніх сайтів і т. д. Для співтовариств розраховуються сумарні хіти-паради, існують форуми, механізми вибору «лідера» (користувача наділеного розширеними модераторськими повноваженнями) [11].

Логотип рекомендаційної системи last.fm представлено на рис. 1.1.

Рисунок 1.1 — Логотип last.fm

2) Pandora — одна з найпопулярніших музичних рекомендаційних систем. Pandora надає рекомендації на основі змісту музичної композиції, використовуючи для цього професійних музикантів, які аналізують композицію по декількох сотнях атрибутів. Система базує свої рекомендації на даних, отриманих з проекту Music Genome. Він приписує до 400 атрибутів кожної пісні. «Геном» Music Genome Project був доповнен і традиційною рекомендаційною надстройкою, що заснована на «лайках»: кожну композицію, що прослуховується можна оцінити як «таку, що сподобалась» або «таку, що не сподобалась». Це змінює вагу певних параметрів в обчисленнях та дозволяє системі швидше пристосуватися саме до ваших персональних смаків, які зовсім

не обов′язково повинні повністю та безумовно підпорядковуються математичним моделям [12].

Логотип рекомендаційної системи Pandora представлено на рис. 1.2.

Рисунок 1.2 — Логотип PandoSpotify

3) Spotify — інтернет-сервіс потокового аудіо (стримінговий сервіс), що дозволяє легально й безкоштовно прослуховувати музичні композиції. Надає послуги легального онлайнового стримінгу аудіозаписів основних світових і незалежних лейблів, включаючи BBC, Sony, EMI, Warner Music Group та Universal. Запущений у жовтні 2008 року шведським стартапом «*Spotify AB*». Одна з найвизначніших особливостей Spotify — це плейлист Discover Weekly, який щопонеділка пропонує 30 композицій, які можуть сподобатися користувачу.

Щоб створити Discover Weekly, Spotify застосовує три головні типи моделей:

1. Моделі колаборативної фільтрації (їх застосовує Last.fm), які аналізують вашу поведінки та поведінки інших користувачів.

2. Моделі обробки природної мови (NLP), які працюють на основі аналізу текста.

3. Аудіо-моделі, які аналізують необроблені аудіо [13].

Логотип рекомендаційної системи Spotify представлено на рис. 1.3.

Рисунок 1.3 — Логотип Spotify

4) Apple Music

Розділ «Для вас» оновлюється протягом всього дня, пропонуючи вам нові музичні композиції, грунтуючись на жанрах та виконувачх, яким ви віддаєте перевагу, а також темах настрою. Крім того, була змінена компонування деяких елементів інтерфейсу. Наприклад, тепер розділи з нещодавно прослуханими треками та треками друзів помінялись місцями, а під ними з'явилася карусель з плейлистами, сформованими на основі ваших интересів. Apple Music при формуванні рекомендацій буде відштовхуватися від конкретних плейлистів, радячи на їх основі окремі треки. Така модель формування підбірок повинна стати більш ефективною, пропонуючи користувачу максимально релевантні треки, які, напевно, йому сподобаються. Логотип рекомендаційної системи Apple Music представлено на рис. 1.4.

Рисунок 1.4 — Логотип Apple Music

5) Google Play Music

Google Play Музика — це сервіс потокового транслювання музики і підкастів та онлайн-сховище для музики, яким керує Google. Більш бюджетний аналог Apple Music, схожий своїми підбірками на Spotify. Є гарна функція представлення, яка запитує вас про улюблені жанри та виконавців, які

допоможуть налаштувати рекомендації Google Play для списків відтворення, альбомів та радіостанцій [15]. Логотип рекомендаційної системи Google Play Music представлено на рис. 1.5.

Рисунок 1.5 — Логотип Google Play Music

6) YouTube Music

YouTube Music — музичний продукт на базі YouTube. Доступний в мобільному додатку (Android, iOS) та за адресою music.youtube.com у двох версіях — платній та безкоштовній.

Рекомендації сервису будуються на базі поточних переваг користувача в YouTube, місцезнаходження та часу дня. Вони оновлюються щохвилини.

Історія переглядів та лайки YouTube Music та самого YouTube синхронізуються, тому рекомендації в музичному додатку повинні бути корисними. Якщо увійти в аккаунт Google, при складанні рекомендацій будть враховуватися також музичні відео, які користувач дивиться в YouTube. Логотип рекомендаційної системи YouTube Music представлено на рис. 1.6.

Рисунок 1.6 — Логотип YouTube Music

Таким чином, рекомендаційні системи у музиці користуються великим попитом та використовуються людьми у всьому світі.

2 РОЗРОБКА РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ ВИКОНАВЦІВ МУЗИЧНИХ КОМПОЗИЦІЙ

2.1 Набір даних для рекомендаційної системи

Набір даних для рекомендацій зазвичай відбувається за допомогою корпусу даних, в даному випадку — корпусу текстів (коментарів).

Корпус текстів — це вид корпусу даних, одиницями якого є тексти або їх достатньо значні фрагменти, що включають, наприклад, якісь повні фрагменти макроструктури текстів даної проблемної області. Корпус текстів характеризується чотирма основними параметрами: по-перше, він повинен бути достатньо великого обсягу; по-друге, корпус повинен бути структурованим або розміченим; по-третє, тексти, складові певного корпусу, повинні бути в електронному варіанті; по-четверте, в поняття «Електронний корпус» входить, як правило, спеціальне програмне забезпечення для роботи з цим корпусом.

Сучасні комп'ютерні програми дозволяють знаходити потрібні приклади з корпусів текстів, які зберігаються в електронному вигляді на комп'ютері. Це економить значну кількість часу в порівнянні з традиційною технологією збору прикладів вручну [16].

Спосіб представлення й зберігання корпусу даних (КД).

Найбільший інтерес мають ті способи, які спираються на сучасні комп’ютерні технології зберігання і обробці даних. Для подальшого викладання важливо підкреслити різницю між двома головними способами репрезентації — неструктурованим текстовим форматом зберігання (запис графем текста в ASCI-кодах) та структурованим форматом зберігання (текст із спеціальной розміткою). До останнього можна віднести також представлення даних у форматах БД різного типу.

Вимоги до корпусу текстів з точки зору користувача.

Корпус даних, якій є відображенням області реалізації мовної системи, повинен поєднувати найбільш суперечливі вимоги. Через те що послідовне дотримування будь - якої з вимог призводить до руйнування корпусу, необхідно

дотримуватись балансу між ними. Стратегія побудови корпусу формується як раз із того, як поєднати різноманітні вимоги.

Репрезентативність є найважливішою властивістю корпусу текстів по відношенню до області реалізації мовної системи. Під репрезентативністю ми розуміємо властивість корпусу текстів відбивати всі властивості області реалізації мовної системи, релевантні до даного типу лінгвістичного дослідження, в певної пропорції, яка визначається частотой явища в області реалізації мовної системи. Іншими словами, частота явища в корпусі повинна бути близька частоті цього ж явища в області реалізації мовної системи. Така вимога орієнтує того, хто збирає корпус текстів на спеціалізацію продукту якій розробляється за рівневою тематикою: фонетичні, морфологічні, синтаксичні, лексичні, текстові та ін. корпуси.

Повнота.

Якщо репрезентативність вказує на пропорційне відображення області реалізації мовної системи в корпусі даних, то за певними умовами деякі релевантні явища зникають при підвищенні порогу відображення. Саме повнота вимагає врахування релевантних явищ, навіть коли це не відповідає ідеє пропорційного звуження. Вимога повноти виникає в тих випадках, коли конструктор корпусу приблизно розуміє, що йому треба шукати. В такій сітуації дослідницький корпус більш схожий на ілюстративний.

Економічність.

Корпус текстів не повинен чітко відбивати особливості області реалізації мовної системи і, таким чином, бути підмножиною текстів області реалізації мовної системи. Він повинен суттєво відрізнятися від останньої за об’ємом. В загальному випадку, чим економічнійший корпус, тим вище поріг відображення. В той же час для дослідницьких корпусів економія не повинна бути реалізована за рахунок репрезентативності: статистичні пропорції повинні біти адекватно враховані.

Структуризація матеріалу.

Конструюванню корпусу передує зіставлення опису даних із корпусом. Опис даних містить одиниці зберігання, які можуть бути важливими користувачеві. Состав одиниць зберігання не повинен містить неоднозначності будь-яких типів, наприклад, займенників, для котрих неможливо відновити антецедент тощо.

Комп'ютерна підтримка.

В конструюванні корпусів необхідне досить повне ПЗ комп'ютерної підтримки. Це, перш за все, програми з обробки даних, які забезпечують функції формування конкордансів, статистичної інвентарізації, автоматичної словникарської обробки, лематизації.

Набір даних для рекомендацій певних виконавців буде здійснюватися за допомогою корпусу коментарів, які буде знайдено у відеохостингу YouTube під кожним відео з виконанням та записано в окремі файли [17].

2.2 Обрання методів та підходів до вирішення задачі

2.2.1 Метод найменших квадратів

Метод найменших квадратів — метод знаходження наближеного розв'язку надлишково-визначеної системи. Часто застосовується в регресійному аналізі. На практиці найчастіше використовується лінійний метод найменших квадратів, що використовується у випадку системи лінійних рівнянь. Зокрема важливим застосуванням у цьому випадку є оцінка параметрів у лінійній регресії, що широко застосовується у математичній статистиці і економетриці.

Нехай ми маємо вибірку початкових даних $f(x_i)=y_i$ $i=\overline{1 \dots n}$. Функція f — невідома.

Якщо ми знаємо приблизний вигляд функції f(x), то задамо її у вигляді функціоналу $F(x_i, a_0,\dots,a_m)\approx y_i$, де $a_0,\dots,a_m$ — невідомі константи.

Потрібно мінімізувати відмінності між F та f. Для цього беруть за міру суму квадратів різниць значень цих функцій у всіх точках x_i і її мінімізують (звідки й назва методу):

$$I(a_0,\dots,a_m)=\Sigma^n_{i=0}(y_i - F(x_i, a_0,\dots,a_m))^2\rightarrow\min$$

Коефіцієнти a_j в яких така міра мінімальна знаходять з системи [18]:

$$\begin{cases} \dfrac{\partial I(a_0, \dots, a_m)}{\partial a_0} = 0 \\ \dots \\ \dfrac{\partial I(a_0, \dots, a_m)}{\partial a_m} = 0 \end{cases}$$

2.2.2 Перевірка якості рекомендацій

Для оцінки релевантності потрібно мати: еталонну колекцію документів, яка буде оцінюватися; еталонний набір запитів; оціночні судження.

В результаті можна порахувати recall (повноту), precision (точность) та f-measure. В задачах розпізнавання образів, інформаційного пошуку та бінарної класифікації, точність – частка правильно спрогнозованих екземплярів серед усіх знайдених, а повнота – частка правильно спрогнозованих екземплярів відносно загальної кількості релевантних. Отже, як точність, так і повнота, ґрунтуються на розумінні і мірі релевантності.

В статистиці, якщо нульова гіпотеза полягає в тому, що всі елементи є *несуттєвими* (коли гіпотеза приймається або відкидається на підставі кількості відібраних у порівнянні з розміром вибірки), відсутність помилок першого і другого роду відповідає максимальній точності (немає помилкових спрацьовувань) і максимальній повноті (немає помилково не відібраних). У наведеному вище прикладі розпізнавання буде 8 – 5 = 3 помилки I типу і 12 – 5 = 7 помилок II типу. Точність може розглядатися як показник точності або *якості*, а повнота є мірою *кількості*.

В простих термінах, висока точність означає, що алгоритм повертає більше релевантних результатів, ніж несуттєвих, в той час як висока повнота, означає, що алгоритм повертає більшу кількість релевантних результатів.

В області інформаційного пошуку, точність — це частка знайдених документів, які мають відповідні до запиту. Тобто, для текстового пошуку у множині документів, точність — число правильних результатів розділене на кількість всіх повернених результатів. Точність використовується разом з

повнотою — відсотком *всіх* релевантних документів, які отримані в результаті пошуку. Ці дві міри іноді використовуються разом у F1-оцінці (або F-мірі), щоб отримати тільки одну оцінку якості роботи системи.

Повнота – це частка релевантних документів, які успішно знайдені системою пошуку відносно загальної кількості релевантних документів. У бінарній класифікації, повнота, називається чутливістю. Її можна розглядати як ймовірність того, що відповідний документ отримано з допомогою запиту [19].

Після виконання рекомендацій було отримано 4 види документів, як представлено на рис. 2.2.

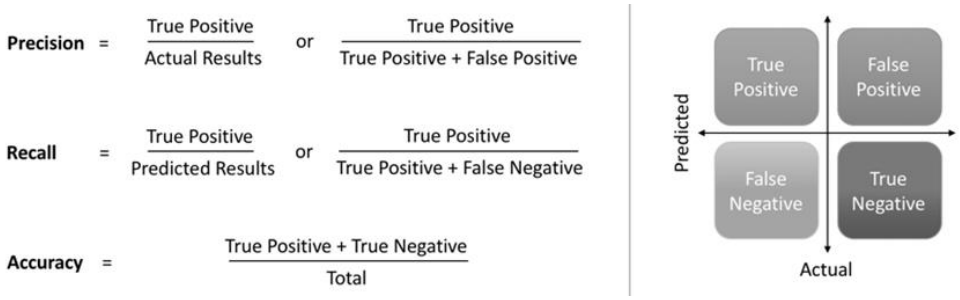

Рисунок 2.2 — Види документів в результаті рекомендацій

Всю колекцію можна розбити на релевантні (true), нерелевантні (negative), видані (positive) та не видані (negative).

Precision вимірюється по горизонталі (відношення правильно виданих до загальної кількості виданих)

$$Precision = tp/tp+fp.$$

Recall вимірюється по вертикалі (відношення правильно виданих до загальної кількості правильных в колекції)

$$Recall=tp/tp+fn.$$

F-measure — середнє між повнотою та точністю:

$$F = 2PR/(P+R).$$

Отже, такі методи оцінки використовуються як для інформаційного пошуку, класифікації, так і для рекомендацій.

2.3 Технології для створення програмного продукту

Задача проекту полягає в наданні рекомендацій щодо виконавців Концерту для фортепіано з оркестром №2 Сергія Рахманінова користувачу згідно з корпусом коментарів у відеохостингу YouTube. В результаті програма повинна видати рейтинг піаністів за найбільшою сумою оцінок за певні коментарі. Для практичної реалізації проекту було обрано мову Python, що сприяє написанню надійного програмного продукту. На даний момент Python є однією з найпопулярніших мов програмування та на ряду з Java всебічно розвивається на ринку програмістів.

Для роботи програми було імпортовано такі бібліотеки та модулі, як os, re, nltk, pymorphy2, sklearn, itertools, collection, webbrowser та tkinter.

Бібліотека ***NLTK*** — це пакет бібліотек та програм для символьної та статистичної обробки природної мови, написанних на мові программування Python. Супроводжується обширною документацією та включає книгу з поясненням основних концепцій, що стоять за тими задачами обробки природної мови, які можна виконувати за допомогою даного пакету. [9]

Пакет *nltk.corpus* визначає колекцію классів *зчитувачів корпусів*, які можна використовувати для доступу до вмісту різних наборів корпусів.

Пакет nltk.corpus автоматично створює набір экземпляров программы чтения корпуса, которые можно использовать для доступа до корпусів в пакете даних NLTK. Даний пакет використано для видалення стоп-слів з текстів.

Модуль *nltk.util* використовується для повернення біграм, згенерованих з послідовності елементів, в якості ітератора.

pymorphy2 — морфологічний аналізатор, написаний на мові Python (працює під 2.7 і 3.3+ версіями мови).

Scikit-learn — це бібліотека машинного навчання з відкритим вихідним кодом, яка підтримує навчання з учителем та без. Вона також надає різні інструменти для підгонки моделі, попередньої обробки даних, вибору і оцінки моделі і багато інших програм.

Модуль *sklearn.feature_extraction* працює з вилученням ознак з вхідних текстів. Підмодуль *sklearn.feature_extraction.text* збирає утиліти для побудови векторів-ознак з текстових документів. [7]

За допомогою класу TfidfVectorizer, було здійснено перетворення колекції необроблених документів у матрицю функцій TF-IDF. У програмі було використано атрибут *idf_* для повернення списку векторів idf та наступні методи:

fit_transform (*self, raw_documents, y=None*) — вивчення словника та idf і повернення матриці термін-документ.

get_feature_names (*self*) — повернення списка імен об’єктів. [11]

Модуль ***itertools*** — це модуль для створення власних ітераторів. Метод *chain* може поєднувати декілька списків в один. [3]

Модуль ***collections*** надає спеціалізовані типи даних, на основі словників, кортежів, множин та списків.

collections.Counter — вид словника, який дозволяє рахувати кількість незмінних об’єктів (в більшості випадків, рядків). [2]

Модуль ***webbrowser*** забезпечує високорівневий інтерфейс, що дозволяє відображати веб-документи для користувачів. В більшості випадків простий виклик функції open() з цього модуля відкриє правильну веб-сторінку. [14]

Tkinter - це пакет для Python, призначений для роботи з бібліотекою Tk. Бібліотека Tk містить компоненти графічного інтерфейсу користувача (graphical user interface - GUI), написані мовою програмування Tcl.

Під графічним інтерфейсом користувача (GUI) маються на увазі всі ті вікна, кнопки, текстові поля для введення, скролери, списки, радиокнопки, прапорці та ін., які користувач може бачити на екрані, відкриваючи ту чи іншу

програму. Через них він взаємодіє з програмою і керує нею. Всі ці елементи інтерфейсу разом називаються віджетами (widgets). [6]

Алгоритм роботи розробленого інтерфейсу представлено у Додатку А. Алгоритм роботи програми представлені на рис. 2.2 та 2.3.

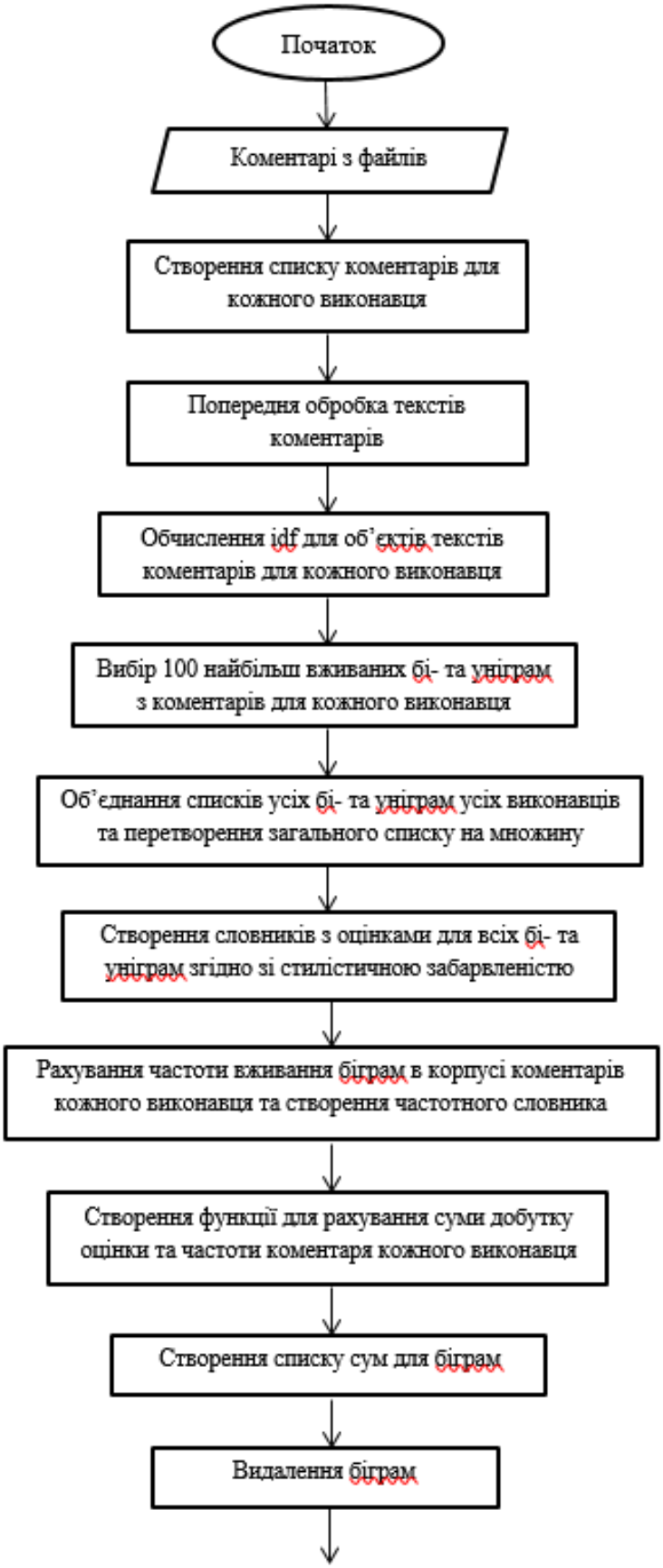

Рисунок 2.2 – Алгоритм роботи програми

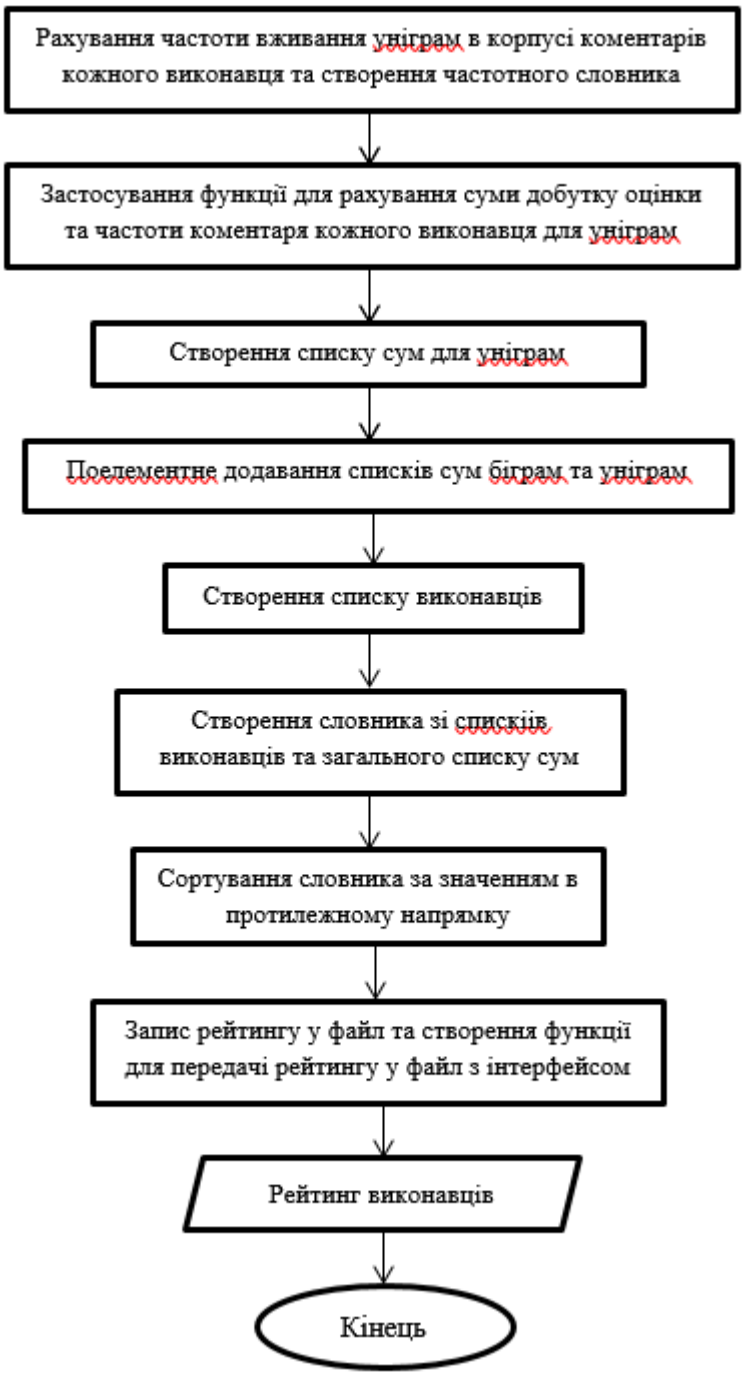

Рисунок 2.3 – Продовження алгоритму роботи програми

Основна проблема при роботі з корпусом була пов’язана з тим, що серед оброблюваних коментарів важко було провести релевантну оцінку на основі негатиной або позитивной лексики. Користувачі, які писали коментарі до творів класичної музики, висловлювали тільки позитивні оцінкі, тобто для визначення ваги того чи іншого виконавця було створено словники позитивних оцінок з відповідними вагами для кожного слова.

2.4 Інструкція для користувача

Програму рекомендацій виконавців було розроблено тільки для Концерту №2 для фортепіно з оркестром С. Рахманінова, тому у випадках вибіру користувачем інших напряму, композитора та жанру, йому буде запропоновано змінити свій вибір та прослухати класику, Рахманінова або концерт. Наведемо інструкцію по використанню даного програмного продукту.

1. Запустити файл interface.py.

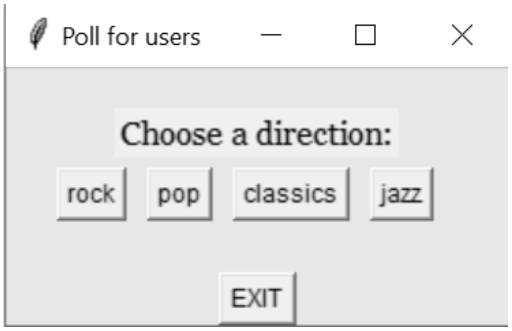

Рисунок 2.4 — Початковий вид анкети для користувача

2. Обрати один з напрямів музики.

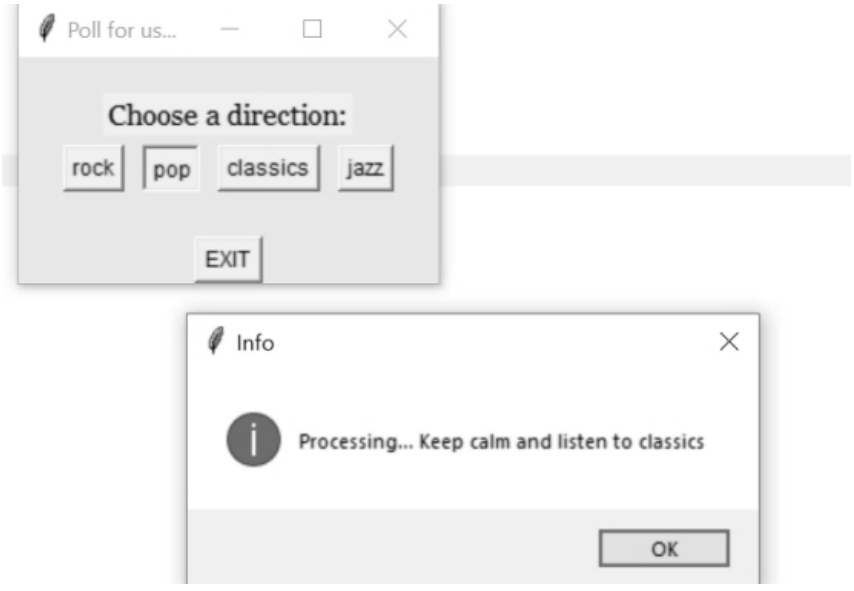

Рисунок 2.5 — Результат вибору НЕ класичного напряму музики

Та отримати вікно з результатом вибору.

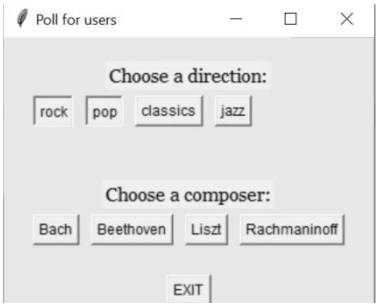

Рисунок 2.6 — Результат вибору класичного напряму музики

3. Обрати композитора.

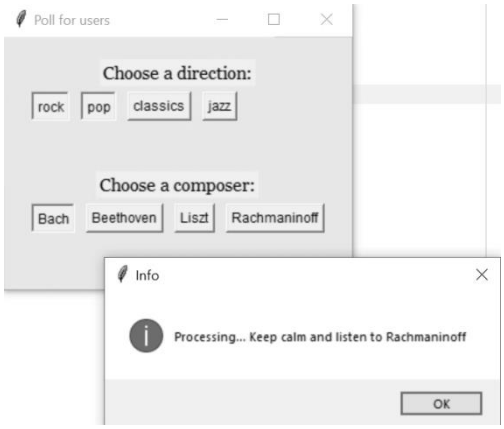

Рисунок 2.7 — Результат вибору НЕ Рахманінова

Або обрати іншого композитора.

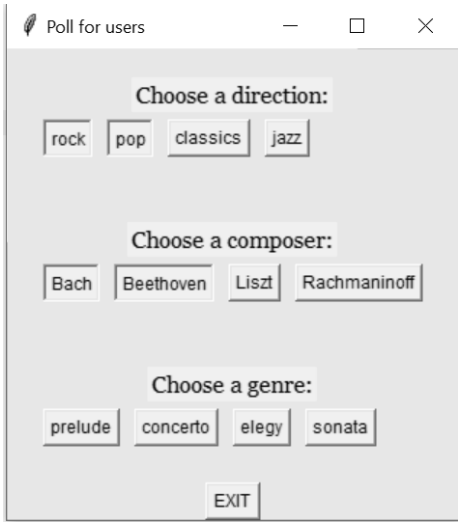

Рисунок 2.8 — Результат вибору Рахманінова

4. Обрати жанр.

Рисунок 2.9 — Результат вибору НЕ концерту

Та отримати посилання на відеороліки в YouTube, причому виконавці будуть відсортовані в порядку зменшення вподобань слухачів.

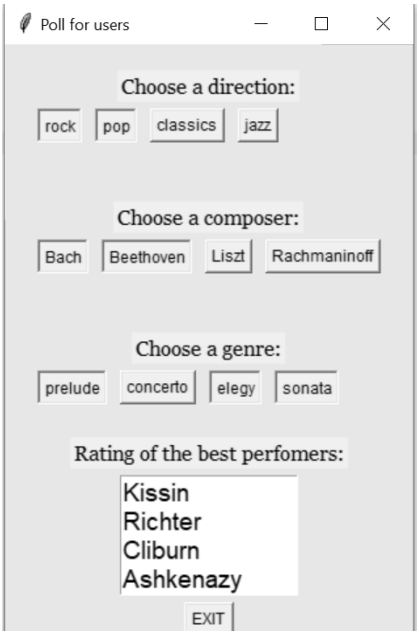

Рисунок 2.10 — Результат вибору концерту

Якщо користувач натискає на кнопку «EXIT», вікно анкети закривається, а робота програми закінчується. Завершити роботу програми користувач може в будь-який час.

ВИСНОВКИ

В ході виконання роботи було розглянуто проблему розробки рекомендаційної системи для виконавців музичних творів та проблеми самих рекомендаційних систем, проведено аналіз існуючих методів та підходів для вирішення цієї задачі.

Було розглянуто існуючі рекомендаційні системи музичних творів та принципи їх роботи. Створено прогамний продукт, який на основі оброблюємого корпусу коментарів до творів класичної музики, створюю список виконавців з урахуванням оцінки коритувачами їхнього виконання. Після обрання жанра, комозитора та концерта, користувач отримує посилання на відеороліки відповідних концертів на YouTube та може насолоджуватися віртуозним виконанням класичних музичних творів.

СПИСОК ДЖЕРЕЛ ІНФОРМАЦІЇ

1. Linden G., Smith B., and York J. (2003). Item-to-Item Collaborative Filtering. Los Alamitos, CA, USA: IEEE Internet Computing. p76–80

2. Sarwar B., Karypis G., Konstan J., and Riedl J. (2001). WWW10 Conference Materials / Item-Based Collaborative Filtering Recommendation Algorithms. Hong Kong: WWW10. p 285–289

3. Рекомендательные системы: Часть 1. Введение в подходы и алгоритмы [Електронний ресурс]. — Режим доступу: https://www.ibm.com/developerworks/ru/library/os-recommender1/

4. Мазурік О. Ю. Покращення результатів роботи рекомендаційних систем за допомогою алгоритму SVD / О. Ю. Мазурік // International scientific journal. – 2015. – № 9. – С. 61-64.

5. Рекомендаційні системи [Електронний ресурс]. — Режим доступу: https://uk.wikipedia.org/wiki/Рекомендаційна_система

6. Інтелектуальні технології Data Mining і Text Mining [Електронний ресурс]. — Режим доступу: https://pidruchniki.com/1623021247786/informatika/intelektualni_tehnologiyi_data_mining_text_mining

7. Добування даних [Електронний ресурс]. — Режим доступу: https://uk.wikipedia.org/wiki/ Добування даних

8. Мелешко Є. В. Проблеми сучасних рекомендаційних систем та методи їх рішення / Є. В. Мелешко // Системи управління, навігації та зв'язку. - 2018. - Вип. 4. - С. 120 - 124

9. Як вирватися з інформаційної бульбашки [Електронний ресурс]. — Режим доступу: https://wz.lviv.ua/blogs/388693-yak-vyrvatysia-z-informatsiinoi-bulbashky

10. Розроблення рекомендаційної системи на основі колаборативної фільтрації та machine learning з врахуванням особистих потреб користувача. В. В. Литвин, В. А. Висоцька, В. В. Шатських, І. В. Когут, О. С. Петрученко, Л. В. Дзюбик, В. В. Бобрівець, В. М. Панасюк, С. І. Саченко, М. П. Комар

11. Last.fm [Електронний ресурс]. — Режим доступу: https://uk.wikipedia.org/wiki/Last.fm

12. Pandora Radio [Електронний ресурс]. — Режим доступу: https://upweek.ru/pandora-radio

13. Как машинное обучение в Spotify находит вашу новую любимую музыку [Електронний ресурс]. — Режим доступу: https://apptractor.ru/info/articles/kak-mashinnoe-obuchenie-v-spotify-nahodit-vashu-novuyu-lyubimuyu-muzyiku.html

14. Как работает рекомендательная система «Яндекс.Музыки» [Електронний ресурс]. — Режим доступу: https://vc.ru/flood/7303-yandex-music

15. Кращі сервіси для прослуховування музики онлайн [Електронний ресурс]. — Режим доступу: https://c.mi.com/forum.php?mod=viewthread&tid=563339&aid=1279227&from=album&page=1

16. Корпусна лінгвістика [Електронний ресурс]. — Режим доступу: https://uk.wikipedia.org/wiki/Корпусна_лінгвістика#Корпуси

17. Терміни та користувальницьки вимоги до лінгвістичного корпусу [Електронний ресурс]. — Режим доступу: https://studopedia.su/10_35830_lektsiya--termini-ta-koristuvalnitski-vimogi-do-lingvistichnogo-korpusu.html

18. Метод найменших квадратів [Електронний ресурс]. — Режим доступу: https://uk.wikipedia.org/wiki/Метод_найменших_квадратів#В_математичному_моделюванні

19. Точність та повнота [Електронний ресурс]. — Режим доступу: https://uk.wikipedia.org/wiki/Точність_та_повнота

ДОДАТОК А – Алгоритм роботи інтерфейсу