URL to Colab:

https://colab.research.google.com/drive/1ZJQV0vvD0e689mzUyPLm70rZ5kLVaEv9

Project Explanation:

The project is to utilize multiple movie data sources to create a movie recommendation system. Four methods are included in this recommendation engine, which are content-based, demographic-based filtering, collaborative filtering and hybrid methods.

Data Processing:

Since I collected datasets from different resources, the data cleaning process that requires most of the time is to query the movie attribute information and merge them into the same table. What should be mentioned that is even though there are movie IDs collected in different tables, the data type could be different. It is important to check the data type before you merge them.

The other data processing part is to ensure what each column really represents in each table. For example, the movie ID in our table could simply represents the order of movies, while the movie ID represents IMDb movie ID in another table.

Data Modeling:

Content-Based Filtering Recommendation System

Content-based filtering recommendation system is built based on the similarity score between movies to solve cold-start problems. The similarity score is computed by the following formula:

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

where A_i and B_i are components of vector A and B respectively. In the content-based filtering recommendation system, vector A and B represent the two movies that we want to calculate the similarity score of and are composed by the features of these two movies. The features include movie key words, major casts, director, genre, popularity, vote average and overview.

Users can enter their favorite movies, and the system will return the movies with the top ten largest similarity scores of this user's favorite movie.

For example, the user whose favorite movie is Superman Returns will get the following movie recommendation list:

```
content_based_filter('Superman Returns')
```

```
1296                          Superman III
14                            Man of Steel
870                            Superman II
813                               Superman
64                       X-Men: Apocalypse
203                                     X2
511                                   X-Men
9       Batman v Superman: Dawn of Justice
232                           The Wolverine
46              X-Men: Days of Future Past
```

Demographic-Based Filtering Recommendation System

Demographic-based filtering method is another way to deal with the cold-start problem. Since we don't have any information about those users' interests, we will provide them the most popular movies alternatively. This recommendation system is built based on the following weighted rating formula defined by IMDb (1):

$$\text{Weighted Rating (WR)} = \left(\frac{v}{v+m}.R\right) + \left(\frac{m}{v+m}.C\right)$$

The top five most popular movies in IMDb data sets are: The Shawshank Redemption, The Dark Knight, Fight Club, Inception and Pulp Fiction.

Collaborative-Filtering Recommendation System

In collaborative-filtering recommendation system, we will utilize Latent Factor Model to capture the similarities between both users and movies and find the hidden factors that influence users' preference. No movie description is required but only users' historical data of movie tastes. Singular Value Decomposition algorithm, which decreases the dimension of the utility matrix by extracting latent factors, is adopted to get the minimal Root Mean Square Error (RMSE) and achieve better model performance. (2)

In our model, the mean RMSE is 0.89, and mean MAE is 0.68. By utilizing the **Surprise** package in Python, we achieved the following movie recommendation result for user 'tt011304':

```
cf('tt0113041',10)
```

```
Your recommended movies are:  ['The Shawshank Redemption' 'The Godfather' 'Modern Times' 'All About Eve'
 'On the Waterfront' 'Roger & Me' '乱' 'The African Queen'
 'The Usual Suspects' 'The Third Man']
```

Hybrid Recommendation System

Hybrid recommendation system combines both collaborative-filtering and content-based filtering methods. We will take the user ID and their favorite movies as input, and then recommend similar movies with users' favorite movies with expected scores from that particular user.

Here is the recommendation for user "tt01133041" who likes Avatar:

```
cf_rec('tt0113041','Avatar',7)
```

```
Your recommended movies are:  ['Moon' 'Drugstore Cowboy' '卧虎藏龙' '22 Jump Street' 'Flags of Our Fathers'
 'Out of Africa' 'Joyeux Noël']
```

Project Conclusion:

Even content-based filtering recommendation system is a good solution to address the cold-start problem, it has a limitation in predicting movies across the genres. In addition, it cannot capture users' personal taste and favors. It will always return the same recommendation result no matter who the users are, as long as they enter the same favorite movie as the input for the engine. (1) Collaborative-filtering recommendation system based on Singular Value Decomposition algorithm can resolve this problem by capturing the similarities between both users and movies.

Hybrid recommendation system combines ideas from both content-based and collaborative filtering algorithms to predict expected scores of the similar movies that a particular user likes.

Reference

1. https://www.kaggle.com/ibtesama/getting-started-with-a-movie-recommendation-system/data
2. https://hackernoon.com/introduction-to-recommender-system-part-1-collaborative-filtering-singular-value-decomposition-44c9659c5e75