ELEVATE LABS PROJECT REPORT

ON:-

Movie Recommendation System

Name:ANANYA PODDAR

Introduction

To develop a movie recommendation system that suggests top 5 movies to users based on their preferences using content-based filtering, leveraging metadata like genres, cast, crew, and movie descriptions. With the exponential growth of movies and online streaming platforms, users often find it difficult to choose what to watch next. Recommendation systems play a vital role in solving this problem by personalising suggestions based on users' interests. This project focuses on building a content-based movie recommendation system that works without explicit user ratings. It uses movie metadata to understand similarities between films and predict recommendations effectively.

Abstraction

The Movie Recommendation System is a machine learning-based project that suggests relevant movies to users based on their preferences. The system uses a content-based filtering approach by analysing key features such as cast, crew, genres, keywords, and plot overview. By leveraging the metadata available in Kaggle's movies.csv and credits.csv datasets, the system identifies similarities between movies and recommends the top 5 closest matches. The project includes a simple user interface using Stream-lit to take user input and display recommendations.

Tools and Technologies Used

- Programming Language: Python
- Libraries: Pandas, NumPy, Scikit-learn, Ast, Streamlit
- Datasets Used:
    - movies.csv – Contains movie metadata such as title, genres, overview, and keywords
    - credits.csv – Includes information on cast and crew
- ML Techniques: TF-IDF / Count Vectorizer, Cosine Similarity

Steps Involved in Building the Project

Step 1: Data Collection

- Downloaded movies.csv and credits.csv from Kaggle.
- Merged both datasets using the movie title as the key.

Step 2: Data Preprocessing

- Removed null values and unnecessary columns.
- Extracted features such as cast, crew (director), genres, and keywords using ast.literal_eval().
- Combined all useful features into a single tags column.

Step 3: Feature Extraction

- Applied text normalization: lowercasing, removing spaces, punctuation, and stopwords.
- Used CountVectorizer to convert text into numerical vectors.

Step 4: Similarity Computation

- Used cosine similarity to compute the similarity between movies based on their tags vector.

Step 5: Building the Recommendation Function

- Defined a function that takes a movie title as input and returns the top 5 most similar movies.

Step 6: Designing the UI with Stream-lit

- Built an interactive web interface using Stream-lit.
- User selects a movie title from a dropdown.
- System outputs top 5 movie recommendations in real-time.

Conclusion

The movie recommendation system developed in this project efficiently suggests similar movies based on the input title. By using content-based filtering and natural language processing techniques, the system provides meaningful recommendations using metadata. The project showcases how machine learning can enhance the user experience in entertainment applications. Future improvements can include hybrid models combining collaborative filtering and sentiment analysis using reviews for more accurate predictions.