```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

df = pd.read_csv('https://github.com/YBI-Foundation/Dataset/raw/main/Movies%20Recommendati
```

```
df.head()
```

	Movie_ID	Movie_Title	Movie_Genre	Movie_Language	Movie_Budget	Movie_Popularit
0	1	Four Rooms	Crime Comedy	en	4000000	22.87623
1	2	Star Wars	Adventure Action Science Fiction	en	11000000	126.39369
2	3	Finding Nemo	Animation Family	en	94000000	85.68878
3	4	Forrest Gump	Comedy Drama Romance	en	55000000	138.13333
4	5	American Beauty	Drama	en	15000000	80.87860

5 rows × 21 columns

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4760 entries, 0 to 4759
Data columns (total 21 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   Movie_ID                4760 non-null   int64  
 1   Movie_Title             4760 non-null   object 
 2   Movie_Genre             4760 non-null   object 
 3   Movie_Language          4760 non-null   object 
 4   Movie_Budget            4760 non-null   int64  
 5   Movie_Popularity        4760 non-null   float64
 6   Movie_Release_Date      4760 non-null   object 
 7   Movie_Revenue           4760 non-null   int64  
 8   Movie_Runtime           4758 non-null   float64
 9   Movie_Vote              4760 non-null   float64
 10  Movie_Vote_Count        4760 non-null   int64  
 11  Movie_Homepage          1699 non-null   object 
 12  Movie_Keywords          4373 non-null   object 
 13  Movie_Overview          4757 non-null   object 
 14  Movie_Production_House  4760 non-null   object 
 15  Movie_Production_Country  4760 non-null   object 
 16  Movie_Spoken_Language   4760 non-null   object 
 17  Movie_Tagline           3942 non-null   object 
 18  Movie_Cast              4733 non-null   object 
 19  Movie_Crew              4760 non-null   object 
 20  Movie_Director          4738 non-null   object 
dtypes: float64(3), int64(4), object(14)
memory usage: 781.1+ KB
```

```
df.shape
```

```
(4760, 21)
```

```
df.columns
```

```
Index(['Movie_ID', 'Movie_Title', 'Movie_Genre', 'Movie_Language',
       'Movie_Budget', 'Movie_Popularity', 'Movie_Release_Date',
       'Movie_Revenue', 'Movie_Runtime', 'Movie_Vote', 'Movie_Vote_Count',
       'Movie_Homepage', 'Movie_Keywords', 'Movie_Overview',
       'Movie_Production_House', 'Movie_Production_Country',
       'Movie_Spoken_Language', 'Movie_Tagline', 'Movie_Cast', 'Movie_Crew',
       'Movie_Director'],
      dtype='object')
```

```
df_features = df[['Movie_Genre','Movie_Keywords','Movie_Tagline','Movie_Cast','Movie_Direc
df_features.shape
```

```
(4760, 5)
```

```
df_features.head()
```

	Movie_Genre	Movie_Keywords	Movie_Tagline	Movie_Cast	Movie_Director
0	Crime Comedy	hotel new year's eve witch bet hotel room	Twelve outrageous guests. Four scandalous requests. And one lone bellhop, in his first day on th...	Tim Roth Antonio Banderas Jennifer Beals Madonna Marisa Tomei	Allison Anders
1	Adventure Action	android galaxy hermit death star	A long time ago in a	Mark Hamill Harrison Ford Carrie Fisher	George Lucas

```
pd.options.display.max_colwidth = 100
```

```
X = df_features['Movie_Genre']+' '+df_features['Movie_Keywords']+' '+df_features['Movie_Ta
X[:5]
```

```
0    Crime Comedy hotel new year's eve witch bet hotel room Twelve outrageous guests
1    Adventure Action Science Fiction android galaxy hermit death star lightsaber A
2    Animation Family father son relationship harbor underwater fish tank great barr
3    Comedy Drama Romance vietnam veteran hippie mentally disabled running based on
4    Drama male nudity female nudity adultery midlife crisis coming out Look closer.
dtype: object
```

```
from sklearn.feature_extraction.text import TfidfVectorizer
```

```
tfid = TfidfVectorizer()
X = tfid.fit_transform(X)
X.shape
```

```
(4760, 17928)
```

```
from sklearn.metrics.pairwise import cosine_similarity
```

```
simi_score = cosine_similarity(X)
```

```
simi_score
```

```
array([[1.        , 0.01348508, 0.03572908, ..., 0.        , 0.        ,
        0.        ],
       [0.01348508, 1.        , 0.00806353, ..., 0.        , 0.        ,
        0.        ],
       [0.03572908, 0.00806353, 1.        , ..., 0.        , 0.0802874 ,
        0.        ],
       ...,
       [0.        , 0.        , 0.        , ..., 1.        , 0.        ,
        0.        ],
       [0.        , 0.        , 0.0802874 , ..., 0.        , 1.        ,
        0.        ],
       [0.        , 0.        , 0.        , ..., 0.        , 0.        ,
        1.        ]])
```

```
simi_score.shape
```

```
(4760, 4760)
```

```
fav_movie_name = input('enter your favourite movie name: ')
```

```
enter your favourite movie name: toy sto
```

```
all_movies_title_list = df['Movie_Title'].tolist()
```

```
import difflib
```

```
movie_recom = difflib.get_close_matches(fav_movie_name, all_movies_title_list)
print(movie_recom)
```

```
['Toy Story']
```

```
close_match = movie_recom[0]
```

```
close_match_movie = df[df.Movie_Title == close_match]['Movie_ID'].values[0]
close_match_movie
```

```
392
```

```
recommendation_score = list(enumerate(simi_score[close_match_movie]))
print(recommendation_score)
```

```
[(0, 0.02632992086096804), (1, 0.0), (2, 0.0437825417878005), (3, 0.0631353569504574
```

```
sorted_similar_movie = sorted(recommendation_score,key = lambda x:x[1],reverse = True)
print(sorted_similar_movie)
```

```
[(392, 1.0), (391, 0.3711922894614562), (1602, 0.22356378820947234), (3052, 0.184257
```

```
print('top 30 suggested movie: \n')

i = 1
for movie in sorted_similar_movie:
  index = movie[0]
  title_from_index = df[df.index == index]['Movie_Title'].values[0]
  if i<31:
    print(i,'.',title_from_index)
    i+=1
```

```
top 30 suggested movie:

1 . Toy Story 2
2 . Toy Story
3 . Toy Story 3
4 . Cradle Will Rock
```

```
5 . Teacher's Pet
6 . Cars 2
7 . 15 Minutes
8 . Friends with Money
9 . Flight
10 . Being John Malkovich
11 . Grosse Pointe Blank
12 . Swing Vote
13 . Ice Princess
14 . An American Carol
15 . Hoodwinked Too! Hood VS. Evil
16 . Transformers: Age of Extinction
17 . Quest for Camelot
18 . War, Inc.
19 . That Thing You Do!
20 . The Expendables 2
21 . Larry Crowne
22 . In & Out
23 . Arlington Road
24 . Cars
25 . Cast Away
26 . Running Forever
27 . Splash
28 . Dirty Work
29 . Wild Hogs
30 . Child's Play 2
```

✓ 3s completed at 5:24 PM