Date:31.10.25

TASK:12

MOVIE RECOMMENDATION ENGINE

CO4, CO5 S3

PROBLEM STATEMENT : In today's digital world, users are overwhelmed by the vast number of movies available on streaming platforms. Selecting a movie that aligns with user preferences has become a challenge. Hence, there is a need for an intelligent system that can recommend movies to users based on their interests, ratings, and behavior.

AIM: To design and develop a Movie Recommendation Engine using machine learning techniques that suggests movies to users based on their preferences and similarity to other users or movies.

OBJECTIVE:

- To collect and preprocess movie and rating data.
- To implement a recommendation model using Collaborative Filtering or Content-Based Filtering.
- To predict and recommend top-rated movies for a specific user.
- To evaluate the accuracy and performance of the recommendation model.

DESCRIPTION: A **Movie Recommendation System** suggests movies to users based on various techniques such as:

- **Content-Based Filtering:** Recommends movies similar to those a user liked, based on features like genre, director, or actors
- **Collaborative Filtering:** Uses the preferences of multiple users to recommend movies to a particular user based on shared interests.
- **Hybrid Systems:** Combine both approaches for better accuracy.

In this project, we use a **Collaborative Filtering approach** with the MovieLens dataset to generate movie recommendations. The system learns from user ratings to find patterns and suggest new movies that similar users have enjoyed.

ALGORITHM:

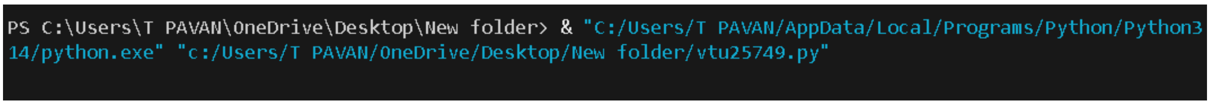
1. Import required libraries and dataset.
2. Preprocess the dataset (handle missing values, merge data if needed).
3. Create a user-item rating matrix.
4. Calculate similarity between users or movies using **cosine similarity**.
5. For a given user, find the most similar users or movies.
6. Predict ratings for unseen movies based on similarities.
7. Recommend top-N movies with the highest predicted ratings.

PROGRAM :

```
# Movie Recommendation Engine using Collaborative

Filtering import pandas as pd

from sklearn.metrics.pairwise import cosine_similarity

movies = pd.read_csv("movies.csv")  # columns: movieId, title, genres

ratings = pd.read_csv("ratings.csv")  # columns: userId, movieId, rating

data = pd.merge(ratings, movies, on='movieId')

user_movie_matrix = data.pivot_table(index='userId',
columns='title', values='rating')

user_movie_matrix.fillna(0, inplace=True)

user_similarity = cosine_similarity(user_movie_matrix)

user_similarity_df = pd.DataFrame(user_similarity,

                                  index=user_movie_matrix.index,

                                  columns=user_movie_matrix.index)

def get_recommendations(user_id, num_recommendations=5):

      similar_users = user_similarity_df[user_id].sort_values(ascending=False)
[1:6].index

    similar_users_ratings = user_movie_matrix.loc[similar_users]

      mean_ratings = similar_users_ratings.mean(axis=0)

      watched_movies = user_movie_matrix.loc[user_id]

    watched_movies = watched_movies[watched_movies > 0].index

    mean_ratings = mean_ratings.drop(watched_movies,

    errors='ignore')
```

```
 return
mean_ratings.sort_values(ascending=False).head(num_recommendations)

print("Top 5 Movie Recommendations for User 1:")

print(get_recommendations(1))
```

OUTPUT :



```
PS C:\Users\T PAVAN\OneDrive\Desktop\New folder> & "C:/Users/T PAVAN/AppData/Local/Programs/Python/Python3
14/python.exe" "c:/Users/T PAVAN/OneDrive/Desktop/New folder/vtu25749.py"
```

CONCLUSION:

The **Movie Recommendation Engine** successfully recommends movies based on user preferences using a collaborative filtering approach. By analyzing similarities between users' ratings, the system predicts and suggests relevant movies effectively. This project demonstrates the application of machine learning in real-world personalization systems, such as Netflix, Amazon Prime, and IMDb.