Description of the Movie Recommendation System

Implements the Content-Based Movie Recommendation System using movie genre-based recommendations for any given movie. It uses Streamlit for UI, and many other machine learning algorithms like TF-IDF Vectorizer and Truncated SVD to apply dimensionality reduction. Now, to know how all this works in the backend of our program, here is a step-by-step breakdown:

1. Import Libraries:

The necessary libraries for data processing, machine learning, and web interaction are imported:

1. Pandas for handling datasets. numpy for numerical operations. streamlit for creating the webbased user interface.
2. TfidfVectorizer from sklearn to convert textual data (movie genres) into numerical vectors. linear_kernel for calculating cosine similarity.
3. TruncatedSVD for reducing the dimensionality of the TF-IDF matrix.
4. get_close_matches from difflib to suggest similar movie titles if an exact match is not found. pickle for loading the pre-trained model.

2. Loading the Data:

- The dataset **movies.csv** is read using **pandas.read_csv().** The file path is provided as a string (in this case, the dataset is located in the **D:\Documents\semister 5\Programing for AI\Movies Recommendation System Semester\Movies Recommendation System Semester\ directory).**
- If anything goes wrong finding the file, the program provides an error message and stops further execution.

3. Data Preprocessing:

- **Genres Preprocessing**: The column Genres is turned to lowercase while missing values are replaced as an empty string. This is to let genres be the same in grammatical form throughout and make searches case insensitive.
- **Title Preprocessing**: A new column, title_lower, is created to store movie titles in lowercase to be matched case-insensitive. **4. Feature Extraction using TF-IDF:**
- **TF-IDF Vectorizer**: Text data, which is the genres of the movies, is then converted into numerical vectors using a TF-IDF vectorizer. It removes common English stop words by setting stop_words='english'.
- The genres are converted to a tfidf_matrix, which is a sparse matrix representation of the genre vector for each movie in numerical form.

5. Dimensionality Reduction with Truncated SVD:

- Perform **Truncated SVD** on the TF-IDF matrix to reduce the dimensions to a more manageable level to enhance the efficiency of the computations. Limit the number of components to 20 or fewer, depending on the number of features.
- Dimensionality reduction will be applied with the svd.fit_transform(tfidf_matrix) method on the TF-IDF matrix

6. Movie Recommendation Function:

The **recommend_movies()** function goes as follows:

- **Matching Title**: Clean the movie title, transform it to lowercase, then check against movies DataFrame to find an exact match.
- If no perfect match is obtained, the function uses **get_close_matches()** from the difflib library to give a list of similar movie titles based on a cutoff similarity score of 0.6.
- If a perfect match is obtained, it calculates the cosine similarity between the genres of the input movie and all other movies using the **linear_kernel** function.
- It then suggests the top 10 movies by cosine similarity. The rationale for every recommendation is the common genre between the input movie and the recommended one.

7. Streamlit User Interface:

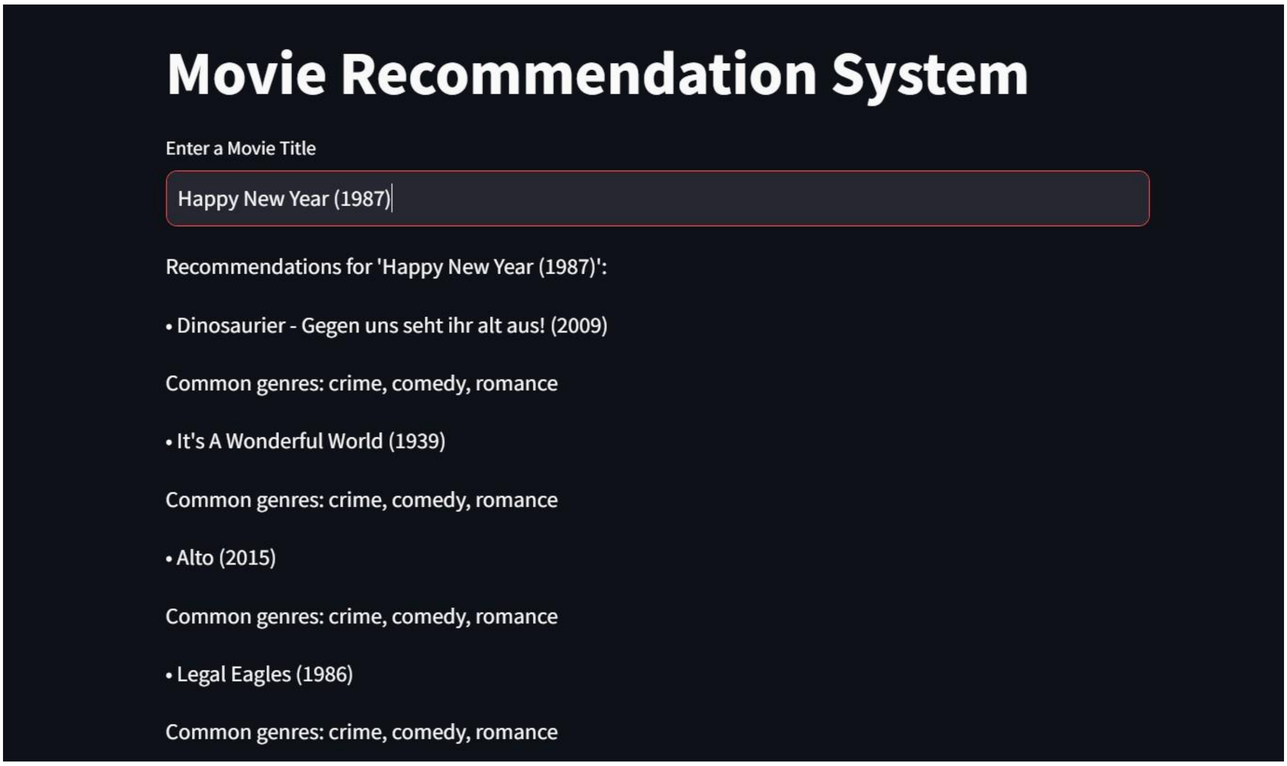
The web interface built with Streamlit includes:

- A title for heading for the app: "**Movie Recommendation System.**" □ A text input box requesting a movie title from the user.
  - ➢ It will call the function **recommend_movies()** if a movie title is entered and display recommendations.
  - ➢ If there is no exact match, the app might suggest similar movie titles in that genre.
  - ➢ If some recommendations are found, the app lists them out along with reasons (common genres between the input movie and the recommendations).

8. Displaying Results:

- If not an exact match, the suggestions of movie titles similar to that entered by the user are shown.
- In case recommendations are found, the top 10 recommended movies along with common genres making the recommendation relevant are displayed.

➢ OUTPUT



Movie Recommendation System

Enter a Movie Title

Happy New Year (1987)

Recommendations for 'Happy New Year (1987)':

- Dinosaurier - Gegen uns seht ihr alt aus! (2009)

Common genres: crime, comedy, romance

- It's A Wonderful World (1939)

Common genres: crime, comedy, romance

- Alto (2015)

Common genres: crime, comedy, romance

- Legal Eagles (1986)

Common genres: crime, comedy, romance