

Mood-Based Movie Recommendation System Documentation

This documentation provides an overview of the code for a mood-based movie recommendation system implemented using FastAPI and various data processing libraries. The system recommends movies based on user input, historical movie preferences, and mood analysis.

Code Structure

The code is structured into the following sections:

1. Import Statements:

- The code begins with importing necessary Python libraries and modules.

2. FastAPI Initialization:

- An instance of the FastAPI framework is created and named `mood_app`.

3. Data Loading and Preprocessing:

- Movie data is loaded from CSV files ("movies_metadata.csv" and "top_1000_movie_reviews.csv").
- A subset of movies with the "Thriller" genre is selected for mood analysis.
- Initial movie recommendations and user history are generated.

4. Sentence Embedding Model:

- The SentenceTransformer model ("all-mpnet-base-v2") is loaded to convert movie plot descriptions, reviews, and genres into embeddings.
- The device (CPU or CUDA) for processing is determined.

5. Utility Functions:

- Several utility functions are defined for calculating embeddings and mood scores, as well as for recommending movies based on user history and user search.

6. Data Preprocessing and Mood Calculation:

- Movie data is preprocessed and mood scores are calculated for all movies.
- User history embeddings and mood scores are computed.
- The similarity between user history and movie mood scores is calculated and stored.

7. FastAPI Endpoints:

- The code defines three FastAPI endpoints:
- `/history/` (GET):
 - Returns the user's historical movie preferences in JSON format.
- `/recommendations/` (GET):
 - Recommends movies based on user history and returns the top 10 recommendations in JSON format.
 - The similarity score is also included.
- `/search/` (POST):
 - Allows users to search for movies based on a mood input.
 - Returns the top 10 search results with their similarity scores.

8. Data Serialization:

- The similarity scores in movie data are converted to float to avoid JSON serialization issues.

9. Running the Application:

- The FastAPI application is run, and it listens for incoming requests.

Usage

1. /history/ Endpoint:

- Access this endpoint via a GET request to retrieve the user's historical movie preferences.

2. /recommendations/ Endpoint:

- Access this endpoint via a GET request to get movie recommendations based on user history.
- The recommendations are provided in JSON format, including movie titles, IMDb IDs, overviews, similarities, and genres.

3. /search/ Endpoint:

- Access this endpoint via a POST request, providing a mood input.
- The system performs a mood-based search and returns the top 10 search results in JSON format, including movie titles, IMDb IDs, overviews, similarities, and genres.

Technical Details

- The system utilizes SentenceTransformer for semantic text embeddings.
- Cosine similarity is used to calculate the similarity between user preferences and movie mood scores.
- User history is used to recommend movies with the most similar mood scores.

Acknowledgments

This code is a simplified example of a mood-based movie recommendation system. It can be extended and improved for production use with additional features, a larger dataset, and further user interaction.