DSC630

Week 10 solution

Abhishek Srivastava

10.2 Assignment: Recommender System

Using the small MovieLens data set, create a recommender system that allows users to input a movie they like (in the data set) and recommends ten other movies for them to watch. In your write-up, clearly explain the recommender system process and all steps performed. If you are using a method found online, be sure to reference the source.

1. **Understanding the Task and Data**

- The goal is to recommend movies to a user based on a movie they say they like.
- The underlying data contains movies, their titles, and their genres, but no user ratings or explicit feedback.
- The chosen recommendation method is **content-based filtering**, which means recommendations are generated by comparing the content (genres) of the user-selected movie with all others in the database.[1][2]

---

2. **Preparing the Movie Data**

- The movie data is loaded into a pandas DataFrame from an Excel sheet.
- Each movie's `genres` column (example: `Comedy|Romance`) is transformed into a Python set (example: `{"Comedy", "Romance"}`).
  - This allows easy comparison for overlap between genre sets.

---

3. **Getting Input From the User**

- The code asks the user to enter the title of a movie they like using the command line.
- It prints some sample movie titles first to make it easier for the user to select correctly.

---

4. **Finding the Selected Movie's Genres**

- The system searches the DataFrame for the row whose title matches the user's input.
- It extracts the set of genres for that movie:
  - For example, if the user picks `"Toy Story (1995)"`, the genre set is `{"Adventure", "Animation", "Children", "Comedy", "Fantasy"}`.

---

5. **Computing Genre Similarity**

- For every other movie in the database:
  - The program compares its genre set with the selected movie's genre set.

- Similarity is computed by counting the number of genres in common (the intersection of the two sets).
- The more genres two movies share, the higher their similarity score.

---

6. **Sorting and Selecting Recommendations**

- All movies (except the one chosen by the user) are ranked by their similarity score.
- The top ten movies with the most genre overlap with the user-selected movie are chosen as recommendations.

---

7. **Displaying Recommendations**

- The code prints the recommended movies, their genres, and their similarity score (number of overlapping genres) for the user to see.

---

8. **Summary of Steps Performed**

1. **Load and Process Data:** Prepare movies and genres for comparison.
2. **User Input:** Ask the user to select a movie.
3. **Extract Chosen Movie's Genres:** Find out which genres that movie belongs to.
4. **Compute Genre Overlap:** Compare every movie's genres to the chosen movie's genres.
5. **Sort and Recommend:** Find and display the ten most similar movies by genre.

Method Reference

This process is called **content-based filtering**, a well-established recommendation technique for situations where you have item features (like genres), but you don't have extensive user data or ratings.[2][1]

---

[1] (https://www.geeksforgeeks.org/machine-learning/python-implementation-of-movie-recommender-system/)

[2] (https://www.kaggle.com/code/ibtesama/getting-started-with-a-movie-recommendation-system)

[3] (https://stratoflow.com/how-to-build-recommendation-system/)


```
import pandas as pd

# Step 1: Load the movie data from the Excel file
df = pd.read_excel('movies.xlsx')

print(df.head(2))

# Step 2: Preprocess genres by splitting into sets
```

```
df['genre_set'] = df['genres'].apply(lambda x: set(str(x).split('|')))

print(df.head(2))
```

```
   movieId              title
genres
0        1  Toy Story (1995)  Adventure|Animation|Children|Comedy|
Fantasy
1        2    Jumanji (1995)                   Adventure|Children|
Fantasy
   movieId              title
genres  \
0        1  Toy Story (1995)  Adventure|Animation|Children|Comedy|
Fantasy
1        2    Jumanji (1995)                   Adventure|Children|
Fantasy

                                             genre_set
0  {Children, Adventure, Comedy, Fantasy, Animation}
1                      {Adventure, Children, Fantasy}
```

```
# Step 3: Define recommendation function based on genre similarity
def recommend_movies(title, df, top_n=10):
    if title not in df['title'].values:
        print('Movie not found. Please check spelling and
punctuation.')
        return []
    selected_genres_set = df[df['title'] == title]
['genre_set'].iloc[0]
    def genre_similarity(genres_set):
        return len(selected_genres_set.intersection(genres_set))
    df['similarity'] = df['genre_set'].apply(genre_similarity)
    recommendations = df[df['title'] != title]
    recommended_movies = recommendations.sort_values(by='similarity',
ascending=False).head(top_n)
    return recommended_movies[['title', 'genres', 'similarity']]


# Step 4: Prompt the user for a movie they like, displaying some
sample choices
print("Example movie titles from dataset:")
print(df['title'].head(10).to_list())
user_movie = input("Please enter the exact name of a movie you like
from the list above (or full dataset): ")

# Step 5: Get recommendations and print them, if available
recommendations = recommend_movies(user_movie, df)
if len(recommendations) > 0:
    print(f"\nBecause you like: {user_movie}, you might also enjoy:")
    print(recommendations)
```

```
else:
    print("No recommendations found. Please verify your movie title.")


Example movie titles from dataset:
['Toy Story (1995)', 'Jumanji (1995)', 'Grumpier Old Men (1995)',
'Waiting to Exhale (1995)', 'Father of the Bride Part II (1995)',
'Heat (1995)', 'Sabrina (1995)', 'Tom and Huck (1995)', 'Sudden Death
(1995)', 'GoldenEye (1995)']

Because you like: Grumpier Old Men (1995), you might also enjoy:
                                    title                    genres
similarity
508              Radioland Murders (1994)   Comedy|Mystery|Romance
2
676              Mrs. Winterbourne (1996)            Comedy|Romance
2
435                           Dave (1993)            Comedy|Romance
2
538  So I Married an Axe Murderer (1993)  Comedy|Romance|Thriller
2
884          It Happened One Night (1934)            Comedy|Romance
2
886              Gay Divorcee, The (1934)   Comedy|Musical|Romance
2
537                    Son in Law (1993)     Comedy|Drama|Romance
2
888                 Apartment, The (1960)     Comedy|Drama|Romance
2
67    French Twist (Gazon maudit) (1995)            Comedy|Romance
2
534           Sleepless in Seattle (1993)     Comedy|Drama|Romance
2
```