Looking for something to watch? Let’s see what you’ve watched in the past.

MOVIE RECOMMENDER SYSTEM

- Oscar Mulei
- Rose Kyalo
- Angel Linah Atungire

HELLO EVERYONE!

Our goal is to enhance the movie-watching experience by suggesting the top films to users based on their unique rating history and the ratings from similar users.

Data Preprocessing and EDA

Model Development, Evaluation and Tuning

Does it actually work?

DATA PRE-PROCESSING AND EDA

- Included cleaning the data, handling missing values, and structuring it in a way that's conducive to our analysis.

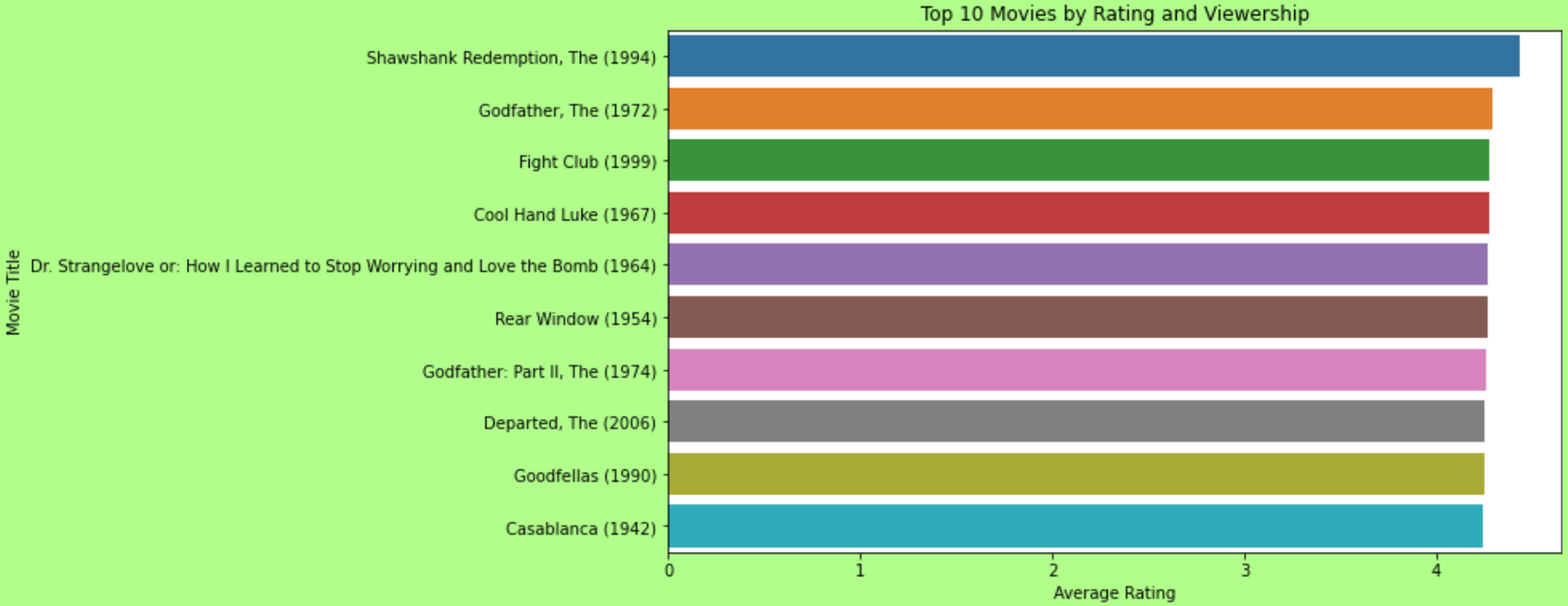
- We applied some EDA to our dataset and through the distribution of the genres, ratings and movies, we are able to gain a deeper understanding of our data.

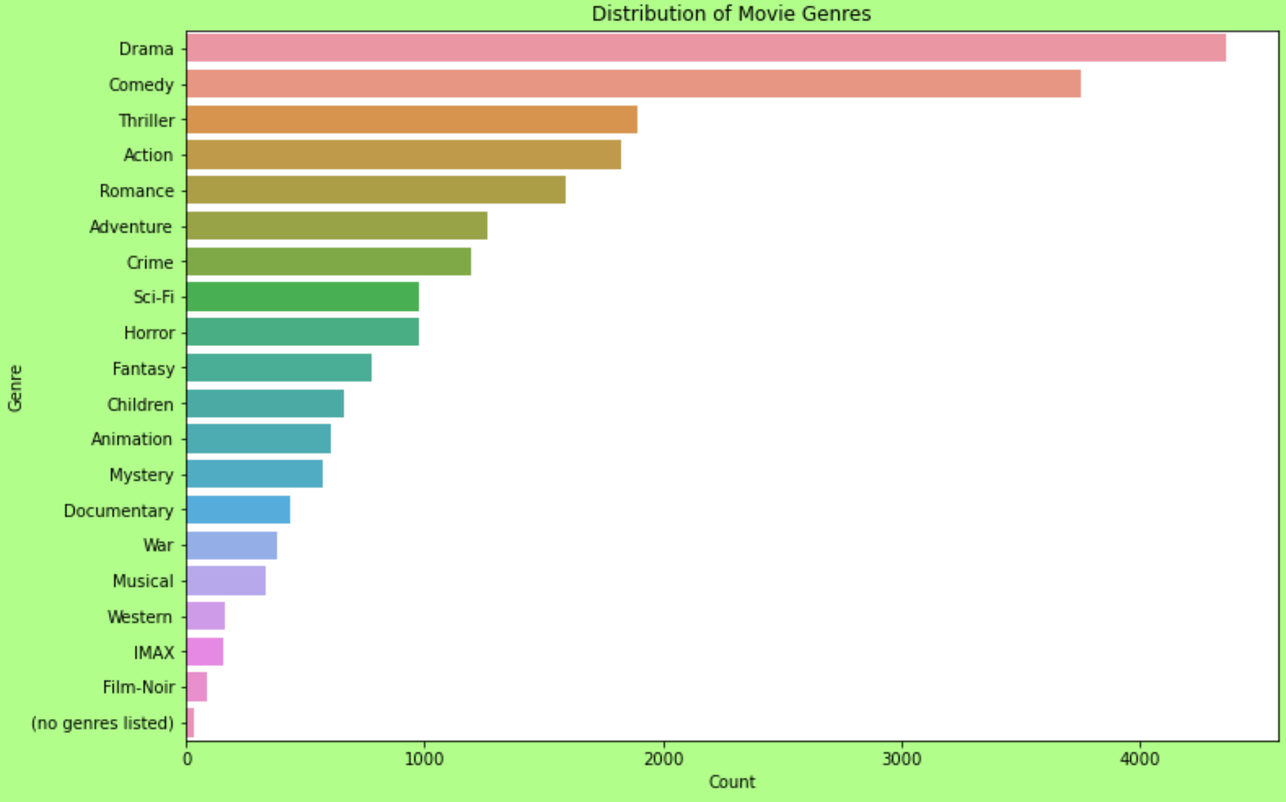
TOP MOVIES AND GENRES BY RATINGS

From our Exploratory Data Analysis, we found that these movies and their respective genres were the most popular.

Action	Crime
• Fight Club (1999)	• Cool Hand Luke (1967)
Horror	Drama
• The Shawshank Redemption (1994)	• The Godfather
Comedy	Adventure
• Dr. Strangelove (1964)	• Jurassic Park (1993)
Thriller	Romance
• Rear Window (1954)	• Casablanca (1942)

Top 10 Movies





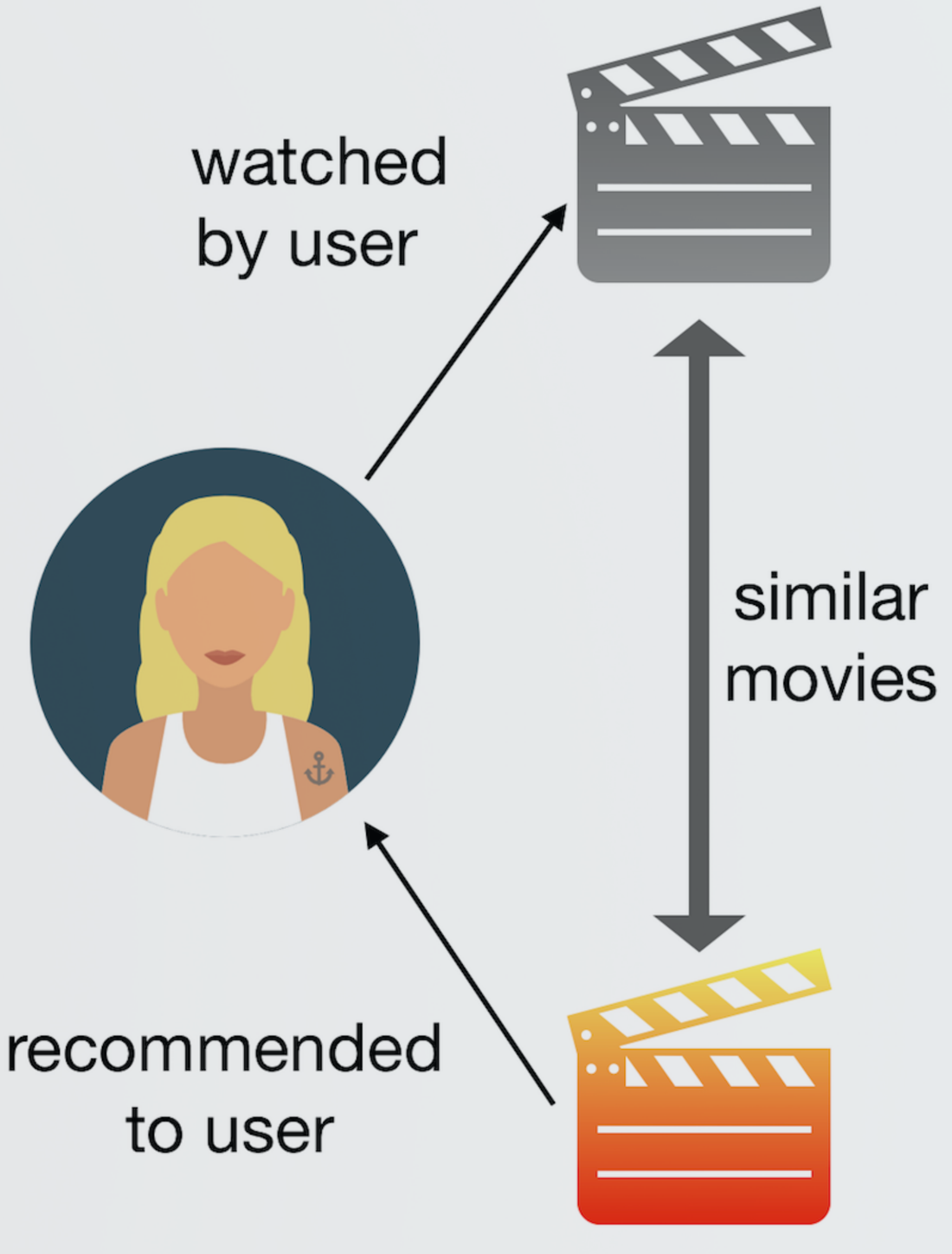
Genre Distribution

THE MODEL

CONTENT BASED RECOMMENDATION SYSTEM

- A Content-Based Recommendation System is a type of recommendation system that suggests items based on a comparison between the content of the items and a user profile.

- We created an interactive widget in our notebook for both genres and the movie titles.



THE RUN DOWN

Content-Based Recommendation System Focused on Movie title
for further details check out the repository
https://github.com/omulei/Movie_Recommender_System/blob/main/Phase%204%20project.ipynb

STEPS	WHAT THEY DO
1. Creating a TF-IDF(Term Frequency- Inverse Document Frequency)	• transforming raw text data (movie titles) into a structured, numerical format that can be effectively utilized in recommendation algorithms
2. Creating a search function: content-based recommendation system	• exemplifies a content-based recommendation system for movies in natural language processing.
3. Building an interactive search	• it executes a search function to find and display relevant movie titles.
4. Finding Users Who Liked the Same Movie	• identify users who liked the same movie based on their ratings.
5. Finding Movies Highly Rated by Similar Users	• find movies that are highly rated by users who have similar tastes

THE RUN DOWN

A step by step process describing how we created our recommendation system, further details can be found in the following repository https://github.com/omulei/Movie_Recommender_System/blob/main/Phase%204%20project.ipynb

STEPS	WHAT THEY DO
6. Calculate Proportion of Similar Users Who Liked Each Movie	• calculate the proportion of similar users who liked each movie.
7. Finding How Much All Users Like Movies Popular Among Similar Users and Calculate the Number of Times Each Movie Was Recommended to Users	• analyze how all users like movies that are popular among the group of similar users and calculate the number of times each movie was recommended to users
8. Calculate a Recommendation Score	• create a recommendation score by combining the recommendation percentages from similar users and all users.
9. Create a recommendation function	• that takes a movie ID as input and returns a DataFrame of recommended movies based on user ratings.
10 . Interactive Recommendation Widget	• the widget will display recommended movies based on their preferences.

Movies screenshot

```
movie_name_input.observe(on_type, names='value')

# Display the movie title input and the recommendation list.
display(movie_name_input, recommendation_list)
```

Movie Title: iron man

	score	title	genres
6743	18.322581	Iron Man (2008)	[Action, Adventure, Sci-Fi]
7324	18.322581	Iron Man 2 (2010)	[Action, Adventure, Sci-Fi, Thriller, IMAX]
8301	14.658065	Day of the Doctor, The (2013)	[Adventure, Drama, Sci-Fi]
7620	14.658065	X-Men: First Class (2011)	[Action, Adventure, Sci-Fi, Thriller, War]
8151	13.087558	Iron Man 3 (2013)	[Action, Sci-Fi, Thriller, IMAX]
8425	12.825806	X-Men: Days of Future Past (2014)	[Action, Adventure, Sci-Fi]
8699	12.215054	Untitled Spider-Man Reboot (2017)	[Action, Adventure, Fantasy]
8695	11.451613	Guardians of the Galaxy 2 (2017)	[Action, Adventure, Sci-Fi]
6746	11.275434	Taken (2008)	[Action, Crime, Drama, Thriller]
8053	10.688172	Hobbit: An Unexpected Journey, The (2012)	[Adventure, Fantasy, IMAX]

WE BUILT A RECOMMENDATION WIDGET...

... FOR ONE TO SEARCH USING EITHER MOVIE TITLE OR GENRE

Genres screenshot

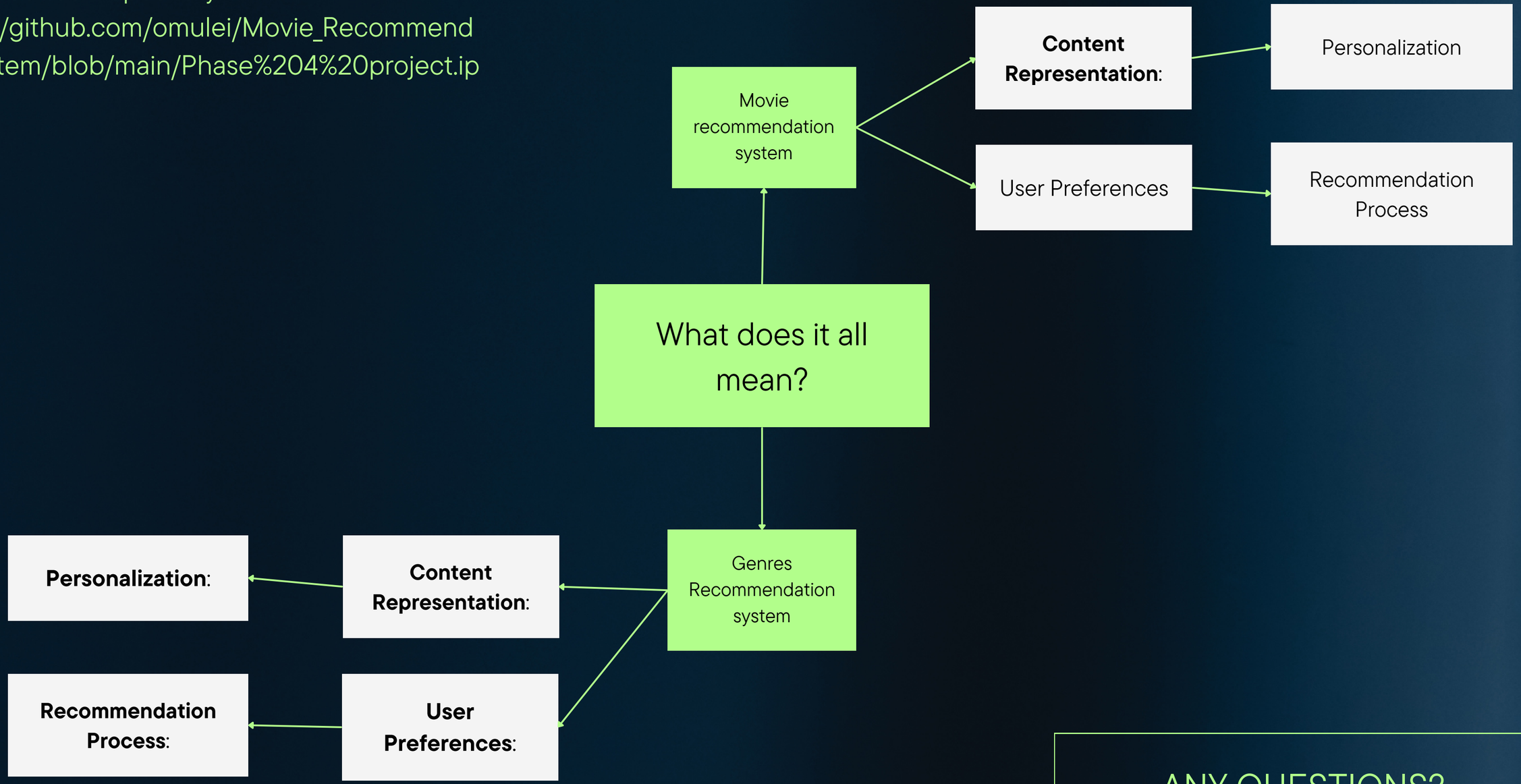
```
genre_input = widgets.Text(value='', description='Genres:')
output_widget = widgets.Output()

def on_input_change(change):
    with output_widget:
        output_widget.clear_output()
        if change['new']:
            display(find_similar_movies(change['new']))

genre_input.observe(on_input_change, names='value')

display(genre_input, output_widget)
```

Genres: romance

	movieId	title	genres	score
1745	2340	Meet Joe Black (1998)	[Romance]	1.0
1140	1493	Love and Other Catastrophes (1996)	[Romance]	1.0
1151	1514	Temptress Moon (Feng Yue) (1996)	[Romance]	1.0
5601	26958	Emma (1996)	[Romance]	1.0
541	638	Jack and Sarah (1995)	[Romance]	1.0

Need more details?

Check out our repository https://github.com/omulei/Movie_Recommender_System/blob/main/Phase%204%20project.ipynb



ANY QUESTIONS? ...

CONCLUSION

We have successfully developed a movie recommendation system with the following objectives:

1. **Precision in Recommendations**
2. **Enhancement of User Engagement**
3. **Generation of Personalized Recommendations**

The inclusion of an interactive recommendation widget further elevates the user experience by allowing users to input movie titles and receive immediate recommendations based on their preferences.

Our recommendation system leverages user behavior and movie ratings to provide accurate and personalized movie suggestions. This not only increases user interaction but also aligns with the distinct interests of each user.