Project Report

Project Title : Movie Recommendation System

Name : Murari Yamini

Introduction :

For this present age of digitalism, movie recommendation systems serve an essential function in maximizing user satisfaction on streaming sites by recommending content related to the users' interests. The goal of this project is to develop a hybrid movie recommender based on combining collaborative filtering and content-based filtering methods with the MovieLens 1M dataset. The system accepts a movie that the user likes as input and suggests movies of similarity based on user behavior patterns and genre similarities.

Abstract :

This film recommender system uses user ratings and film genres to construct a hybrid recommender system. Collaborative filtering employs patterns in user ratings to suggest movies with similar ratings, while content-based filtering employs genre similarity (transformed through TF-IDF vectorization). Both of these methods are combined to generate the final recommendation list. The application offers an easy-to-use interface where the user can enter any movie title and receive recommendations right away.

Tools Used:

Technology stack	Purpose
Python	Programming language
Pandas	for preprocessing and Data handling
scikit-learn	for TF-IDF vectorization and cosine similarity
streamlit	for web app deployment
MovieLens 1M Dataset	contains movies.dat and ratings.dat

Steps Involved in Building the Project :

- **Data Loading and Preprocessing**
  - Loaded `movies.dat` and `ratings.dat` with appropriate separators and encoding.
  - Merged both datasets using `movieId`

- **Collaborative Filtering**

  - Constructed a **user-movie rating matrix** using `pivot_table`.
  - Calculated **cosine similarity** between movie vectors.
  - Built a recommendation function based on similarity scores.

- **Content-Based Filtering**

  - preprocessed the genres column.
  - Converted genres into TF-IDF vectors.
  - Calculated cosine similarity between these genre vectors.
  - Stored results in a similarity matrix.

- **Hybrid Recommendation**

  - Created a function to combine results from both filtering techniques.
  - Ensured unique and relevant movie recommendations.

- **Web Interface with Streamlit**

  - Developed a clean UI with a movie search feature.
  - Suggested possible matches using partial search.
  - Displayed recommendations upon selection and button click.

- **Conclusion**

  - This project well illustrates how hybrid recommender systems can enhance precision by combining user activity and content similarity. With the MovieLens 1M dataset and Python's data science libraries, it provides a real-world application with a simple-to-use interface using Streamlit. The system is extensible to a greater extent by adding user login functionality, further metadata such as tags or directors, or implementing it as a full-stack web application.

How to run app :

- Download code and open it in Visual Studio code
- Open new terminal
- Install streamlit by using command :'pip install streamlit'

Command is : streamlit run app1.py

- It will open in web browser .