Building a Song Recommendation System with Spotify Million Playlist Dataset

1st Ankita Anand
Data Science and Artificial Intelligence
Indian Institute of Technology Guwahati
Guwahati, India
ankita.anand@iitg.ac.in

***Abstract*—The project focuses on developing an advanced Song Recommendation System using the Spotify Million Playlist Dataset. It employs content-based filtering, analyzing song features and metadata to provide personalized music recommendations. Metadata such as popularity, genres, and release years are used to fine-tune recommendations. Users can input preferences like favorite artists, genres, and mood on a custom website. The project uses Big Data techniques like collaborative filtering, NLP, deep learning, and ensemble methods, and MongoDB for dataset management. It addresses the cold start problem for new users/items and aims for scalability using technologies like Apache Hadoop, Spark, Kafka, Flink, and storage solutions. Moreover, this project aims to create a Spotify clone website, enabling users to fine-tune their recommendations based on a multitude of parameters such as artist, genre, popularity, and more. The project explores various recommendation methodologies and aims to create a cutting-edge, scalable, and highly personalized music recommendation system.**

***Index Terms*—song recommendation, Spotify, content-based filtering, collaborative filtering, Big Data, NLP, deep learning, ensemble methods, MongoDB**

I. Introduction and Background

In the realm of music streaming services, the quest for enhancing user experience and personalization has become increasingly paramount. This project embarks on the ambitious journey of developing a state-of-the-art Song Recommendation System, driven by the unwavering motivation to deliver music suggestions that are not just tailored but deeply resonate with individual user preferences.

At its core, the system aims to revolutionize the way users discover and engage with music by offering highly personalized recommendations. Unlike generic approaches, our system is set to harness the vast reservoir of musical data encapsulated in the Spotify Million Playlist Dataset. This dataset stands as a testament to the diversity and richness of musical content, providing a robust foundation for our recommendation system.

Objective:

The primary objective of this project is to develop a Song Recommendation System that offers highly personalized music suggestions to users based on their preferences. This system will utilize the Spotify Million Playlist Dataset, which contains a vast collection of songs and their associated metadata.

II. Methodology

A. Content-Based Filtering

The core methodology of the recommendation system will be content-based filtering. This approach involves analyzing the features and metadata of each song to calculate its similarity to other songs in the dataset. By assigning similarity scores, we can recommend songs that are most similar to the user's preferences.

B. Metadata Utilization

The recommendation system will incorporate song metadata, including popularity, genres, release year, and other relevant attributes. These metadata features will play a crucial role in fine-tuning recommendations, making them more contextually relevant.

C. User-Defined Parameters

To enhance user experience, a Spotify clone website will be created as part of this project. Users will have the freedom to input various parameters such as their favorite artist, preferred genre, mood, and more. The system will incorporate these user-defined parameters into its recommendation algorithms, allowing users to customize their music suggestions to a high degree.

D. Big Data Algorithms

1) Collaborative Filtering: This algorithm analyzes user behavior and preferences to identify patterns and suggest songs liked by users with similar tastes.

2) Matrix Factorization: By decomposing user-item interaction matrices, matrix factorization techniques like Singular Value Decomposition (SVD) and Alternating Least Squares (ALS) can be applied to recommend songs based on latent factors.

3) Natural Language Processing (NLP): Incorporating NLP techniques, the system can analyze song lyrics, reviews, and descriptions to suggest songs with similar lyrical content.

4) Deep Learning: Utilizing deep neural networks, the project can create recommendation models that learn complex patterns and relationships within the music dataset.

5) Ensemble Methods: Combining multiple recommendation algorithms can provide a more robust and accurate recommendation system.

E. Modeling Maestro: TF-IDF and Cosine Similarity

The overture of this modeling masterpiece commences with the melodic application of TF-IDF (Term Frequency-Inverse Document Frequency) vectorization specifically tailored for the 'Artist genres' column. This column, akin to a musical score, encapsulates the diverse genres associated with artists. TF-IDF, a skilled composer in this musical analogy, bestows weights upon metadata based on their frequency of appearance, transforming the mundane into a rich symphony of weighted nuances. Tfidf Vectorizer, a key player in this musical composition, orchestrates the TF-IDF transformation for the Artist Genres. This vectorization process assigns significance to metadata elements based on their frequency, providing a nuanced understanding of the musical landscape encoded in the dataset.

F. Data Storage

The vast Spotify Million Playlist Dataset will be stored and managed using a MongoDB server. MongoDB is a NoSQL database that can efficiently handle large datasets, making it a suitable choice for this project's requirements.

III. Previous Investigations

A. Spotify Recommendation System

1) Collaborative Filtering: Spotify employs collaborative filtering techniques, analyzing user behavior like song skips, listens, and playlist additions. It identifies users with similar preferences and recommends songs based on what similar users enjoy.

2) Audio Analysis: Utilizes audio analysis to extract song features such as tempo, key, mood, and acoustic attributes. Helps in understanding the musical characteristics of songs and recommending similar ones.

3) Playlist Curation: Features curated playlists created by human editors, personalized for users. These playlists play a significant role in recommendations.

4) Natural Language Processing (NLP): Likely incorporates NLP techniques to analyze song titles, artist descriptions, and user-generated comments to identify thematic and contextual connections between songs.

5) User Feedback: Allows users to provide feedback through likes, dislikes, and playlist additions. This feedback fine-tunes recommendations over time.

B. Netflix Recommendation System (and Other Streaming Services)

1) Collaborative Filtering: Analyzes user viewing history, ratings, and interactions to identify similar user profiles. Recommends content based on what similar users have enjoyed.

2) Content-Based Filtering: Analyzes movie and TV show attributes like genres, directors, actors, and keywords. Recommends content that is thematically or stylistically similar to what a user has previously watched.

3) Personalization: Customizes artwork, trailers, and episode order based on individual preferences to enhance user engagement.

4) A/B Testing: Continuously experiments with different recommendation algorithms and user interfaces through A/B testing to optimize user engagement and retention.

5) Contextual Recommendations: Provides context-based recommendations such as ”Because you watched [Title],” connecting users with content they might find interesting based on their recent viewing history.

IV. Proposed Research Recommendation Methodologies

A. Genre-Based Recommendation

1) Methodology:: Content-Based Filtering

2) Approach::

- Data Preparation: Obtain a genre-labeled track dataset and preprocess genre labels.
- Feature Engineering: Extract audio, metadata, and textual features.
- Similarity Calculation: Use TF-IDF, cosine similarity, or Euclidean distance.
- Recommendation: Recommend tracks based on the user's preferred genres.

B. Artist-Based Recommendation

1) Methodology:: Collaborative Filtering

2) Approach::

- Data Preparation: Collect user interaction data with artists.
- Similarity Calculation: Calculate user similarity based on artist interactions.
- Recommendation: Recommend tracks liked by similar users with artist preferences.

C. Mood-Based Recommendation

1) Methodology:: Context-Aware Recommendation

2) Approach::

- Data Preparation: Extract mood features from textual data and label tracks.
- Context Analysis: Use NLP to analyze user input for mood.
- Recommendation: Suggest tracks matching user-indicated mood.

D. Listening History Analysis

1) Methodology:: Collaborative Filtering

2) Approach::

- Data Preparation: Collect and preprocess user listening history.
- Similarity Calculation: Identify similar users based on listening history.
- Recommendation: Recommend tracks similar users interacted with.

E. Liked Tracks-Based Recommendation

1) Methodology:: Collaborative Filtering

2) Approach::

- Data Preparation: Collect explicit user feedback on liked tracks.
- Recommendation: Suggest tracks similar to liked ones.

F. Disliked Tracks-Based Recommendation

1) Methodology:: Collaborative Filtering with Exclusions

2) Approach::

- Data Preparation: Gather explicit user feedback on disliked tracks.
- Recommendation: Modify the model to exclude disliked tracks.

G. Skipped Tracks Analysis

1) Methodology:: Collaborative Filtering

2) Approach::

- Data Preparation: Collect implicit user feedback on skipped tracks.
- Recommendation: Use implicit feedback to improve future suggestions.

H. Research Objectives

- Design a scalable and personalized music recommendation system.
- Address the cold start problem effectively.
- Utilize Big Data technologies for efficient data processing.

I. Methodologies for Research Objectives

1) Cold Start Problem Mitigation:

- Explore hybrid recommendation techniques to tackle the cold start problem for new users and items.
- Investigate meta-learning approaches to provide initial recommendations for new users based on knowledge from existing users or other domains.

2) Scalable Architecture and Big Data Processing:

- Evaluate distributed processing frameworks like Apache Hadoop and Apache Spark for efficient handling of large-scale music datasets.
- Research real-time data processing architectures using technologies such as Apache Kafka and Apache Flink to provide near real-time recommendations.
- Explore scalable storage solutions like Apache HDFS, Amazon S3, or Google Cloud Storage to handle large volumes of multimedia data.

3) Data Collection and Preprocessing:

- Music Data Sources: Aggregate music data from diverse sources, including streaming platforms, public datasets, and APIs, to create a comprehensive dataset.
- Data Cleaning and Normalization: Develop robust data cleaning and normalization pipelines to ensure consistency and quality of the collected music data.

4) Feature Engineering:

- Audio Features: Extract audio features using libraries like Librosa and analyze their relevance for improving music recommendations.
- Textual Analysis: Utilize NLP techniques for sentiment analysis of lyrics and other textual data to enhance recommendation accuracy based on mood and emotion.

5) Model Development:

- Deep Learning Models: Develop deep learning models such as CNNs and RNNs to learn intricate patterns and embeddings from audio and textual features.
- Ensemble Learning: Investigate ensemble learning techniques to combine predictions from multiple models for improved recommendation accuracy.

6) Evaluation and Performance Metrics:

- Metrics for Cold Start Problem: Propose metrics to assess the effectiveness of cold start mitigation techniques, considering factors like recommendation coverage and diversity for new users and items.
- Recommendation Quality Metrics: Utilize metrics such as precision, recall, F1-score, and NDCG to evaluate the quality of recommendations provided by the system.

V. Conclusion

The ”Building a Song Recommendation System with Spotify Million Playlist Dataset” project is a promising endeavor that addresses the growing need for personalized music recommendations.

The methodology primarily involves content-based filtering, utilizing song features and metadata, along with user-defined parameters to fine-tune recommendations. To address the cold start problem and ensure scalability, the project explores hybrid recommendation techniques, meta-learning approaches, distributed data processing using technologies like Apache Hadoop and Spark, and scalable storage solutions. By aggregating data from various sources, incorporating audio and textual analysis, and developing deep learning models, this project seeks to design an advanced recommendation system that enhances user engagement and satisfaction.
Evaluation metrics encompass both cold start mitigation effectiveness and recommendation quality.

In conclusion, our music recommendation system stands as a testament to the effectiveness of combining TF-IDF and cosine similarity models. The seamless integration of textual and audio features, multilingual adaptability, and a well-architected MongoDB filter system collectively contribute to a personalized, user-centric music discovery experience. As the system evolves with user interactions and dataset updates, it continues to hit the right notes, providing a symphony of recommendations that resonates with the diverse musical preferences of users.

REFERENCES

1) Dataset: https://www.aicrowd.com/challenges/spotify-million-playlist-dataset-challenge
2) https://towardsdatascience.com/part-iii-building-a-song-recommendation-system-with-spotify-cf76b52705e7
3) https://www.codewithfaraz.com/content/147/create-a-stunning-spotify-clone-project-with-html-and-css
4) https://paperswithcode.com/task/recommendation-systems
5) https://thecleverprogrammer.com/2021/03/03/spotify-recommendation-system-with-machine-learning/