k200183-user-based-cf

March 13, 2023

```python
[143]: import pandas as pd
       import numpy as np
       from sklearn.metrics.pairwise import cosine_similarity

       ratings_df = pd.read_csv(r'C:\Users\Bilal\Desktop\RS
        ↪assignment\ml-latest-small\ratings.csv')
       movies_df = pd.read_csv(r'C:\Users\Bilal\Desktop\RS
        ↪assignment\ml-latest-small\movies.csv')
       movie_ratings_df = pd.merge(ratings_df, movies_df, on='movieId')
       user_means = movie_ratings_df.groupby('userId')['rating'].mean()
       movie_ratings_df['rating'] = movie_ratings_df['rating'] -
        ↪movie_ratings_df['userId'].map(user_means)

       ratings_matrix = movie_ratings_df.pivot_table(index='userId', columns='title',
        ↪values='rating')

       ratings_matrix.fillna(0, inplace=True)

       user_similarity = cosine_similarity(ratings_matrix)

       input_movie = input("Enter the movie name: ")

       movie_index = ratings_matrix.columns.get_loc(input_movie)

       weighted_sum = np.dot(user_similarity, ratings_matrix.iloc[:, movie_index])

       sum_of_similarities = np.sum(user_similarity, axis=1)

       predicted_ratings = user_means + (weighted_sum / sum_of_similarities)

       recommendations = pd.Series(predicted_ratings, index=ratings_matrix.index).
        ↪sort_values(ascending=False)[:5]

       print("Top-5 recommended movies:")
       for i, movie in enumerate(recommendations.index):
           print(movies_df.loc[movie].title)
```

```
Enter the movie name: Tom and Huck (1995)
Top-5 recommended movies:
Once Were Warriors (1994)
Wild Bunch, The (1969)
Othello (1995)
Dangerous Minds (1995)
Hellraiser: Bloodline (1996)

C:\Users\Bilal\AppData\Local\Temp\ipykernel_14248\2649398282.py:25:
RuntimeWarning: invalid value encountered in true_divide
  predicted_ratings = user_means + (weighted_sum / sum_of_similarities)
```

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]: