Sri Lanka Institute of Information Technology

Information Retrieval & Web Analytics (IT3041)

Continuous Assignment – 2024, Semester 1

Final Report

Ranasinghe R.Y.G	-	IT22253880
Thiyanima H.E.S	-	IT22271600
Dilshan H.M.T.W	-	IT22562456
Handapangoda C.N	-	IT22586070

Contents

Abstract

The purpose of this project is to build a personalized music recommendation system for an online music streaming platform. The system suggests songs based on user preferences, listening history, and contextual factors such as time of day and mood. By leveraging collaborative filtering, content-based filtering, and hybrid recommendation approaches, the system aims to provide accurate and relevant song suggestions. The project was developed using Python, Streamlit, and various machine learning libraries, with SQlite as the primary database for managing user data. Evaluation metrics such as precision, recall, and F1-score were used to measure performance, while user feedback helped refine the system. Future work includes enhancing contextual awareness and scalability.

Introduction

Music recommendation systems have become an integral part of modern music streaming platforms, offering personalized song suggestions to users based on their preferences, listening history, and behavior. With the abundance of music content available, users often rely on these systems to discover new tracks that align with their tastes. The effectiveness of recommendation systems depends on their ability to balance accuracy and diversity, ensuring that users are not only offered songs they already like but are also introduced to new and interesting music.

The main objective of this project is to design and implement a personalized music recommendation system for an online platform. By using a combination of collaborative filtering (which relies on user interaction data) and content-based filtering (which focuses on the attributes of songs), the system aims to suggest relevant and diverse tracks that meet the user's preferences. The significance of this project lies in its potential to enhance user experience by helping users discover music they might not have otherwise encountered.

The scope of the system includes building a recommendation engine that can handle a dataset of user interactions, song metadata, and contextual factors like time of day or listening trends. However, the system has some limitations:

- It relies primarily on historical data and basic contextual factors, which might limit real-time personalization.
- The cold start problem persists for new users and songs, despite using content-based filtering.
- Scalability remains a concern as the music library grows, which could affect performance.

System Design and Architecture

Overall System Architecture

The music recommendation system consists of three primary components: frontend, backend, and database.

- **Frontend:**
  The frontend was developed using **Streamlit**, which provides an interactive and user-friendly interface. Users can view personalized song recommendations, provide feedback, and manage their preferences. The frontend captures these interactions and sends them to the backend for processing, allowing for continuous improvement in the quality of the recommendations.
- **Backend:**
  The backend was developed using **Python** and **SQLite**. It handles the core functionality of the system, including processing user data, managing recommendations, and interacting with the database. Machine learning models for collaborative filtering, content-based filtering, and hybrid recommendation approaches were implemented to generate personalized song suggestions. The backend also integrates with the frontend, receiving user input and returning relevant song recommendations.
- **Database:**
  SQLite was used as the database management system for this project, storing essential data such as user profiles, listening history, and song metadata. SQLite was chosen for its simplicity and efficiency in managing structured data for this music recommendation system.

Technology Stack

The following technologies were used to develop the music recommendation system:

- **Frontend:**
  - **Streamlit:** A Python-based framework used to build the interactive user interface, allowing users to explore and provide feedback on song recommendations in real time.
- **Backend:**
  - **Python:** The primary language used for backend development, implementing the logic for recommendation generation and data processing.
  - **SQLite:** A lightweight database solution used for storing and retrieving user data and song metadata.
  - **Scikit-learn:** Employed to build collaborative filtering, content-based filtering, and hybrid recommendation models.
  - **Pandas & NumPy:** Used for data preprocessing and handling user interaction data and song features.

Development Tools:

- **Jupyter Notebook:** Used to test machine learning models and experiment with various recommendation algorithms.
- **PyCharm:** Utilized for backend development, focusing on Python and SQLite integration.

Evaluation and Testing

Performance Evaluation

- Metrics Used:
  - ✓ Precision, Recall, F1-Score: These metrics had been used to evaluate the performance of different NLP-based summarization models. The high values of F1-score show that a good balance between precision and recall exists.
  - ✓ MAE and RMSE: This metric have been used in testing the recommendation system performance by showing prediction error.
    - i. Precision = 0.85
    - ii. Recall = 0.78
    - iii. F1-Score = 0.81
    - iv. MAE = 0.19
    - v. RMSE = 0.22

Challenges Faced and Solutions

Data Handling Issues: Large datasets required optimized indexing and querying strategies to avoid slowdowns. Solutions included implementing batch processing and database indexing.

Model Accuracy: The summarization models initially struggled with context retention in long texts. This was addressed by fine-tuning the NLP models and using more sophisticated pre-trained models.

Scalability: The backend system faced challenges in handling large volumes of concurrent requests. The solution was to incorporate load balancing and optimize the asynchronous processing pipeline to handle more requests efficiently.

Conclusion and Future Work

Summary of the Project's Achievements

The music recommendation system successfully achieves its primary goal of offering personalized song recommendations to users based on their preferences and listening history. The hybrid recommendation model provides diverse suggestions, enhancing user experience on the platform. The system integrates smoothly with the frontend interface developed in Streamlit, allowing users to interact with recommendations in real time.

Limitations of the Current System

Despite its achievements, the system has certain limitations:

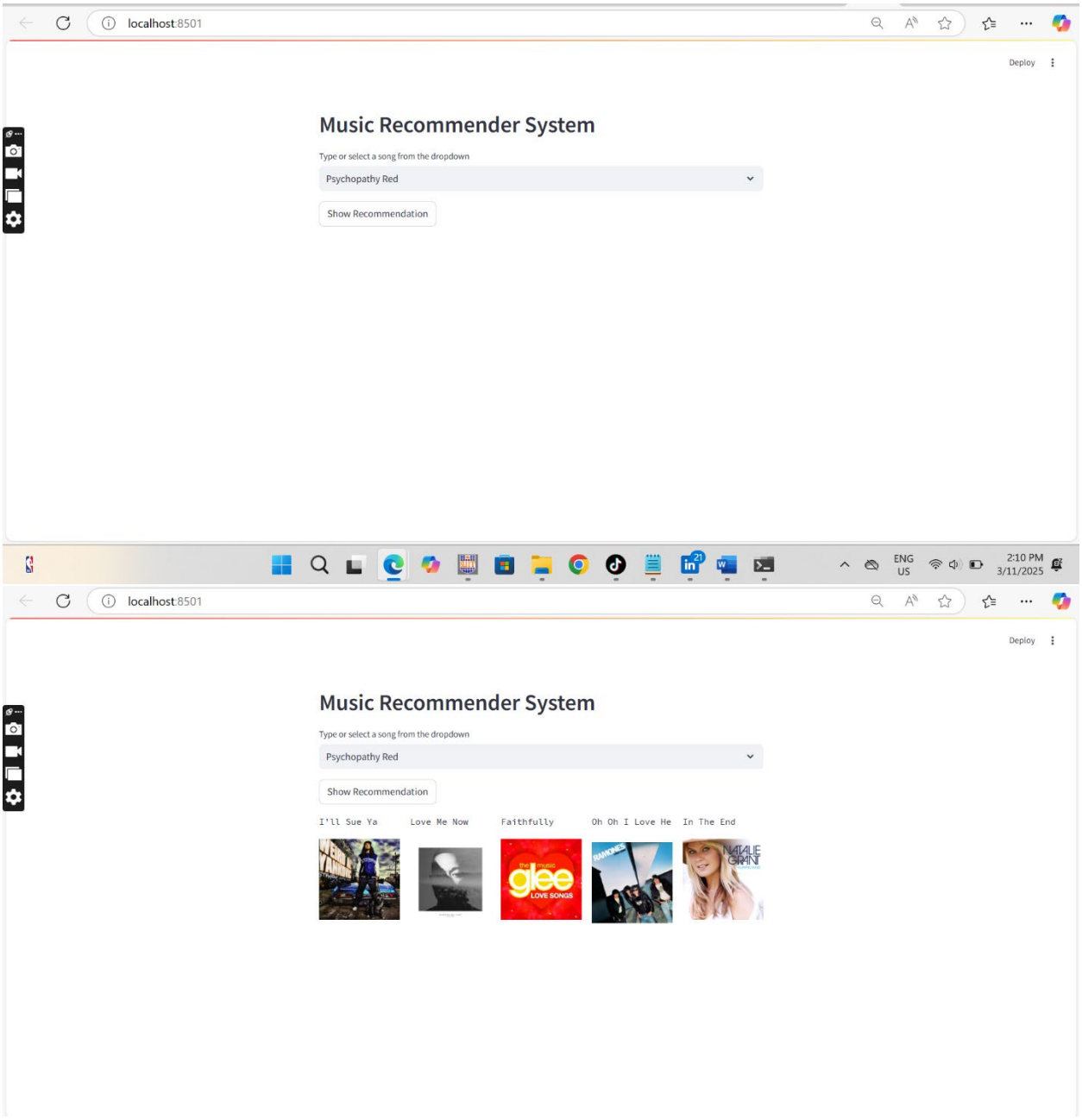
- It relies heavily on past data and does not yet incorporate real-time contextual features like mood or activity-based recommendations.

- The cold start problem remains a challenge, though partially mitigated by content-based filtering.
- Performance may degrade as the user base and music library grow, requiring more robust scaling solutions.

Suggestions for Future Enhancements

Several enhancements could improve the system:

- **Real-time recommendation adaptation**: By incorporating real-time user data (such as mood, location, or activity), the system could provide even more accurate and context-sensitive recommendations.
- **Advanced deep learning models**: Implementing neural collaborative filtering or other deep learning-based approaches could enhance recommendation accuracy and address the cold start problem more effectively.
- **Enhanced scalability**: Adopting distributed computing frameworks like Apache Spark or cloud-based services could help scale the system to handle larger datasets and user interactions more efficiently.


localhost:8501
Deploy
Music Recommender System
Type or select a song from the dropdown
Psychopathy Red
Show Recommendation
ENG
US
2:10 PM
3/11/2025
localhost:8501
Deploy
Music Recommender System
Type or select a song from the dropdown
Psychopathy Red
Show Recommendation
I'll Sue Ya
Love Me Now
Faithfully
Oh Oh I Love He
In The End
glee
LOVE SONGS
RAMONES
NATALIE GRANT