

## Import Required Libraries

+ Code

+ Text

```
1 import pandas as pd
2 import numpy as np
3 import matplotlib.pyplot as plt
4 import seaborn as sns
```

## Load or Create Your Dataset

```
1 df = pd.read_csv('/content/movies.csv')
2 df.head()
```

	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

Next steps:

Generate code with df

View recommended plots

New interactive sheet

```
1 df.tail()
```

	movieId	title	genres
62418	209157	We (2018)	Drama
62419	209159	Window of the Soul (2001)	Documentary
62420	209163	Bad Poems (2018)	Comedy Drama
62421	209169	A Girl Thing (2001)	(no genres listed)
62422	209171	Women of Devil's Island (1962)	Action Adventure Drama

## Exploring The DataSet

```
1 df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 62423 entries, 0 to 62422
Data columns (total 3 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   movieId    62423 non-null  int64
1   title      62423 non-null  object
2   genres     62423 non-null  object
dtypes: int64(1), object(2)
memory usage: 1.4+ MB
```

```
1 df.columns
```

```
Index(['movieId', 'title', 'genres'], dtype='object')
```

## Preprocess Genres into Sets

```
1 # Fill null genres with empty strings
2 df['genres'] = df['genres'].fillna('')
3
4 # Convert genre strings to sets
5 df['genre_set'] = df['genres'].apply(lambda x: set(x.lower().replace('|', ',').split(',')))
```

## Define Jaccard similarity Function

```
1 def jaccard_similarity(set1, set2):
2     intersection = len(set1 & set2)
3     union = len(set1 | set2)
4     if union == 0:
5         return 0
6     return intersection / union
```

✓ Build the Recommendation Function

```
1 def recommend_by_genre(movie_title, top_n=5):
2     if movie_title not in df['title'].values:
3         return f"'{movie_title}' not found in dataset."
4
5     target_genres = df[df['title'] == movie_title]['genre_set'].values[0]
6
7     similarities = []
8
9     for index, row in df.iterrows():
10         if row['title'] == movie_title:
11             continue
12         sim = jaccard_similarity(target_genres, row['genre_set'])
13         similarities.append((row['title'], sim))
14
15     # Sort by similarity score
16     similarities.sort(key=lambda x: x[1], reverse=True)
17
18     top_recommendations = [title for title, sim in similarities[:top_n]]
19
20     return top_recommendations
```

```
1 recommend_by_genre("We (2018)")
```

➡ ['Nixon (1995)',  
'Othello (1995)',  
'Dangerous Minds (1995)',  
'Cry, the Beloved Country (1995)',  
'Restoration (1995)']

Summary

In this project, I developed a content-based recommender system using the movies.csv dataset. The dataset includes movie titles and their associated genres.

We used Jaccard similarity to compare the genre sets of movies. This approach measures how similar two movies are based on the proportion of shared genres. Each movie’s genre was converted into a set, and similarity was calculated based on set intersection and union.

The recommender system:

Takes a movie title as input (e.g., "We (2018)")

Finds the most similar movies based on genre overlap

Returns the top N recommendations

Conclusion

This project demonstrates a simple yet effective method for building a genre-based movie recommender system using Jaccard similarity. Unlike more complex models, this approach is lightweight, fast, and easy to interpret.

While it does not consider user preferences or behavioral data, it is a great starting point for exploring recommendation systems. It can be extended in the future by:

Overall, this method is ideal for content-based filtering when minimal user interaction data is available.