Music2Movie:
Personalized Recommendation System from Sound to Screen

Marina Mitiaeva, Arunava Das, Raghuveera Narsimha

Introduction

In an era where personalized content is the cornerstone of user engagement, recommendation systems have become integral to enhancing entertainment experiences. While platforms have excelled at curating suggestions within a single medium—such as recommending movies or music—cross-domain recommendations remain a largely untapped frontier. Imagine a system that could suggest movies based on the vibe and mood of your favorite songs, seamlessly connecting the auditory and visual worlds. Such an innovation not only enriches content discovery but also deepens emotional resonance by aligning two powerful storytelling mediums. This project explores the exciting intersection of music and movies, leveraging advancements in artificial intelligence and deep learning to deliver a novel approach to cross-media recommendations. By combining different music and cinema features, this effort seeks to redefine how users engage with entertainment, offering a dynamic, mood-driven journey from sound to screen.

Prediction, Inference, and Goals

The primary goal of this project is to develop a predictive system that generates a numerical match score between 0 and 1 as the final output, quantifying the compatibility between a song and a movie. The inputs to the model will include audio features from songs, such as valence, energy, tempo, and genre, extracted from the Spotify Tracks dataset, and movie attributes, such as sentiment (derived from descriptions), genre, runtime, and popularity, sourced from the TMDb Movies dataset. The prediction task focuses on evaluating how well the mood, energy, and stylistic features of a song align with those of a movie. Inference goals aim to uncover deeper insights into cross-domain relationships, such as the role of musical energy in aligning with action genres or the influence of mood positivity on compatibility. Beyond predictions, the project seeks to establish a scalable, data-driven framework that can enhance cross-media content discovery and inform personalization strategies for entertainment platforms.

Dataset Description

We plan to leverage two datasets in this project: one from Spotify [1] and one from the TMDb [2] Movies dataset. The **Spotify dataset** sourced from Hugging Face contains audio features and metadata for songs available on Spotify, capturing various musical characteristics such as danceability, energy, mood, and loudness. Each row corresponds to a unique track, offering attributes that define its rhythmic, tonal, and structural qualities. In our project, the Spotify data will be linked with the movie dataset to recommend movies based on the mood and characteristics of music tracks using deep learning models.

Number of Rows and Columns for the Spotify dataset

- **Rows (Tracks):** 113,999
- **Columns (Features):** 20

The **TMDb Movies dataset** sourced from Kaggle provides detailed metadata for over 930,000 movies. This dataset captures essential attributes about each film, such as genre, production

companies, and popularity, which will allow us to align movies with the characteristics of music tracks to create meaningful recommendations.

Number of Rows and Columns for the TMDb Movies dataset

- **Rows (Movies):** 1,110,815
- **Columns (Features):** 24

Preliminary Data Exploration

For Spotify the exploratory data analysis (EDA) involved using descriptive statistics to summarize key metrics, along with histograms and density plots to visualize the distribution of popularity and tempo. Box plots were used to compare the impact of explicit versus non-explicit content on popularity, while bar charts highlighted the most popular genres. A scatter plot with popularity encoding revealed relationships between valence and danceability, and a correlation heatmap was generated to explore relationships among audio features, such as the strong correlation between energy and loudness.

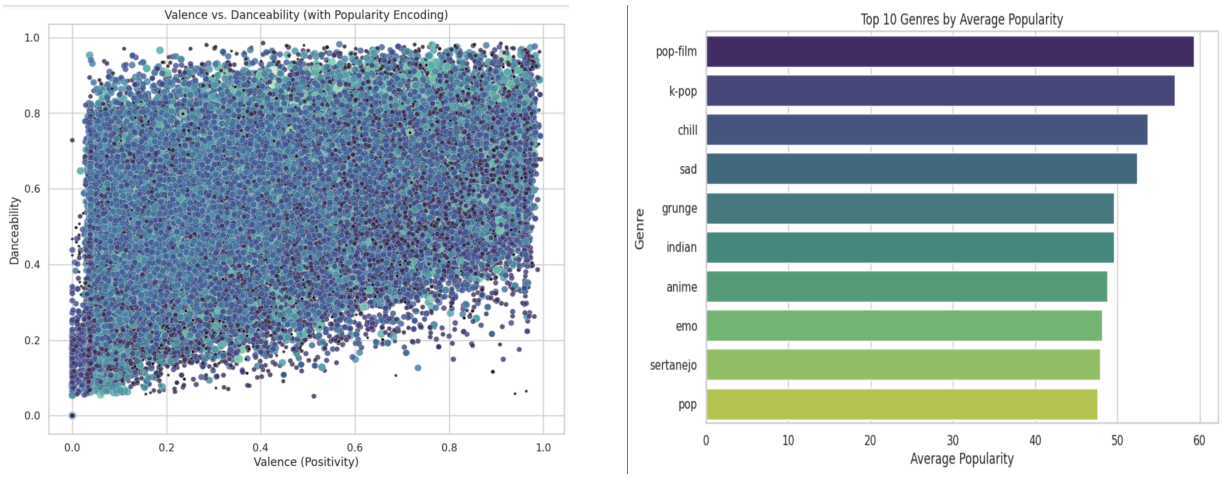

Fig 1. Valence vs Danceability with Popularity Encoding and Genres with highest average popularity

One of the most interesting findings is the role of **danceability** in driving popularity. Although valence (emotional positivity) and danceability are positively correlated, danceability has a stronger impact on track success. Some songs with lower valence still achieve high popularity due to their rhythmic appeal, suggesting that listeners value danceability over mood when selecting music. This insight emphasizes the importance of rhythm in commercial music, particularly for tracks aimed at social or recreational settings. Additionally, the **genre analysis** reveals that niche genres like anime and grunge enjoy surprisingly high popularity, demonstrating that mainstream genres do not dominate as much as expected. These findings highlight the growing influence of niche communities and their ability to push non-mainstream content to

prominence. Lastly, the **weak correlations between popularity and most audio features** underscore the role of external factors such as marketing, playlist placement, and exposure in determining a song's success, rather than its intrinsic musical properties. This insight suggests that strategies beyond production quality—such as visibility and promotion—play a critical role in a track's performance.

For the TMDb dataset, we conducted an extensive analysis to determine which features are most relevant from the recommendation system perspective. In addition to the descriptive methods applied earlier, we ran a correlation analysis between numerical features, which revealed notable relationships such as a strong correlation between revenue and budget (0.69) and between vote count and revenue (0.7). We also explored unparsed categorical data, assessing the relationships between genres, production countries, and spoken languages using Cramér's V correlation, though these relationships appeared weak. To gain further insights, we generated a word cloud from the keywords associated with movies, revealing that many keywords reflect explicit or mature content.

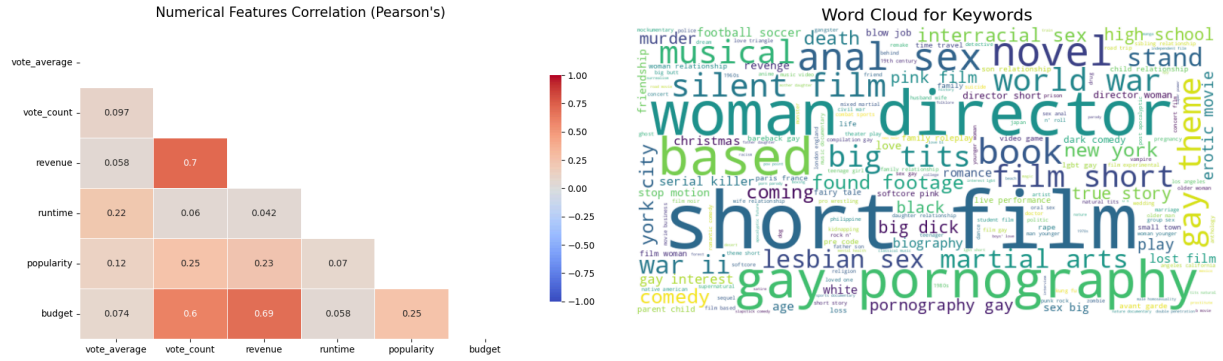

Fig 2. Correlation analysis for numerical features and word cloud for keywords in the TMDb dataset.

An interesting comparison emerged when examining the explicit content across both datasets. While only a small portion (8.5%) of tracks in the Spotify dataset are marked explicit, keywords in the TMDb movie data frequently contain mature or explicit terms. This discrepancy between music and movie content is noteworthy, and we plan to explore whether it will impact our cross-domain recommendation analysis as the project progresses. The combination of these insights provides a robust foundation for aligning mood, themes, and genres across music and movie data, guiding the development of our recommendation system.

Methodology

In our methodology, the goal was to design a recommendation system, relying on [3] and [4], that takes a song title and artist name as input and generates two key outputs: five recommended movie titles (externally visible) and their corresponding similarity scores (internally calculated).

Following our exploratory data analysis (EDA), we undertook preprocessing of the TMDB dataset to prepare it for modeling. The key preprocessing steps were as follows:

1. **Duplicate Removal:** Eliminated duplicate rows to ensure data consistency.

2. **Handling Missing Values:** Identified and removed rows with missing values in critical columns, including title, overview, poster_path, popularity, and genres. These features were deemed essential for prediction, and their missing values could not be reasonably imputed.

3. **Filtering by Release Status:** Retained only movies with a release_status of "Released," excluding others to focus on finalized, publicly available content.

4. **Explicit Content Exclusion:** Filtered out movies flagged as explicit (adult = True) as they were unlikely to contribute meaningfully to the recommendation system.

5. **Column Reduction:** Dropped irrelevant columns, such as ids and adult, which either contained excessive null values or were not beneficial for our recommendation system.

After completing the cleaning steps, the dataset was reduced to **436,591 rows and 13 columns**, representing **24%** of the **original dataset** (1,111,181 rows). Despite this significant reduction, it remained larger than the **114,000 unique songs** in the Spotify dataset. Given our goal of associating five unique songs with each movie, this dataset size was sufficient to meet our requirements.
In the TMDB dataset, we utilized the available links to movie posters as an additional source of information. To extract meaningful features from these images, we employed a pre-trained **MobileNetV2** model, fine-tuned for image classification tasks, to generate feature embeddings. To achieve this, we implemented specialized functions to retrieve and save poster images from the provided URLs. The images were downloaded using a multi-threaded approach, which significantly reduced download time. To ensure robustness, failed download attempts were logged for further inspection and potential retries. Once downloaded, the poster images were resized to **32x32 pixels** and preprocessed using the MobileNetV2 pipeline. The extracted embeddings offer a compact numerical representation of the visual content, providing an additional layer of rich, high-dimensional information to enhance the movie dataset.

We also incorporated **sentiment analysis** for the textual column describing movies, leveraging **TextBlob**, a library for natural language processing. TextBlob computes a **sentiment polarity score** ranging from **-1** (negative sentiment) to **1**(positive sentiment). These sentiment scores provide insights into the emotional tone of movie descriptions and were included as a feature in the final movie embeddings. This step ensures that textual data, a critical aspect of movie characterization, is effectively represented and integrated into the model.

To ensure compatibility between movie and song features, we performed **feature normalization** using **Min-Max Scaling**. For movies, attributes such as popularity and runtime were normalized to a uniform range. Similarly, for Spotify features, attributes like popularity and tempo were normalized. This step was crucial to align the essence

of features between the two datasets, facilitating meaningful comparisons. To prepare the datasets for merging, we carefully selected relevant features from both datasets based on their impact and alignment:

- **Movie Features:** (vote_average, vote_count, revenue, runtime, popularity_normalized, runtime_normalized, overview_sentiment, poster_features)
- **Spotify Features:** (popularity_normalized, tempo_normalized, danceability, energy, loudness, valence, speechiness, acousticness, instrumentalness)

We explored various methods for merging the datasets, including scaling-based matching and creating a **shared embedding space** [5]. The embedding space approach yielded better results, as it allowed us to leverage more features and simultaneously perform two critical tasks: merging the data and finding similarities between movies and songs. Our merging process involved an **embedding construction pipeline**, which combined multiple features into a unified representation. This approach captured the intrinsic relationships across datasets while preserving the unique characteristics of each. Each training record consisted of concatenated embedding vectors derived from both movie and Spotify features, creating a unified representation.

We employed **cosine similarity** to measure the alignment between song and movie embeddings [5]. This metric evaluates the angle between two vectors, emphasizing their directional alignment rather than their magnitude, making it particularly suitable for comparing multi-dimensional embeddings. For a given song, its embedding was compared against all movie embeddings to compute similarity scores. These scores quantify the degree of alignment between the song and each movie, serving as the foundation for ranking and generating recommendations.

To generate recommendations, the top 5 movies with the highest similarity scores for a given song were selected. These similarity scores were appended to the movie dataset, and the details of the top-ranked movies were extracted. Additionally, metadata from the input Spotify songs, such as track_name, tempo, and popularity, was retained and included in the output. While some features were not used directly in training or embedding creation, they were preserved in the dataset to support additional functionality in the user interface. This includes advanced filtering options for songs, enabling users to go beyond just relying on the song name and artist. This approach ensures a richer, more flexible, and customizable user experience.

Predictions

The prediction phase focuses on generating movie recommendations for a given input song. Once the system receives a song's metadata (e.g., song title and artist name), the following steps are executed:

1. Input Processing:
   - The input song's features, including its metadata and audio-based embeddings, are processed to ensure compatibility with the trained model.
   - Metadata, such as track_name, tempo, popularity, and other relevant features, is incorporated into the recommendation pipeline.

2. Similarity Calculation:
    - The song embedding is compared against all movie embeddings in the dataset using cosine similarity.
    - Similarity scores are calculated to determine the alignment between the input song and each movie.
3. Top-5 Recommendations:
    - The top 5 movies with the highest similarity scores are selected as recommendations.
    - Each recommended movie is ranked based on its similarity score, ensuring that the most relevant movies are presented first.
4. Result Formatting:
    - The recommendation output includes detailed information about the selected movies, such as their title, overview, popularity, and genres.
    - Metadata from the input song, such as track_name, tempo, and popularity, is included for context.
    - Additional filters or attributes from the dataset, retained during preprocessing, are offered to allow users to customize their results further.

Currently, our recommendation system relies solely on cosine similarity to rank movies based on their alignment with song embeddings, without a labeled dataset to directly evaluate recommendation relevance. In future iterations, we plan to integrate metrics like precision and recall to systematically assess and improve the system's performance. By incorporating user feedback mechanisms, such as allowing users to rate or mark recommendations as relevant, we can create a labeled dataset to serve as ground truth. Precision will measure the proportion of relevant movies among the recommendations, while recall will evaluate how effectively the system identifies all relevant movies. This feedback-driven approach will enable iterative improvements, potentially combining cosine similarity with machine learning models to predict relevance more accurately. Ultimately, integrating these metrics will ensure the recommendation system evolves to provide more meaningful and user-centric results.

Results

The developed pipeline successfully generates movie recommendations by embedding similarity between Spotify track features and movie attributes. The system's functionality is demonstrated in **Fig. 3**, which provides an example of the code workflow and output for generating movie recommendations based on a selected song.

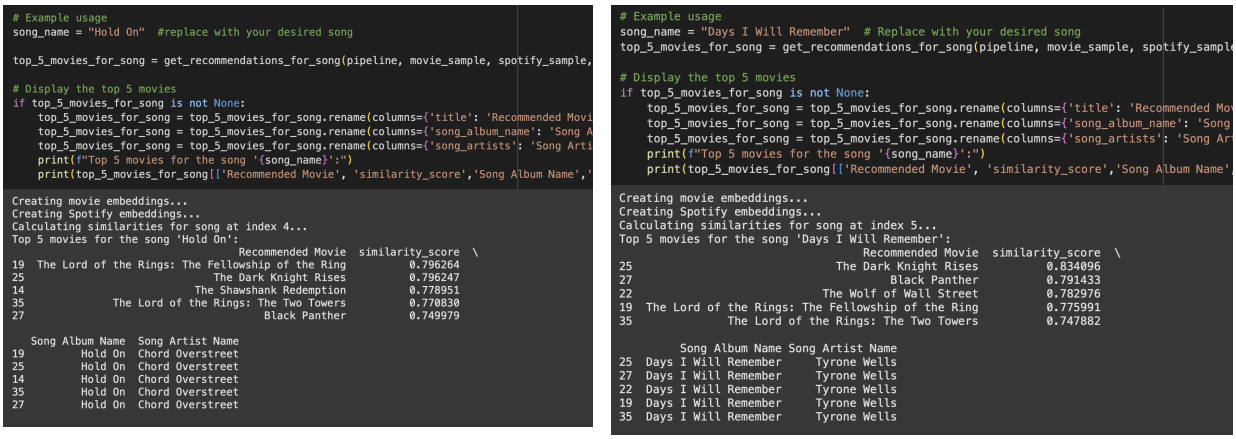

Fig 3. Example Output and Code Workflow for Song-to-Movie Recommendations

For the track "Hold On" by Chord Overstreet, the top five recommended movies are: The Lord of the Rings: The Fellowship of the Ring (0.796), The Dark Knight Rises (0.796), The Shawshank Redemption (0.779), The Lord of the Rings: The Two Towers (0.771), and Black Panther (0.750).

For the song "Days I Will Remember" by Tyrone Wells, the top five recommendations are: The Dark Knight Rises (0.834), Black Panther (0.791), The Wolf of Wall Street (0.783), The Lord of the Rings: The Fellowship of the Ring (0.776), and The Lord of the Rings: The Two Towers (0.748).

The user interface (UI) for our Song-to-Movie Recommendation System was designed to provide an intuitive and seamless user experience. Users can simply select a song from the dropdown menu, and the system displays the top recommended movies along with their similarity scores. Additional features, such as a toggle to view detailed recommendations, enhance usability and make the tool both interactive and insightful (see **Fig. 4**).

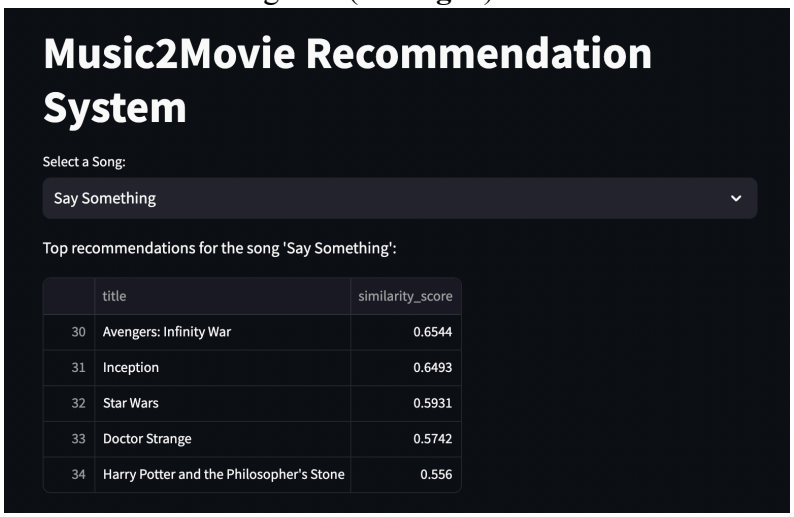

Fig 4. Interactive User Interface for Music2Movie Recommendations

Overall, the results validate the effectiveness of our approach, demonstrating that the calculated similarity scores align well with the mood, atmosphere, and vibe of the input song. At the same time, the recommended movies appear thematically and stylistically compatible with each other, showcasing the strength of the features we engineered. The integration of poster embeddings and sentiment analysis plays a pivotal role in capturing the visual and emotional essence of movies, while audio-based features like tempo, danceability, and energy ensure a meaningful connection with the song. This synergy highlights the potential of leveraging cross-domain embeddings to create cohesive and context-aware recommendations, paving the way for innovative tools in personalized entertainment curation.

References

[1] Maharshipandya. (2023). *Spotify Tracks Dataset*. Hugging Face. Retrieved from https://huggingface.co/datasets/maharshipandya/spotify-tracks-dataset
[2] Asaniczka. (2023). *TMDb Movies Dataset 2023 (930K+ Movies)*. Kaggle. Retrieved from https://www.kaggle.com/datasets/asaniczka/tmdb-movies-dataset-2023-930k-movies
[3] Stratoflow. (2022). *How to Build a Recommendation System: Explained Step by Step.* Retrieved from https://stratoflow.com/how-to-build-recommendation-system/
[4] GeeksforGeeks. (2023). *What are Recommender Systems?* Retrieved from https://www.geeksforgeeks.org/what-are-recommender-systems/
[5] Chen, C.-M., Wang, C.-J., Tsai, M.-F., & Yang, Y.-H. (2019). *Collaborative Similarity Embedding for Recommender Systems.* Retrieved from https://arxiv.org/pdf/1902.06188