Music Recommendation System Using Million Songs Dataset

Introduction

The project aims to build a music recommendation system using the Million Songs Dataset. Recommendation systems are essential for personalizing user experience by suggesting relevant items. This project demonstrates two types of recommendation systems: popularity-based and item similarity-based, to recommend songs to users.

Data Description

- **Dataset**: The Million Songs Dataset consists of two files:
  - `triplet_file`: Contains user interactions with songs, including `user_id`, `song_id`, and `listen_time`.
  - `metadata_file`: Contains metadata about the songs, including `song_id`, `title`, `release`, `year`, and `artist_name`.
- **Features**:
  - **triplet_file**: `user_id`, `song_id`, `listen_time`
  - **metadata_file**: `song_id`, `title`, `release`, `year`, `artist_name`

Steps and Methodology

1. **Importing Libraries**:
   - Imported necessary libraries such as pandas, numpy, and a custom Recommenders module.
2. **Loading the Dataset**:
   - Loaded the triplet file and metadata file using pandas.
   - Merged both datasets on `song_id` to combine user interactions with song metadata.
3. **Data Preprocessing**:
   - Created a new feature combining the song title and artist name for better identification.
   - Selected the top 10,000 samples for quick processing and results.
   - Grouped the songs by their combined feature and calculated the cumulative listen count and percentage of total listens for each song.
4. **Popularity Recommendation Engine**:
   - Implemented a popularity-based recommendation system using a custom `popularity_recommender_py` class from the Recommenders module.
   - Created the model using the combined dataset.
   - Displayed the top 10 popular songs for specific users.
5. **Item Similarity Recommendation Engine**:
   - Implemented an item similarity-based recommendation system using a custom `item_similarity_recommender_py` class from the Recommenders module.
   - Created the model using the combined dataset.

    - Retrieved the listening history of a user and provided song recommendations based on item similarity.
    - Displayed similar songs based on specified input songs.

Results

1. **Popularity Recommendation Engine**:
    - The popularity-based model recommended the most listened-to songs across all users.
2. **Item Similarity Recommendation Engine**:

- The item similarity-based model recommended songs based on the listening history of the user.

Conclusion

- The popularity-based recommendation system effectively recommends the most popular songs to users.
- The item similarity-based recommendation system provides personalized recommendations based on user listening history and item similarities.
- Both models demonstrated the potential to enhance user experience by providing relevant and personalized song recommendations.

Future Work

- **Model Improvement**: Incorporate additional features such as user demographics, song genres, and timestamps to improve recommendation accuracy.
- **Hybrid Recommendation System**: Develop a hybrid recommendation system combining popularity, item similarity, and user-based collaborative filtering for better performance.
- **Scalability**: Optimize the models for scalability to handle larger datasets and real-time recommendations.
- **User Feedback Loop**: Implement a feedback loop to continuously improve recommendations based on user interactions and feedback.
- **Deployment**: Deploy the recommendation system as a web application or mobile app to provide real-time recommendations to users.