Movie Recommendation System

Introduction

The Movie Recommendation System is a data-driven application designed to provide movie recommendations to users based on their input movie title. This project utilizes collaborative filtering techniques and string similarity calculations to suggest movies that are similar to the user's preferences. This report discusses the key features and functions of the project.

Features and Functions

1. Data Loading and Preprocessing

Functionality: The project begins by loading movie data and user ratings data from CSV files (movies.csv and ratings.csv) into Pandas DataFrames (movies and ratings). It then merges these DataFrames based on the 'movieId' column to create a user-item matrix.

Purpose: Data loading and preprocessing are essential steps in preparing the dataset for recommendation. Merging the data allows us to associate user ratings with movie titles and create a user-movie matrix for collaborative filtering.

2. User-Item Matrix

Functionality: The user-item matrix is created from the merged data, where rows represent users, columns represent movie titles, and the values represent user ratings. Any missing values (NaN) in this matrix are filled with 0.

Purpose: The user-item matrix is the foundation for collaborative filtering. It represents user interactions with movies and is used to calculate similarities between movies.

3. Cosine Similarity Calculation

Functionality: Cosine similarity is calculated between item vectors, which are the columns of the user-movie matrix transposed. This results in an item-item similarity matrix (item_similarity_df), where each cell represents the similarity between two movies.

Purpose: Cosine similarity is used to quantify the similarity between movies. This similarity matrix is a crucial component of the recommendation system as it forms the basis for finding similar movies.

4. find_closest_movie Function

Functionality: The find_closest_movie function takes a user's input movie title, the dataset of movie titles, and an optional threshold for similarity. It employs fuzzy string matching (using the fuzz.token_set_ratio scorer from the fuzzywuzzy package) to find the closest matching movie title in the dataset based on the user's input. If the similarity score of the

closest match is above or equal to the threshold, it returns the closest matching movie title; otherwise, it returns None.

Purpose: This function helps identify a relevant movie in the dataset that closely matches the user's input, even if there are minor discrepancies in the movie title.

5. recommend_movies Function

Functionality: The recommend_movies function is responsible for generating movie recommendations based on user input. It first calls the find_closest_movie function to find the closest matching movie. If a match is found, it retrieves similarity scores for movies similar to the closest movie from the item_similarity_df. It then returns the top recommendations based on these similarity scores. The number of recommendations and the threshold for similarity can be customized.

Purpose: This function forms the core of the recommendation system. It leverages collaborative filtering to recommend movies that are similar to the user's input movie title.

6. User Input and Recommendations

Functionality: The user provides a movie title as input (e.g., "Titanic"). The project then calls the recommend_movies function to generate movie recommendations. If recommendations are available, it prints the recommended movies for the user's input. If no matching movie is found, it prints a message stating that no matching movie was found.

Purpose: This is the user-facing aspect of the project. Users interact with the system by inputting a movie they like, and the system responds with relevant movie recommendations.

Conclusion

The Movie Recommendation System project demonstrates the application of collaborative filtering and string similarity techniques to provide personalized movie recommendations. It encompasses data loading and preprocessing, user-item matrix creation, cosine similarity calculation, and user-friendly recommendation generation. This system can be further enhanced with larger datasets, advanced recommendation algorithms, and additional user features to provide even more accurate and personalized recommendations to users.