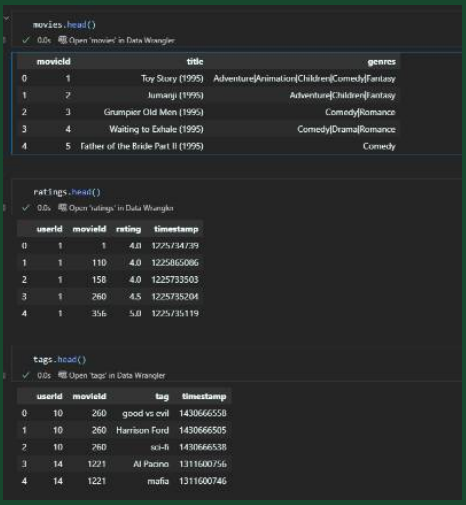
ChatGPT - Movie Recommend



```
movies.head()
```

✓ 0.0s Open 'movies' in Data Wrangler

	movieId	title	genres
0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
1	2	Jumanji (1995)	Adventure|Children|Fantasy
2	3	Grumpier Old Men (1995)	Comedy|Romance
3	4	Waiting to Exhale (1995)	Comedy|Drama|Romance
4	5	Father of the Bride Part II (1995)	Comedy

```
ratings.head()
```

✓ 0.0s Open 'ratings' in Data Wrangler

	userId	movieId	rating	timestamp
0	1	1	4.0	1225734739
1	1	110	4.0	1225865086
2	1	158	4.0	1225733503
3	1	260	4.5	1225735204
4	1	356	5.0	1225735119

```
tags.head()
```

✓ 0.0s Open 'tags' in Data Wrangler

	userId	movieId	tag	timestamp
0	10	260	good vs evil	1430666558
1	10	260	Harrison Ford	1430666505
2	10	260	sci-fi	1430666538
3	14	1221	Al Pacino	1311600756
4	14	1221	mafia	1311600746

So, there are three data frames you can see. I need to understand which of the columns can I use for hybrid recommendation system. I want to use collaborative filtering as well as this content-based filtering using cosine similarities. So, help me from this data frame to identify.

To build a hybrid recommendation system using both collaborative filtering and content-based filtering, we can utilize columns from these three data frames you provided:

1. **Collaborative Filtering**: Collaborative filtering relies on user-item interactions (e.g., user ratings). From the data frames you provided:
    - **ratings**: The `userId`, `movieId`, and `rating` columns will be useful. You can use `userId` and `movieId` to identify which movies users have rated, and the `rating` column will provide feedback that will be used to build the collaborative filtering model.
2. **Content-Based Filtering**: This method uses the attributes of the items themselves (in this case, the movies). From the data frames:
    - **movies**: The `movieId`, `title`, and `genres` columns will be useful. You can treat the `genres` column as the item's features (as a list of genres). By transforming this text data into numerical form (e.g., using one-hot encoding or TF-IDF), you can calculate cosine similarity between items based on their genres. The `movieId` will connect it with the `ratings` data frame.
3. **Combining Both**: After computing similarity scores from the content-based filtering approach (using genres), and the collaborative filtering approach (using user ratings), you can combine the results for a hybrid recommendation system.

Here's a quick outline of how the data will be used:

- **Collaborative Filtering**: Use the `ratings` data for user-item interactions (userId, movieId, rating).
- **Content-Based Filtering**: Use the `movies` data for item attributes (movieId, title, genres). You can compute the cosine similarity based on the genres of the movies.

By combining both approaches, you'll create a hybrid recommendation system. Would you like help on specific implementation details for this?

give me the approach

Here's an approach to build a hybrid recommendation system using **Collaborative Filtering** and **Content-Based Filtering** with the data you provided.

Step 1: Data Preparation

1.1 Collaborative Filtering Data (Ratings Data)

- We'll use the `ratings` data frame to build a collaborative filtering model. This requires user-movie interactions, which are captured in the `userId`, `movieId`, and `rating` columns.

1.2 Content-Based Filtering Data (Movies Data)

- We'll use the `movies` data frame to extract movie features (i.e., the genres). The `movieId` and `genres` columns will help us create a feature vector for each movie.

Step 2: Collaborative Filtering (Matrix Factorization)

2.1 Create User-Item Rating Matrix

- Create a user-item rating matrix where each row represents a user, and each column represents a movie.
- The values in the matrix will be the ratings given by users to each movie.

python

```python
import pandas as pd

# Create a user-item matrix with ratings
user_movie_ratings = ratings.pivot(index='userId', columns='movieId', values='rating')
```

2.2 Matrix Factorization Using SVD (Singular Value Decomposition)

- Apply matrix factorization to decompose the user-item rating matrix. This is commonly done with **SVD** (Singular Value Decomposition) or **ALS** (Alternating Least Squares).