Week6_Assignment

February 26, 2024

[3]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics.pairwise import cosine_similarity, euclidean_distances
import numpy as np
```

0.1 Load data

[4]:
```
movies_df = pd.read_csv('movies.csv')
movies_df.head(5)
```

[4]:
```
   movieId                               title  \
0        1                    Toy Story (1995)
1        2                      Jumanji (1995)
2        3             Grumpier Old Men (1995)
3        4            Waiting to Exhale (1995)
4        5  Father of the Bride Part II (1995)

                                        genres
0  Adventure|Animation|Children|Comedy|Fantasy
1                   Adventure|Children|Fantasy
2                               Comedy|Romance
3                         Comedy|Drama|Romance
4                                       Comedy
```

[5]:
```
ratings_df = pd.read_csv('ratings.csv')
ratings_df.head(5)
```

[5]:
```
   userId  movieId  rating  timestamp
0       1        1     4.0  964982703
1       1        3     4.0  964981247
2       1        6     4.0  964982224
3       1       47     5.0  964983815
4       1       50     5.0  964982931
```

0.2 User2 Subset

[6]:
```
u2ratings = ratings_df[ratings_df['userId'] == 2]
u2ratings
```

[6]:

	userId	movieId	rating	timestamp
232	2	318	3.0	1445714835
233	2	333	4.0	1445715029
234	2	1704	4.5	1445715228
235	2	3578	4.0	1445714885
236	2	6874	4.0	1445714952
237	2	8798	3.5	1445714960
238	2	46970	4.0	1445715013
239	2	48516	4.0	1445715064
240	2	58559	4.5	1445715141
241	2	60756	5.0	1445714980
242	2	68157	4.5	1445715154
243	2	71535	3.0	1445714974
244	2	74458	4.0	1445714926
245	2	77455	3.0	1445714941
246	2	79132	4.0	1445714841
247	2	80489	4.5	1445715340
248	2	80906	5.0	1445715172
249	2	86345	4.0	1445715166
250	2	89774	5.0	1445715189
251	2	91529	3.5	1445714891
252	2	91658	2.5	1445714938
253	2	99114	3.5	1445714874
254	2	106782	5.0	1445714966
255	2	109487	3.0	1445715145
256	2	112552	4.0	1445714882
257	2	114060	2.0	1445715276
258	2	115713	3.5	1445714854
259	2	122882	5.0	1445715272
260	2	131724	5.0	1445714851

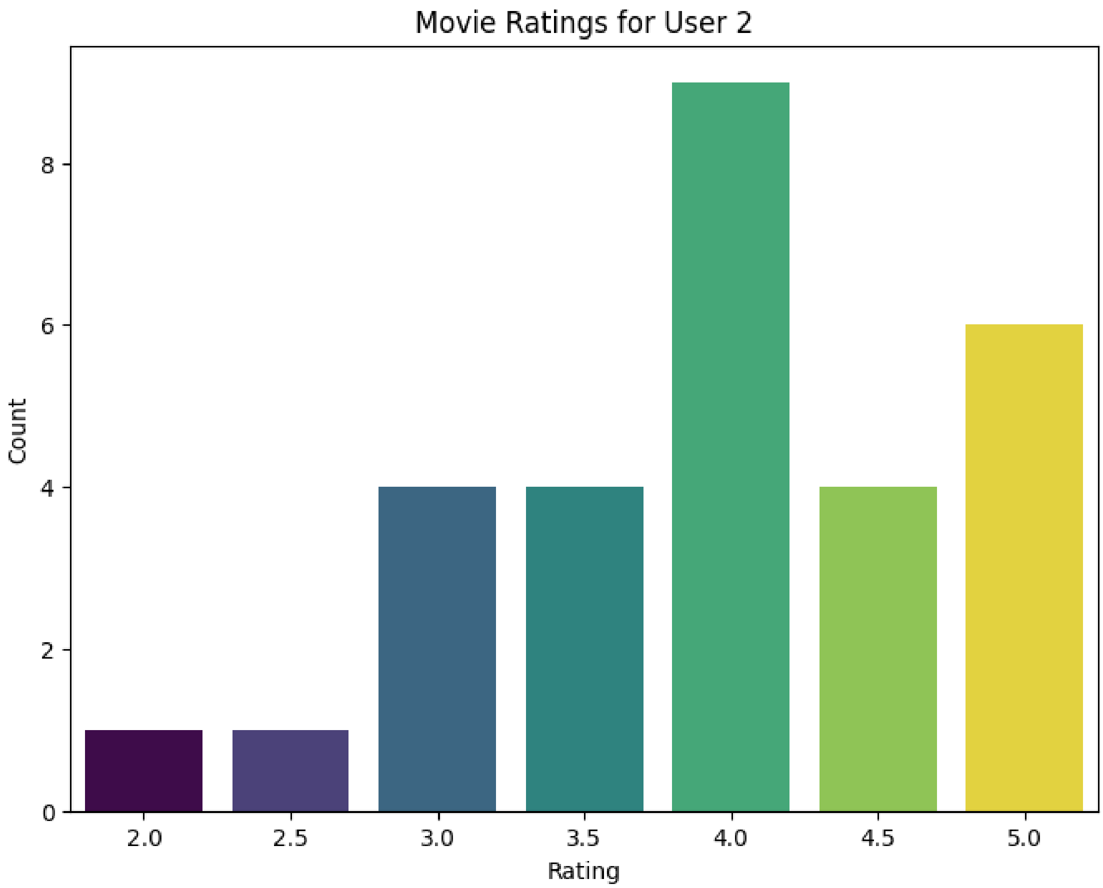
0.3 Number of movies watched by user2

[7]:
```
nm = u2ratings.shape[0]
print("Number of movies watched by user2: ", nm)
```

```
Number of movies watched by user2:  29
```

0.4 User 2 ratings bar chart

[8]:
```
plt.figure(figsize=(8, 6))
sns.countplot(x='rating', data=u2ratings, palette='viridis', hue='rating',␣
↪legend=False)
plt.xlabel('Rating')
plt.ylabel('Count')
plt.title('Movie Ratings for User 2')
plt.show()
```



This bar chart is created using Seaborn's countplot function. It visualizes the distribution of movie ratings for User 2.

X-axis (x='rating') - Represents the different rating values, could be 1 star, 2 stars, 3 stars, etc.

Y-axis (y='Count') - Represents the count of each rating. Each bar's height indicates how many movies User 2 has rated with a particular rating.

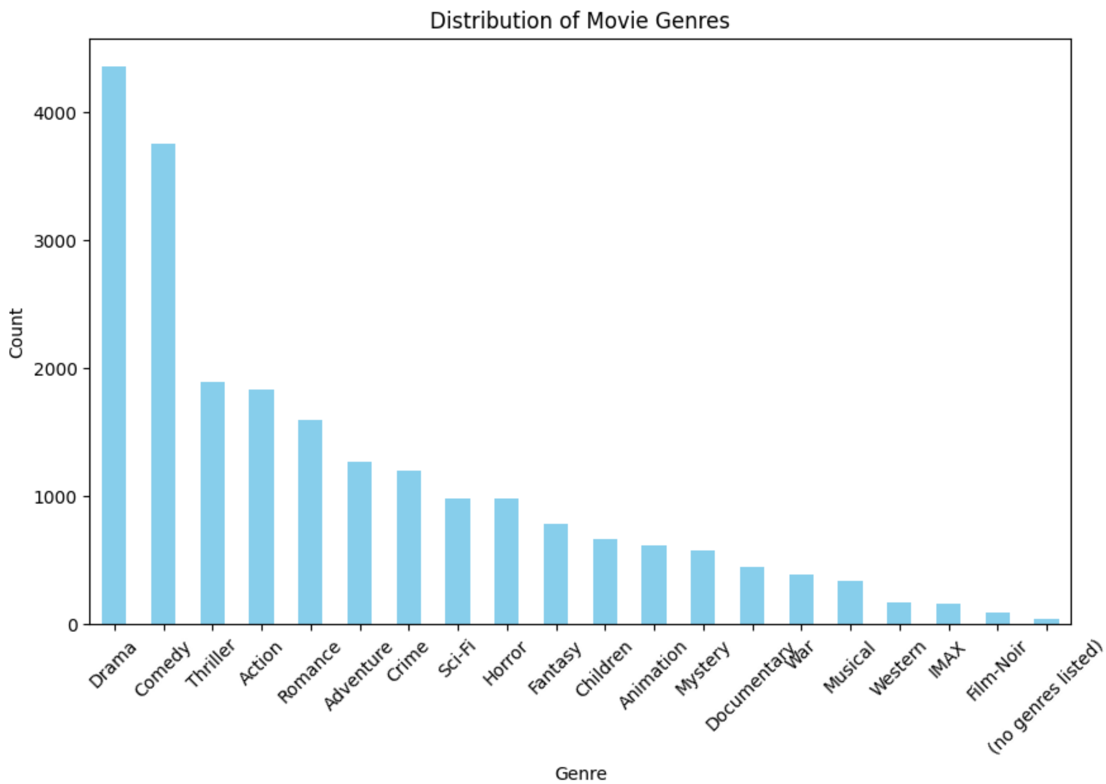
Color Palette (palette='viridis') - Specifies the color palette used for the bars.

Figure Size (plt.figure(figsize=(8, 6))) - Sets the dimensions of the plot, where the width is 8 units and the height is 6 units.

The bars in the chart represent different ratings, and the height of each bar corresponds to the number of movies rated by User 2 with that specific rating.

0.5 Other EDA - Distribution of movie genres

[9]:
```
plt.figure(figsize=(10, 6))
gc = movies_df['genres'].str.split('|', expand=True).stack().value_counts()
gc.plot(kind='bar', color='skyblue')
plt.xlabel('Genre')
plt.ylabel('Count')
plt.title('Distribution of Movie Genres')
plt.xticks(rotation=45)
plt.show()
```



The bar plot visualizes the distribution of movie genres in the dataset, showing the count of movies for each genre. This can help in understanding the diversity and popularity of different genres in the given dataset.

0.6 Top movies for user2

[10]:
```
topmovies = u2ratings[u2ratings['rating'] >= 4.5].merge(movies_df,␣
  ↪on='movieId')[['title', 'rating']].sort_values(by='rating', ascending=False)
print("User 2's top movies:")
print(topmovies)
```

```
User 2's top movies:
                                               title  rating
2                               Step Brothers (2008)     5.0
5                                 Inside Job (2010)     5.0
6                                    Warrior (2011)     5.0
7                     Wolf of Wall Street, The (2013)     5.0
8                          Mad Max: Fury Road (2015)     5.0
9  The Jinx: The Life and Deaths of Robert Durst …     5.0
0                          Good Will Hunting (1997)     4.5
1                           Dark Knight, The (2008)     4.5
3                      Inglourious Basterds (2009)     4.5
4                                  Town, The (2010)     4.5
```

0.7 Most similar user to user 2 using cosine and Euclidean distances

[11]:
```
u2vector = u2ratings.pivot(index='userId', columns='movieId', values='rating').
  ↪fillna(0)
all_users_vector = ratings_df.pivot(index='userId', columns='movieId',␣
  ↪values='rating').fillna(0)

cmn_columns = u2vector.columns.intersection(all_users_vector.columns)

u2vector = u2vector[cmn_columns]
all_users_vector = all_users_vector[cmn_columns]
```

[12]:
```
cos_similarity = cosine_similarity(u2vector, all_users_vector)
euclidean_distances = euclidean_distances(u2vector, all_users_vector)

print(f"Most similar user to User 2 (cosine): {cos_similarity}")
print(f"Most similar user to User 2 (euclidean): {euclidean_distances}")
```

```
Most similar user to User 2 (cosine): [[0.26086186 1.         0.
0.20751434 0.13834289 0.22825413
  0.24495082 0.13834289 0.         0.35144117 0.24455799 0.
  0.18445719 0.13834289 0.49455168 0.39615764 0.43027895 0.77644315
  0.2557961  0.18445719 0.60584711 0.46293651 0.18445719 0.48089623
  0.46106757 0.         0.         0.47295934 0.38972905 0.37121293
  0.         0.13834289 0.24846359 0.23045899 0.         0.23518292
  0.13834289 0.13834289 0.         0.13834289 0.58971767 0.35410056
  0.13834289 0.         0.33206224 0.13834289 0.33328977 0.
  0.26687054 0.61863671 0.26809823 0.55354535 0.         0.13834289
```

0.18445719	0.22134863	0.27207436	0.22825413	0.	0.24846359
0.22325714	0.77977403	0.57293207	0.38879219	0.46973355	0.39010257
0.3432886	0.83476077	0.22825413	0.24455799	0.	0.13834289
0.60259073	0.18445719	0.	0.49600239	0.34992614	0.13834289
0.18445719	0.53635901	0.13834289	0.18445719	0.38615679	0.13834289
0.20751434	0.35609211	0.	0.44953772	0.13834289	0.
0.38974874	0.	0.24846359	0.13834289	0.	0.18445719
0.18445719	0.35138059	0.22825413	0.	0.18445719	0.13834289
0.67971357	0.	0.79538375	0.13834289	0.13834289	0.
0.13834289	0.	0.76533592	0.43129063	0.	0.31974512
0.18445719	0.27300317	0.21353049	0.	0.67217531	0.
0.13834289	0.65273801	0.41003535	0.13834289	0.65606915	0.13834289
0.	0.	0.26086186	0.13834289	0.39079356	0.48317866
0.13834289	0.	0.19695383	0.18445719	0.22825413	0.
0.4663538	0.42161507	0.5518402	0.	0.22685235	0.18445719
0.13834289	0.	0.	0.28065806	0.	0.
0.	0.45828757	0.35244558	0.30617717	0.26086186	0.30128404
0.	0.	0.37678124	0.18445719	0.13834289	0.13834289
0.20751434	0.18445719	0.	0.42931207	0.13834289	0.20751434
0.28173982	0.13834289	0.13834289	0.18445719	0.	0.18445719
0.	0.13834289	0.65706475	0.13834289	0.13834289	0.
0.	0.39741418	0.	0.45234816	0.	0.18445719
0.28258593	0.	0.43437052	0.35738581	0.13834289	0.
0.13834289	0.	0.29929141	0.30977174	0.13834289	0.29346959
0.37298329	0.44711516	0.	0.13834289	0.25616126	0.35254487
0.31748392	0.	0.	0.	0.46647338	0.20751434
0.48502631	0.63712365	0.20751434	0.	0.24164924	0.18445719
0.	0.	0.48983055	0.36719262	0.13834289	0.62811663
0.40734296	0.	0.18445719	0.45550032	0.62822688	0.13834289
0.13834289	0.35200082	0.30296944	0.59471542	0.63839333	0.
0.13834289	0.	0.33581513	0.	0.47763126	0.22934757
0.41118582	0.13834289	0.	0.18445719	0.	0.36879624
0.6977416	0.20751434	0.82339478	0.13834289	0.24455799	0.18445719
0.	0.34659106	0.	0.47411066	0.	0.13834289
0.	0.24347427	0.39053846	0.13834289	0.24455799	0.18445719
0.13834289	0.	0.18445719	0.20751434	0.	0.
0.	0.25180416	0.	0.72351453	0.30617717	0.18445719
0.	0.13834289	0.70376786	0.33144012	0.	0.63624856
0.24455799	0.22825413	0.20751434	0.25689247	0.18445719	0.39286742
0.	0.23766083	0.20751434	0.47597877	0.	0.18445719
0.35673856	0.30617717	0.	0.83414426	0.18445719	0.40307602
0.31948923	0.	0.	0.24455799	0.75578313	0.19484292
0.50788978	0.34423393	0.13834289	0.	0.	0.
0.22934757	0.13834289	0.	0.18445719	0.69171446	0.67102847
0.38267884	0.22557955	0.	0.35497272	0.19266935	0.
0.13834289	0.58082836	0.3000034	0.47779536	0.13834289	0.50488132
0.61016953	0.45602509	0.18445719	0.34959159	0.22622482	0.23766083
0.13834289	0.22622482	0.59191141	0.13834289	0.24486267	0.

```
  0.28511644 0.13834289 0.         0.27546733 0.         0.34416734
  0.         0.         0.53145804 0.7214811  0.22825413 0.39822899
  0.         0.4613716  0.55439421 0.         0.20751434 0.
  0.13834289 0.46840006 0.24846359 0.         0.62555243 0.54018706
  0.24455799 0.18445719 0.18445719 0.22991761 0.         0.13834289
  0.13834289 0.13834289 0.22825413 0.30625026 0.         0.56455746
  0.13834289 0.58622459 0.35954901 0.52071514 0.         0.
  0.13834289 0.22825413 0.43784064 0.         0.         0.20751434
  0.25332902 0.         0.45369696 0.         0.         0.13834289
  0.         0.25824007 0.18445719 0.39789582 0.18445719 0.
  0.         0.13834289 0.31710049 0.         0.20751434 0.57290041
  0.13834289 0.13834289 0.13834289 0.         0.38162752 0.81285723
  0.42815419 0.2300827  0.63797738 0.35392682 0.50152921 0.28804078
  0.13834289 0.13834289 0.13834289 0.65784851 0.42535965 0.13834289
  0.18445719 0.2557961  0.         0.13834289 0.         0.51487694
  0.39686301 0.54698899 0.51016418 0.22134863 0.13834289 0.42877776
  0.13834289 0.         0.42069787 0.         0.26327685 0.13834289
  0.42978741 0.         0.13834289 0.83317362 0.13834289 0.
  0.         0.18445719 0.29117315 0.18445719 0.13834289 0.
  0.18445719 0.         0.36726586 0.44527091 0.43088789 0.41789961
  0.         0.4423443  0.         0.45767134 0.         0.13834289
  0.2253933  0.13834289 0.27716573 0.34585723 0.13834289 0.42172618
  0.44018307 0.13834289 0.57983571 0.         0.         0.46685142
  0.         0.         0.59857647 0.51488916 0.         0.13834289
  0.37460742 0.23766083 0.4443302  0.39400306 0.23057149 0.
  0.18445719 0.18445719 0.81009624 0.20751434 0.41046935 0.22825413
  0.         0.         0.         0.13834289 0.35743337 0.
  0.13834289 0.21911417 0.         0.         0.5018984  0.
  0.43950173 0.13834289 0.13834289 0.61478442 0.43299991 0.
  0.29097047 0.         0.24455799 0.30197878 0.         0.6123471
  0.49674264 0.13834289 0.55185926 0.26086186 0.         0.56799016
  0.         0.         0.         0.23518292 0.18445719 0.52396154
  0.         0.         0.50577508 0.20751434 0.19444977 0.18445719
  0.13834289 0.27268144 0.26086186 0.         0.         0.
  0.         0.33291319 0.35647306 0.47902654 0.42264638 0.20751434
  0.27367005 0.         0.22825413 0.         0.13834289 0.
  0.19695383 0.75058407 0.71108708 0.48757338 0.18445719 0.41130484
  0.13834289 0.13834289 0.51858499 0.         0.         0.35429365
  0.         0.24455799 0.68036964 0.13834289 0.         0.
  0.13834289 0.         0.         0.43018733 0.37715714 0.51372095
  0.26086186 0.         0.43095879 0.48046792 0.         0.22325714
  0.13834289 0.48140573 0.18445719 0.22134863 0.29426159 0.18445719
  0.         0.61198145 0.20751434 0.18445719 0.84138527 0.30503688
  0.59331773 0.13834289 0.27207436 0.         0.         0.40268434
  0.13834289 0.43193285 0.13834289 0.75986137]]
Most similar user to User 2 (euclidean): [[20.98213526  0.         21.68524844
21.5        21.47673159 21.21909517
  21.02974084 21.56965461 21.68524844 20.3039405  21.02974084 21.68524844
```

21.33658829	21.47673159	18.85470763	19.91230775	19.60867155	13.72953022
21.05350327	21.31900561	17.41407477	19.22238279	21.33658829	19.01315334
19.35200248	21.68524844	21.68524844	19.13765921	20.03746491	20.37768387
21.68524844	21.56965461	21.02974084	21.14828598	21.68524844	21.07723891
21.5	21.56965461	21.68524844	21.56965461	17.59971591	20.3039405
21.5	21.68524844	20.47559523	21.5	20.45116134	21.68524844
20.97021698	17.4642492	20.93442142	18.1934054	21.68524844	21.5
21.33658829	21.14828598	20.88659857	21.21909517	21.68524844	21.02974084
21.19551839	13.78404875	17.90949469	20.03122562	19.17028951	20.03746491
20.38381711	11.98957881	21.21909517	21.05350327	21.68524844	21.56965461
17.44276354	21.33658829	21.68524844	18.8348082	20.33469941	21.52905014
21.40677463	18.54049622	21.47673159	21.31900561	20.08730943	21.56965461
21.26617032	20.27929979	21.68524844	19.42935923	21.48255106	21.68524844
20.03122562	21.68524844	21.02974084	21.56965461	21.68524844	21.31314149
21.31314149	20.35313244	21.21909517	21.68524844	21.31314149	21.5
15.99218559	21.68524844	13.27591805	21.56965461	21.56965461	21.68524844
21.56965461	21.68524844	13.96424004	19.67866865	21.68524844	20.60339778
21.33658829	20.86264605	21.21909517	21.68524844	16.14001239	21.68524844
21.56965461	16.75559608	19.81161276	21.52905014	16.37070554	21.56965461
21.68524844	21.68524844	20.94636007	21.56965461	19.96246478	19.00657781
21.5	21.68524844	21.28966886	21.33658829	21.11279233	21.68524844
19.37137063	19.67231557	18.1934054	21.68524844	21.28966886	21.31314149
21.47673159	21.68524844	21.68524844	20.89258242	21.68524844	21.68524844
21.68524844	19.27433527	20.54872259	20.74246851	20.93442142	20.68211788
21.68524844	21.68524844	20.08730943	21.33658829	21.56965461	21.56965461
21.21909517	21.33658829	21.68524844	19.66596044	21.56965461	21.21909517
20.84466359	21.56965461	21.56965461	21.31314149	21.68524844	21.31314149
21.68524844	21.56965461	16.43167673	21.56965461	21.56965461	21.68524844
21.68524844	19.89974874	21.68524844	19.42935923	21.68524844	21.31314149
20.8146583	21.68524844	19.5512148	20.2546291	21.5	21.68524844
21.48255106	21.68524844	20.71834936	20.65187643	21.5	20.74246851
20.1246118	19.44865034	21.68524844	21.5	20.98213526	20.35927307
20.57304061	21.68524844	21.68524844	21.68524844	19.2418814	21.21909517
19.08533468	16.7182535	21.21320344	21.68524844	21.09502311	21.31314149
21.68524844	21.68524844	18.92749323	20.22992832	21.5	16.87453703
19.80530232	21.68524844	21.33658829	19.31320792	16.87453703	21.5
21.56965461	20.40220576	20.67002661	17.5071414	16.71077497	21.68524844
21.5	21.68524844	20.42669822	21.68524844	19.11805429	21.12463017
19.7673974	21.5	21.68524844	21.33658829	21.68524844	20.21756662
15.54027027	21.30727575	12.61942946	21.56965461	21.10094785	21.31900561
21.68524844	20.43893344	21.68524844	19.21587885	21.68524844	21.56965461
21.68524844	21.03568397	20.	21.47673159	21.10094785	21.40677463
21.5	21.68524844	21.33658829	21.26617032	21.68524844	21.68524844
21.68524844	21.00595154	21.68524844	14.98332406	20.64582282	21.33658829
21.68524844	21.56965461	15.4677083	20.46948949	21.68524844	16.73320053
21.10094785	21.12463017	21.21909517	20.96425529	21.36586062	19.94367068
21.68524844	21.10094785	21.21909517	19.07878403	21.68524844	21.52324325
20.35927307	20.74246851	21.68524844	12.51998403	21.33658829	19.88089535

20.5669638	21.68524844	21.68524844	21.02974084	14.49137675	21.27204739
18.68823159	20.47559523	21.5	21.68524844	21.68524844	21.68524844
21.12463017	21.56965461	21.68524844	21.31314149	15.7876534	16.10900369
20.21138293	21.13054661	21.68524844	20.27313493	21.36586062	21.68524844
21.56965461	17.76231967	20.70024154	19.09842925	21.5	18.88782677
17.39971264	19.30025907	21.31314149	20.34084561	21.20141505	21.10094785
21.56965461	21.20141505	17.61391495	21.56965461	21.05350327	21.68524844
20.80264406	21.56965461	21.68524844	20.85665361	21.68524844	20.46338193
21.68524844	21.68524844	18.37117307	15.2561463	21.21909517	19.89974874
21.68524844	19.2418814	18.07622748	21.68524844	21.21320344	21.68524844
21.47673159	19.17680891	21.02974084	21.68524844	17.01469953	18.26198237
21.10094785	21.33658829	21.31900561	21.10687092	21.68524844	21.56965461
21.56965461	21.5	21.21909517	20.64582282	21.68524844	17.958285
21.47673159	17.5855054	20.26079959	18.5876303	21.68524844	21.68524844
21.56965461	21.12463017	19.5128163	21.68524844	21.68524844	21.30727575
20.98213526	21.68524844	19.39716474	21.68524844	21.68524844	21.52905014
21.68524844	20.95829192	21.33658829	19.94993734	21.33658829	21.68524844
21.68524844	21.5	20.5669638	21.68524844	21.21909517	17.79747173
21.5	21.56965461	21.5	21.68524844	20.22374842	12.86468033
19.65960325	21.10687092	16.87453703	20.32855135	18.76832438	20.77859476
21.56965461	21.56965461	21.56965461	16.33248297	19.6468827	21.48255106
21.31314149	20.96425529	21.68524844	21.56965461	21.68524844	18.60779407
19.90602924	18.20714146	18.74833326	21.14828598	21.56965461	19.65960325
21.56965461	21.68524844	19.6977156	21.68524844	20.93442142	21.5
19.58315603	21.68524844	21.56965461	12.04159458	21.56965461	21.68524844
21.68524844	21.31314149	20.79062289	21.31900561	21.56965461	21.68524844
21.33658829	21.68524844	20.21138293	19.45507646	19.57677195	19.73575436
21.68524844	19.49358869	21.68524844	19.28081948	21.68524844	21.47673159
21.14828598	21.5	20.83866598	20.5	21.56965461	19.66596044
19.49358869	21.56965461	17.7552809	21.68524844	21.68524844	19.25486951
21.68524844	21.68524844	17.54280479	18.59435398	21.68524844	21.56965461
20.1246118	21.10094785	19.42935923	19.93113143	21.10687092	21.68524844
21.33658829	21.33658829	12.97112177	21.21320344	19.78004044	21.21909517
21.68524844	21.68524844	21.68524844	21.56965461	20.2546291	21.68524844
21.5	21.20141505	21.68524844	21.68524844	18.78829423	21.68524844
19.65324401	21.56965461	21.56965461	17.1318417	19.75474627	21.68524844
20.79663434	21.68524844	21.10094785	20.71231518	21.68524844	17.48570845
18.9802529	21.56965461	18.09005252	20.93442142	21.68524844	17.88155474
21.68524844	21.68524844	21.68524844	21.07723891	21.31314149	18.4729532
21.68524844	21.68524844	18.734994	21.21320344	21.38340478	21.33658829
21.5	20.87462575	20.93442142	21.68524844	21.68524844	21.68524844
21.68524844	20.45727255	20.3039405	19.08533468	19.65324401	21.21909517
20.86264605	21.68524844	21.21909517	21.68524844	21.52905014	21.68524844
21.28966886	14.36140662	15.26433752	18.94069692	21.31900561	19.7673974
21.5	21.56965461	18.70160421	21.68524844	21.68524844	20.27929979
21.68524844	21.10094785	16.31716887	21.56965461	21.68524844	21.68524844
21.56965461	21.68524844	21.68524844	19.65324401	20.174241	18.70160421
20.98213526	21.68524844	19.58315603	19.07223112	21.68524844	21.19551839

```
21.56965461 19.01315334 21.33658829 21.14828598 20.74849392 21.31900561
21.68524844 17.17556404 21.21909517 21.33658829 11.92686044 20.65187643
17.62101019 21.56965461 20.88659857 21.68524844 21.68524844 19.84943324
21.56965461 19.57677195 21.5        14.23024947]]
```

`Cosine Similarity`

The cosine similarity scores range between 0 and 1, where 1 indicates perfect similarity. Users with higher cosine similarity scores are more similar to User 2 in terms of their preferences or behaviors.

`Euclidean Distance`

The Euclidean distance scores represent dissimilarity, with lower values indicating higher similarity. Users with lower Euclidean distances are more similar to User 2 in terms of their preferences or behaviors. The most similar user has a Euclidean distance of 0 (perfect similarity), and others have distances ranging from approximately 13.73 to 21.69.

0.8 Most similar user

[13]:
```
cos_similarity[:, 1] = -1
euclidean_distances[:, 1] = np.inf
```

[14]:
```
ms_cos_index = np.argmax(cos_similarity)
ms_cos_score = cos_similarity[0, ms_cos_index]

print(f"Most similar user to User 2 (cosine): User {ms_cos_index + 1} with␣
↪similarity score {ms_cos_score}")
```

```
Most similar user to User 2 (cosine): User 599 with similarity score
0.8413852739119481
```

[15]:
```
ms_euclidean_index = np.argmin(euclidean_distances)
ms_euclidean_distance = euclidean_distances[0, ms_euclidean_index]

print(f"Most similar user to User 2 (euclidean): User {ms_euclidean_index + 1}␣
↪with distance {ms_euclidean_distance}")
```

```
Most similar user to User 2 (euclidean): User 599 with distance
11.926860441876563
```

0.9 Movies rated by the most similar user as user2

[16]:
```
most_similar_user_ratings = ratings_df[ratings_df['userId'] == (ms_cos_index +␣
↪1)]

# Merge with movies_df to include movie details
most_similar_user_ratings_with_details = most_similar_user_ratings.
↪merge(movies_df, on='movieId')

print("Movies rated by the most similar user:")
```

```python
print(most_similar_user_ratings_with_details)
```

```
Movies rated by the most similar user:
      userId  movieId  rating   timestamp  \
0        599        1     3.0  1498524204
1        599        2     2.5  1498514085
2        599        3     1.5  1498505071
3        599        6     4.5  1498539623
4        599        7     2.5  1498514161
…         …       …      …           …
2473     599   179817     3.0  1516604716
2474     599   180031     3.5  1518298493
2475     599   180297     3.0  1516604804
2476     599   181315     3.5  1517370374
2477     599   183301     3.0  1519148271

                                   title  \
0                       Toy Story (1995)
1                         Jumanji (1995)
2                Grumpier Old Men (1995)
3                            Heat (1995)
4                         Sabrina (1995)
…                                    …
2473                 Darkest Hour (2017)
2474           The Shape of Water (2017)
2475          The Disaster Artist (2017)
2476               Phantom Thread (2017)
2477  The Tale of the Bunny Picnic (1986)

                                           genres
0     Adventure|Animation|Children|Comedy|Fantasy
1                      Adventure|Children|Fantasy
2                                  Comedy|Romance
3                           Action|Crime|Thriller
4                                  Comedy|Romance
…                                             …
2473                                    Drama|War
2474                      Adventure|Drama|Fantasy
2475                                 Comedy|Drama
2476                                Drama|Romance
2477                                     Children

[2478 rows x 6 columns]
```

0.10 Recommend movies with cosine similarity

[17]:
```
user_2_recommended_movies = all_users_vector.loc[ms_cos_index]
movies_not_rated_by_user_2 =␣
↪user_2_recommended_movies[user_2_recommended_movies == 0].index

# threshold rating
threshold_rating = 3.0

# Extract recommended movies information with ratings above the threshold
recommended_movies_info = movies_df[movies_df['movieId'].
↪isin(movies_not_rated_by_user_2)]
recommended_movies_info = recommended_movies_info[['movieId', 'title']].
↪merge(ratings_df[ratings_df['userId'] == (ms_cos_index + 1)], on='movieId')
recommended_movies_info = recommended_movies_info.rename(columns={'rating':␣
↪'user_rating'})

# Filter movies above threshold
recommended_movies_info =␣
↪recommended_movies_info[recommended_movies_info['user_rating'] >=␣
↪threshold_rating]

print("Movies recommended for User 2 based on similarity with User",␣
↪ms_cos_index + 1, "with ratings above", threshold_rating)
print(recommended_movies_info)
```

```
Movies recommended for User 2 based on similarity with User 599 with ratings
above 3.0
    movieId                              title  userId  user_rating  \
0       318   Shawshank Redemption, The (1994)     599          4.0
2      1704           Good Will Hunting (1997)     599          4.5
3      3578                   Gladiator (2000)     599          3.5
4      6874           Kill Bill: Vol. 1 (2003)     599          5.0
5      8798                  Collateral (2004)     599          3.0
7     48516               Departed, The (2006)     599          3.0
8     58559            Dark Knight, The (2008)     599          3.5
10    68157        Inglourious Basterds (2009)     599          3.5
11    71535                  Zombieland (2009)     599          3.0
12    77455  Exit Through the Gift Shop (2010)     599          3.5
13    80489                   Town, The (2010)     599          3.5
14    91529      Dark Knight Rises, The (2012)     599          3.0
15    99114            Django Unchained (2012)     599          3.5
16   106782     Wolf of Wall Street, The (2013)    599          3.0
17   109487                Interstellar (2014)     599          3.5
18   112552                    Whiplash (2014)     599          3.0
19   115713                  Ex Machina (2015)     599          3.5
20   122882          Mad Max: Fury Road (2015)     599          4.0
```

```
    timestamp
0   1498498867
2   1498762601
3   1498501113
4   1498457174
5   1498523618
7   1498522886
8   1498798185
10  1498500693
11  1498524922
12  1498542480
13  1498542459
14  1498527139
15  1498528776
16  1498528478
17  1498532289
18  1498589282
19  1498528866
20  1498854698
```

This output shows the movies recommended for User 2 based on similarity with User 599, with ratings above 3.0. Each row in the output represents a recommended movie, and the columns provide details about each movie, including the movie ID, title, the user ID who rated the movie (in this case, User 599), the rating given by User 599, and the timestamp of the rating.

To achieve this output, we first calculated the most similar user to User 2 using cosine similarity (User 599). Then, we extracted movies not rated by User 2 but rated by User 599. Next, we filtered out movies with ratings below the threshold of 3.0. Finally, we merged the information about these recommended movies with the ratings given by User 599 to provide additional details such as the movie title and the rating given by User 599.

0.11 Analysis

The method used in this recommendation system is user-based collaborative filtering. This approach recommends items -movies - to a user based on the preferences and behavior of similar users.

The recommendations are based on the ratings of the most similar user to User 2, as determined by cosine similarity. Cosine similarity measures the cosine of the angle between two vectors and is commonly used in recommendation systems to compute the similarity between users or items.

Movie Recommendations

The recommended movies seem to make sense based on the ratings of User 599, who is most similar to User 2. These movies have ratings above the threshold of 3.0 and are likely to be enjoyed by User 2 since they were highly rated by a similar user.

Choice of Similarity Metric

Cosine similarity was chosen as the similarity metric. Cosine similarity is effective when the magnitude of the vectors is important, as it measures the cosine of the angle between them rather than their distance. It's commonly used in recommendation systems because it captures the direction of similarity between users or items regardless of their magnitude.

Justification of Cosine Similarity

Cosine similarity was chosen because it's well-suited for high-dimensional data like user-item rating matrices and is computationally efficient. It also handles sparse data well and is robust to scale differences between users. Additionally, cosine similarity considers the direction of the vectors rather than their magnitude, making it suitable for recommendation systems where the absolute ratings may not be as important as the relative preferences between users.