A Hybrid Approach to Personalized Music Recommendation

1st Raja Tejasvi Prasad Panduga
Department of Computer Science
University of North Texas
Denton, Texas, United States
RajaTejasviPrasadPanduga@my.unt.edu

2nd Shiva Kumar Ganji
Department of Computer Science
University of North Texas
Denton, Texas, United States
ShivaKumarGanji@my.unt.edu

***Abstract*—In today's world of online streaming, it has become significant to provide users with personalized music recommendations. The goal of our project is to create a smart music recommendation system that will enhance the user experience along with recommendations on latest songs released.**

The project aims to design and implement a content-based music recommendation system by utilizing different NLP techniques. The system will analyze different features such as song title, artist's name, lyrics and genre to extract meaning insights. Based on the user's input such as genre, artist name, the system will generate closely matched results.

Content-Based Recommendation system considers the song itself rather than the user's data. This will help accurate suggestions of new songs and also recommend the users songs based on different genres. Unlike knowledge-based systems which use collaborative-based filtering that only depend on user data such as user feedback, history and a number of times the user listened to songs on loop and popularity. This approach is time-consuming and expensive to collect such huge data and store them in memory.

I. Problem Statement

Music streaming platforms are offering a wide range of songs composed by artists across the world in various genres. It's a challenge for the user to discover music that aligns with their interest. To address this challenge a recommendation system is built using Natural language processing(NLP) that will suggest music based on user preference, listening history, and taste.

II. Team Collaboration

The team will collaborate closely throughout the project. We will schedule weekly meetings every Monday to discuss progress, and specific milestones of the project and plan the next steps.

III. Data Abstraction

The dataset we are using is a core data which is part of the Million Song Dataset. It involves two sub datasets which are song data that provides details of songs like song id, title, release, artist name, year of release of the album. This information helps us to identify the songs based on song titles and artists.

The other dataset is the count data that provides information of user and the number of times a user listens to a particular song i.e., song id and play count. This dataset specifies the interaction between user and songs. It provides insights into user behavior and user interaction with the songs.

A. Song Dataset

- **songid**: A unique id for every song in the dataset.
- **title**: Defines the song title.
- **release**: The album name.
- **artistname**: Name of the artist.
- **year**: Year of release.

B. Count Dataset

- **userid**: A unique id for each user in the dataset.
- **songid**: A unique id of songs that the user listened to.
- **playcount**: Frequency of user listening to the song.

IV. Project Functions

The music recommendation system uses the multi-model approach where the system is built on three main methods. The following design is used to create the music recommendation system.

- **User-User Collaborative Filtering:** This model recommends songs by analyzing similar user preferences and also leveraging the user-item interaction patterns.
- **Matrix Factorization:** It utilizes the Singular Value Decomposition (SVD) model to predict user preferences based on a matrix representation which is of user-item interactions.
- **Content-Based Filtering:** This is the main model that recommends music based on the content attributes of the songs, such as artist information and song metadata.
- **Feature Extraction Techniques:**
  - For the content-based model, features are extracted from textual data like song titles and artist names.
  - For this we implemented techniques such as TF-IDF and Word2Vec inorder to derive meaningful features from the text.
- **Evaluation Metrics:**
  - The above models used are thoroughly assessed using the metrics like accuracy, RMSE (Root Mean Square Error), recall, and F1-score.

- All these metrics help in evaluating and refining the above models to ensure in creating a reliable music recommendation system for the target users.

V. Functionality and Coding Details

A. Data Management

1) Data Loading and Preprocessing: Data is loaded using the `pandas` library from million song dataset containing user-song interactions and song metadata. The preprocessing steps include merging user and song data, handling all the null or missing values, and also encoding categorical data using LabelEncoder.

B. Model Development

1) Model Training and Evaluation:

- **Collaborative Filtering:** Implemented using libraries like `surprise`, where the system is trained on a KNN (k-Nearest Neighbors) model to find the similar users based on their music listening history and interaction with a particular song.
- **Matrix Factorization:** SVD is then used to decompose the user-item interaction matrix, predicting the overall user preferences.
- **Content-Based Filtering:** This technique uses the methods such as TF-IDF to convert text data into a vector space model, which is then followed by cosine similarity used to recommend songs that have similar text.

C. Recommendation System

1) Recommendation Generation: For each model, the system can generate song recommendations based on the user's history and preferences. Along with the textual data of each song understanding the semantics behind the song title and similar artists.

D. Performance Enhancement

1) Optimization and Tuning: Hyperparameter tuning is performed using the techniques like grid search used to optimize the model parameters in order to improve the accuracy and performance of the recommendations.

VI. Project Design

Technology and Methods:

I. Programming Language: Python

II. Python libraries: Pandas, NLTK, spaCy, Scikit-learn.

III. Google Colab: Environment for coding

IV. Exploratory data analysis which involves data preprocessing techniques like Label Encoding, removing duplicates.

V. NLP techniques: Tokenization, Lemmatization, Ranking Algorithm, Matrix Factorization

VI. Data Visualization libraries: Matplotlib, Seaborn

VII. Model evaluation: Scikit-learn library, RMSE, precision, recall and F1 Score.

The first step involves the collection of the dataset. We perform the Exploratory data analysis on the dataset which involves data preprocessing, and feature extraction.

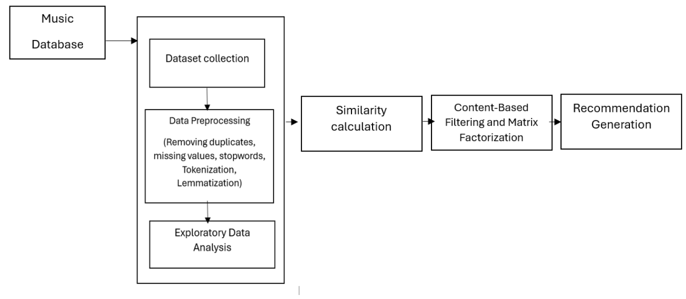

Fig. 1. Work Flow Diagram

Later using different approaches of NLP techniques such as matrix factorization, ranking algorithms, cosine similarity we perform the content based filtering where we consider only the songs as an object.

The recommendation is evaluated using different evaluation techniques such as accuracy, RMSE, precision, recall and F1 score

VII. Project Milestones

The project was completed in the three milestones and the details are listed below:

A. Milestone-1: Data Collection and Preprocessing

- Collect and load the dataset.
- Explore the dataset to understand the features.
- Preprocess the dataset to identify missing values.
- Perform text analysis techniques like tokenization, lemmatization, and frequency analysis.

B. Milestone-2: Content-Based Recommendation System

- Develop a content-based system that recommends similar songs based on artist and genre similarity.
- Utilize techniques such as TF-IDF or Word2Vec for feature extraction.
- Implement matrix factorization techniques like Singular Value Decomposition (SVD) to factorize the user-song interaction matrix.

C. Milestone-3: System Refinement and Evaluation

- Refine the recommendation system based on feedback and performance analysis.
- Evaluate the performance of the system using evaluation metrics such as accuracy, RMSE.

VIII. Graphical Exploration of Dataset Characteristics

Data visualization of the important features of the attributes in the dataset.

The fig-2 screenshot shows the distribution of the number of songs heard by each user.

The fig-3 screenshot shows the top artists whose songs are repeatedly heard by users. As seen Florence songs have been heard most times by the users followed by coldplay.

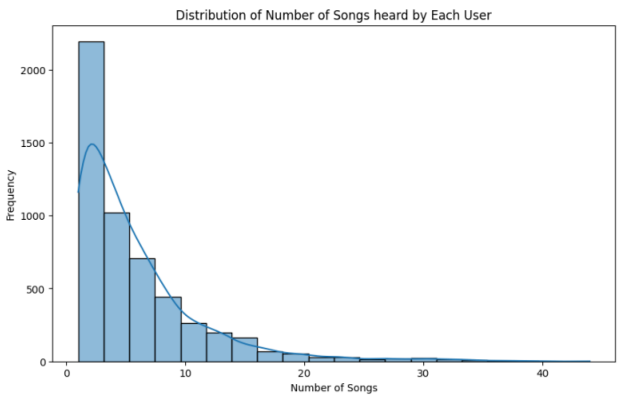

Fig. 2. Distribution of number of songs heard by each user

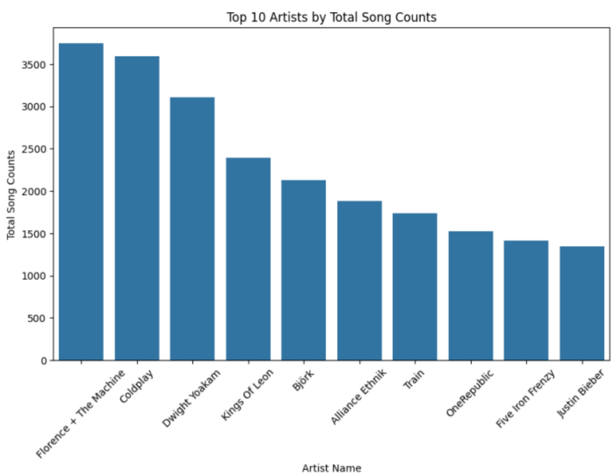

Fig. 3. Top 10 artists by total song counts

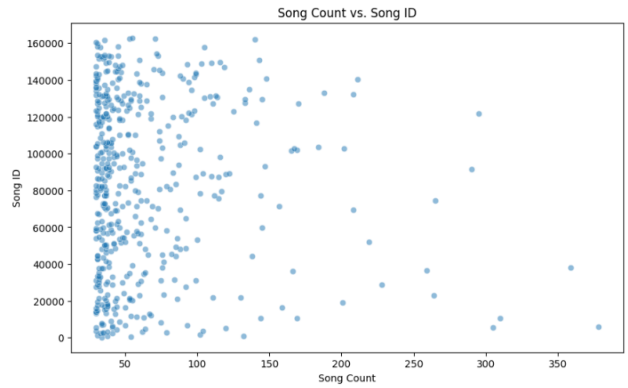

Fig. 4. Scatter plot depicting the each song id and corresponding song count

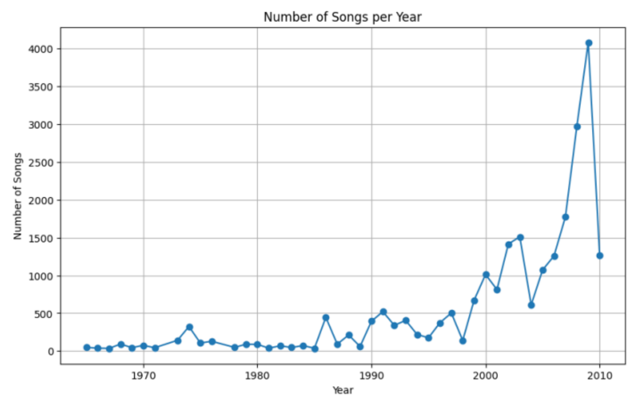

Fig. 5. Scatter plot depicting the each song id and corresponding song coun

The fig-4 screenshot shows the scatter plot of the song count corresponding to the song id.

The fig-5 screenshot shows the line plot of the number of songs released per year. The most number of songs release was in 2010.

IX. Project Results

A. User-User Collaborative Filtering

The User-User Collaborative Filtering model showed significant improvements after performing the hyperparameter tuning:

- The F1 score increased from 0.5299 to 0.6673.
- Recall improved dramatically to 0.9703, indicating that the model is highly effective at identifying relevant items.

B. Matrix Factorization (SVD)

The Matrix Factorization model is created using the SVD which showed substantial gains in precision and F1-score:

- Precision reached up to 0.9633.
- F1 score was elevated to 0.8850, suggesting highly accurate predictions.

C. Content-Based Filtering

Content-Based Filtering was evaluated qualitatively:

- The model successfully provided relevant song recommendations based on the content analysis of song attributes, demonstrating its effectiveness in capturing user tastes and preferences.

X. Results Evaluation

Below are the detail evaluation of the three models: **User-User Collaborative Filtering:**

Before Hyperparameter Tuning:

- RMSE: 7.5403
- Precision: 0.4539
- Recall: 0.6364
- F1 Score: 0.5299

After Hyperparameter Tuning:

- RMSE: 7.2820
- Precision: 0.5085
- Recall: 0.9703
- F1 Score: 0.6673

Matrix Factorization (SVD):

Before Hyperparameter Tuning:

- RMSE: 6.4610

- Precision: 0.4348
- Recall: 0.6689
- F1 Score: 0.5270

After Hyperparameter Tuning:

- RMSE: 1.0882
- Precision: 0.9633
- Recall: 0.8184
- F1 Score: 0.8850

A. User-User Collaborative Filtering

The User-User Collaborative Filtering model shows significant improvements in performance after hyperparameter tuning:

- **Precision:** Precision has been improved after hyperparameter tuning.
- **Recall:** Dramatically increased to 0.9703 indicating high effectiveness in identifying relevant songs.
- **F1-Score:** Increased from 0.5299 to 0.6673 depicting a much better balance between precision and recall.

B. Matrix Factorization (SVD)

The SVD model showed better improvements, especially in the precision:

- **Precision:** Very high at 0.9633 after performing hyperparameter tuning suggesting that most recommendations were relevant.
- **Recall:** Recall is high, indicating good coverage of relevant items.
- **F1-Score:** Enhanced to 0.8850 after tuning, reflecting effective model performance.

C. Content-Based Filtering

This model was evaluated qualitatively based on the relevance of the song recommendations:

- **Evaluation Approach:** Qualitative assessment focused on the contextual relevance of recommendations. It was able to suggest other relevant songs similar to the song that has been given as input.

XI. State of art

The research in the field of music recommendation systems are categorized into knowledge-based system and content-based system. The knowledge-based system involves collecting external data from users like user history and feedback. We can use algorithms such as K-nearest neighbors algorithm and get accurate results but it's again based on the user input or external data which is very expensive and time-consuming.

By the existing methods, the system generates results more dependent on popularity. The system is dependent entirely on the information which is limited. Further this will also lead to memory utilization.

The content-based recommendation system is not completely dependent on user data but on the content of the song. We extract the features from the song itself rather than collecting the user feedback and using memory. We can thus achieve better results with this approach which is not commonly used in the music recommendation systems today.

The main challenge is we can use song features like song title, album name, and artist name but finding a dataset to also analyze the lyrics is difficult.

XII. Conclusion

Each model displayed strengths in various aspects of recommendation quality. The quantitative improvements in User-User Collaborative Filtering and Matrix Factorization models highlight their capability to provide accurate and relevant music suggestions. The qualitative success of the Content-Based Filtering model emphasizes its utility in matching songs with user preferences based on content analysis. The results has shown that the models were effective in predicting the music using the various approaches. The performance of the model has also been improved using hyperparamter tuning. Overall the project has met the performance expectations efficiently.

References

[1] Dataset: http://millionsongdataset.com/tasteprofile/

[2] Deldjoo, Y., Schedl, M., & Knees, P. (2024). Content-driven music recommendation: Evolution, state of the art, and challenges. Computer Science Review, 51, 100618.

[3] Garanayak, M., Nayak, S. K., Sangeetha, K., Choudhury, T., Shitharth, S. (2022). Content and Popularity-Based Music Recommendation System. International Journal of Information System Modeling and Design (IJISMD), 13(7), 1-14.

[4] Aiolli, F. (2013, January). A Preliminary Study on a Recommender System for the Million Songs Dataset Challenge. In IIR (pp. 73-83).

[5] Ramalakshmi, E., Hussain, H., Agarwal, K. (2023). Euphonia: Music Recommendation System Based on Facial Recognition and Emotion Detection. In Intelligent Computing and Networking: Proceedings of IC-ICN 2022 (pp. 85-93). Singapore: Springer Nature Singapore.

[6] Kaggle:https://www.kaggle.com/code/akershishukla/music-recommendation-system