In [1]:
```python
import pandas as pd
import numpy as np
```

In [2]:
```python
df_movies_genres = pd.read_csv("C:\\Users\\hsahn\\Design of AI systems\\movie_genres.csv")
df_user_reviews = pd.read_csv("C:\\Users\\hsahn\\Design of AI systems\\user_reviews.csv")
```

In [3]:
```python
df_movies_genres.head()
```

Out[3]:

	Unnamed: 0	movie_title	genre_action	genre_adventure	genre_animation	genre_biography	genre_comedy	genre_crime	genre_documentary
0	0	The Net	1	0	0	0	0	1	0
1	1	Happily N'Ever After	0	1	1	0	1	0	0
2	2	Tomorrowland	1	1	0	0	0	0	0
3	3	American Hero	1	0	0	0	1	0	0
4	4	Das Boot	0	1	0	0	0	0	0

5 rows × 27 columns

In [4]:
```python
df_user_reviews.head()
```

Out[4]:

	Unnamed: 0	User	The Net	Happily N'Ever After	Tomorrowland	American Hero	Das Boot	Final Destination 3	Licence to Kill	The Hundred-Foot Journey	...	The Martian	Micmacs	Solomon and Sheba	In the Company of Men
0	0	Vincent	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0
1	1	Edgar	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0
2	2	Addilyn	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0
3	3	Marlee	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0
4	4	Javier	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0

5 rows × 2002 columns

In [5]:
```python
df_movies_genres = df_movies_genres.drop('Unnamed: 0', axis = 1)
```

In [6]:
```python
df_movies_genres
```

Out[6]:

	movie_title	genre_action	genre_adventure	genre_animation	genre_biography	genre_comedy	genre_crime	genre_documentary	genre_
0	The Net	1	0	0	0	0	1	0	
1	Happily N'Ever After	0	1	1	0	1	0	0	
2	Tomorrowland	1	1	0	0	0	0	0	
3	American Hero	1	0	0	0	1	0	0	
4	Das Boot	0	1	0	0	0	0	0	
...	...	...	...	...	...	...	...	...	
1995	Big Fish	0	1	0	0	0	0	0	
1996	Get Real	0	0	0	0	1	0	0	
1997	Trading Places	0	0	0	0	1	0	0	
1998	DOA: Dead or Alive	1	1	0	0	0	0	0	
1999	Hey Arnold! The Movie	0	1	1	0	1	0	0	

2000 rows × 26 columns

In [7]:

```
df_movies_genres.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2000 entries, 0 to 1999
Data columns (total 26 columns):
 #   Column             Non-Null Count  Dtype 
---  ------             --------------  ----- 
 0   movie_title        2000 non-null   object
 1   genre_action       2000 non-null   int64 
 2   genre_adventure    2000 non-null   int64 
 3   genre_animation    2000 non-null   int64 
 4   genre_biography    2000 non-null   int64 
 5   genre_comedy       2000 non-null   int64 
 6   genre_crime        2000 non-null   int64 
 7   genre_documentary  2000 non-null   int64 
 8   genre_drama        2000 non-null   int64 
 9   genre_family       2000 non-null   int64 
 10  genre_fantasy      2000 non-null   int64 
 11  genre_film-noir    2000 non-null   int64 
 12  genre_history      2000 non-null   int64 
 13  genre_horror       2000 non-null   int64 
 14  genre_music        2000 non-null   int64 
 15  genre_musical      2000 non-null   int64 
 16  genre_mystery      2000 non-null   int64 
 17  genre_news         2000 non-null   int64 
 18  genre_reality-tv   2000 non-null   int64 
 19  genre_romance      2000 non-null   int64 
 20  genre_sci-fi       2000 non-null   int64 
 21  genre_short        2000 non-null   int64 
 22  genre_sport        2000 non-null   int64 
 23  genre_thriller     2000 non-null   int64 
 24  genre_war          2000 non-null   int64 
 25  genre_western      2000 non-null   int64 
dtypes: int64(25), object(1)
memory usage: 406.4+ KB
```

In [8]:

```
# df_movies_genres.describe()
```

In [9]:

```
df_user_reviews = df_user_reviews.drop('Unnamed: 0', axis = 1)
```

In [10]:

```
df_user_reviews
```

Out[10]:

	User	The Net	Happily N'Ever After	Tomorrowland	American Hero	Das Boot	Final Destination 3	Licence to Kill	The Hundred-Foot Journey	The Matrix	...	The Martian	Micmacs	Solomon and Sheba	In the Company of Men
0	Vincent	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0
1	Edgar	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0
2	Addilyn	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0
3	Marlee	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0
4	Javier	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
595	Mariana	5.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0
596	Ivy	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0
597	Kevin	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0
598	Nora	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0
599	Sarai	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0

600 rows × 2001 columns

In [11]:

```
df_user_reviews.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 600 entries, 0 to 599
Columns: 2001 entries, User to Hey Arnold! The Movie
dtypes: float64(2000), object(1)
memory usage: 9.2+ MB
```

In [12]:
```
# df_user_reviews.describe()
```

In [13]:
```
df_reviews = df_user_reviews.drop(['User'],axis=1)
```

In [14]:
```
df_reviews.head()
```

Out[14]:

	The Net	Happily N'Ever After	Tomorrowland	American Hero	Das Boot	Final Destination 3	Licence to Kill	The Hundred-Foot Journey	The Matrix	Creature	...	The Martian	Micmacs	Solomon and Sheba	In the Company of Men
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0

5 rows × 2000 columns

In [15]:
```
user_movie_reviews_matrix = df_reviews.to_numpy()
```

In [16]:
```
print(user_movie_reviews_matrix.shape)
```

```
(600, 2000)
```

In [17]:
```
df_genres = df_movies_genres.drop(['movie_title'],axis=1)
```

In [18]:
```
df_genres.head()
```

Out[18]:

	genre_action	genre_adventure	genre_animation	genre_biography	genre_comedy	genre_crime	genre_documentary	genre_drama	genre_fam
0	1	0	0	0	0	1	0	1	
1	0	1	1	0	1	0	0	0	
2	1	1	0	0	0	0	0	0	
3	1	0	0	0	1	0	0	1	
4	0	1	0	0	0	0	0	1	

5 rows × 25 columns

In [19]:
```
movies_genres_matrix = df_genres.to_numpy()
```

In [20]:
```
print(movies_genres_matrix.shape)
```

```
(2000, 25)
```

In [21]:
```
from sklearn.decomposition import TruncatedSVD
from sklearn import preprocessing as pp
```

In [22]:
```
epsilon =1e-9
n_latent_factors =60

# Generate user-preferences matrix
user_svd = TruncatedSVD(n_components = n_latent_factors)
user_preferences_matrix = user_svd.fit_transform(user_movie_reviews_matrix)+ epsilon
print("Calculated preferences matrix size: \n Number of rows:"+str(user_preferences_matrix.shape[0]))
print(" Number of columns:"+str(user_preferences_matrix.shape[1]))
```

```
Calculated preferences matrix size:
```

```
Number of rows:600
Number of columns:60
```

In [23]:

```
# Generate movie-features matrix
movies_features_matrix = user_svd.fit_transform(np.transpose(user_movie_reviews_matrix))+ epsilon
print("\n\nCalculated features matrix size: \nNumber ofrows: "+str(movies_features_matrix.shape[0]))
print("Number of columns: "+str(movies_features_matrix.shape[1]))

movies_features_matrix = np.transpose(movies_features_matrix)
print("\nCalculated transposed features matrix size: \nNumber of rows: "+str(movies_features_matrix.shape[0]))
print(" Number of columns: "+str(movies_features_matrix.shape[1]))
```

```
Calculated features matrix size:
Number ofrows: 2000
Number of columns: 60

Calculated transposed features matrix size:
Number of rows: 60
 Number of columns: 2000
```

In [24]:

```
# Defining a predicted rating user-movies matrix

predicted_rating_user_movies = np.matmul(user_preferences_matrix, movies_features_matrix)
print(predicted_rating_user_movies.shape)
```

```
(600, 2000)
```

In [25]:

```
# Creating a DataFrame for the predicted ratings matrix
df_predicted_ratings_transposed = pd.DataFrame(data = np.transpose(predicted_rating_user_movies), columns = df_us
df_predicted_ratings_transposed.insert(0,"Movie",df_movies_genres['movie_title'],True)
```

In [26]:

```
df_predicted_ratings_transposed
```

Out[26]:

User	Movie	Vincent	Edgar	Addilyn	Marlee	Javier	Marcus	Mary	Rosalie	Giovanni	...	Piper	Tatum
0	The Net	40.025947	20.213570	6.433343	-1.184818	11.847483	4.041149	3.249017	-20.774679	1.427107	...	7.125299	2.368483
1	Happily N'Ever After	16.657554	14.431209	-1.269168	-9.210225	6.851340	-3.269541	0.714389	12.293941	5.094824	...	5.515058	-3.396427
2	Tomorrowland	19.280057	6.359654	-1.219049	14.819289	3.406498	2.034879	5.122319	4.338521	-2.584919	...	3.902584	-0.985276
3	American Hero	0.099764	0.565712	13.899108	10.323125	10.941424	4.917005	6.519980	-15.419562	1.633899	...	1.473725	5.526886
4	Das Boot	-9.390540	6.336434	-19.220875	12.232732	2.258776	2.945869	1.992850	16.289123	18.252436	...	3.861262	0.724036
...	...	...	...	...	...	...	...	...	...	...	...	...	...
1995	Big Fish	16.063815	-0.652903	-2.021362	-5.504993	7.947084	3.762757	3.501988	6.246595	10.676819	...	3.113555	4.685914
1996	Get Real	-0.469248	7.897512	-2.039754	4.023073	3.823180	1.445964	0.346362	-1.255287	-3.766957	...	2.613299	0.100680
1997	Trading Places	2.643561	3.287785	2.860893	2.694172	1.105632	-0.189350	2.459871	-1.031778	0.370192	...	-2.961052	0.272246
1998	DOA: Dead or Alive	0.364401	-8.006250	-2.241899	1.016075	-1.013620	3.662127	0.894472	7.156675	2.070980	...	7.535351	-2.283420
1999	Hey Arnold! The Movie	-5.496613	-5.774971	3.755295	-2.553848	6.303035	7.566863	1.391986	6.308091	7.153386	...	7.617804	-0.629350

2000 rows × 601 columns

In [27]:

```
df_predicted_ratings_transposed.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2000 entries, 0 to 1999
Columns: 601 entries, Movie to Sarai
dtypes: float64(600), object(1)
memory usage: 9.2+ MB
```

In [28]:

```
# pd.set_option('display.max_columns', None)
```

```
# pd.set_option('display.max_rows', None)
```

Predictions for Vincent

In [29]:

```
df_predicted_ratings_transposed = df_predicted_ratings_transposed.sort_values(by = ['Vincent'], ascending = [Fals
```

In [30]:

```
df_predicted_ratings_transposed
```

Out[30]:

User	Movie	Vincent	Edgar	Addilyn	Marlee	Javier	Marcus	Mary	Rosalie	Giovanni	...	Piper	Tatu
1152	The Good Thief	88.211599	11.908133	34.517762	9.623634	11.509013	5.875535	2.176277	11.100027	-6.162804	...	4.396194	16.4993
998	Maximum Risk	66.192104	23.805301	17.960942	2.593969	-2.361729	8.161664	8.771724	-10.594854	-1.906108	...	-1.539121	9.6887
236	Seeking a Friend for the End of the World	65.086042	7.559983	11.007957	13.080320	2.634399	2.863187	12.362403	1.727687	0.666249	...	-11.927133	0.6985
330	Much Ado About Nothing	63.172636	9.678998	15.135156	22.318075	13.050241	1.369987	9.611998	9.502457	5.496020	...	0.670805	3.9777
1717	The Longest Ride	56.615262	-16.842782	-19.723903	2.118985	8.969191	0.728445	6.492308	8.143902	19.421277	...	-1.609083	2.9069
...	...	...	...	...	...	...	...	...	...	...	...	...	
1454	Theresa Is a Mother	-33.754247	38.402545	22.659887	7.950540	4.303658	4.006535	5.232447	7.518906	8.473907	...	4.987931	8.1304
1097	Flyboys	-34.655883	8.006143	0.667566	1.189215	24.292088	7.276421	6.565342	10.486078	19.162952	...	12.163400	3.2510
1894	ATL	-34.789057	22.723915	44.893895	22.030965	6.008132	7.043894	-6.920832	21.885799	10.001234	...	8.734572	-12.4661
900	The Adventures of Elmo in Grouchland	-37.565296	-18.205279	19.228832	14.871042	1.076135	2.211882	1.033579	-2.074409	-4.939155	...	2.549439	-3.6299
85	Me You and Five Bucks	-51.802303	13.685014	12.336508	4.673090	12.752234	5.848900	6.474717	-3.840239	0.831167	...	7.015140	3.8566

2000 rows × 601 columns

Predictions for Edgar

In [31]:

```
df_predicted_ratings_transposed = df_predicted_ratings_transposed.sort_values(by = ['Edgar'], ascending = [False]
```

In [32]:

```
df_predicted_ratings_transposed
```

Out[32]:

User	Movie	Vincent	Edgar	Addilyn	Marlee	Javier	Marcus	Mary	Rosalie	Giovanni	...	Piper	Ta
1812	Lars and the Real Girl	-7.698890	45.864747	14.341037	0.414661	3.078123	5.495452	-2.718180	1.503533	14.153455	...	0.543453	-12.791
1454	Theresa Is a Mother	-33.754247	38.402545	22.659887	7.950540	4.303658	4.006535	5.232447	7.518906	8.473907	...	4.987931	8.130
669	500 Days of Summer	28.162234	36.442004	1.333752	-5.913937	8.930365	5.603714	6.304354	20.368425	9.135261	...	17.963776	13.907
1194	Drop Dead Gorgeous	-24.030721	34.464130	22.802065	4.599327	4.675005	4.573010	4.925485	-9.433380	-9.341482	...	-0.886166	1.768
1542	Wild Things	-3.105704	33.013388	-29.976951	-8.401328	-1.975028	5.208159	-4.180030	9.576653	1.259552	...	-1.745085	-6.081
...	...	...	...	...	...	...	...	...	...	...	...	...	
1980	Mrs Henderson Presents	22.563650	-23.830431	7.590607	19.451964	1.072814	1.950632	5.528323	-3.376242	-0.402748	...	1.449577	0.401
1432	The Best Exotic Marigold Hotel	20.586130	-25.762550	26.410914	28.094529	-4.142744	2.750570	15.343701	-0.318251	21.760175	...	5.991432	8.550
1236	Election	-1.235109	-27.012663	2.690063	7.647084	1.017874	1.464979	0.854975	3.784757	6.223271	...	0.931578	5.256
1694	Adventureland	-7.223958	-32.661174	32.659379	33.822747	1.530436	5.558817	2.488765	24.868207	-7.487804	...	3.471969	2.037
1088	Dysfunctional	-21.911179	-32.704348	48.969347	5.439767	4.793710	12.557240	1.807911	-3.322193	21.385147	...	7.151401	-12.316

Friends

2000 rows × 601 columns

Predictions for Addilyn

In [33]:
```
df_predicted_ratings_transposed = df_predicted_ratings_transposed.sort_values(by = ['Addilyn'], ascending = [Fals
```

In [34]:
```
df_predicted_ratings_transposed
```

Out[34]:

User	Movie	Vincent	Edgar	Addilyn	Marlee	Javier	Marcus	Mary	Rosalie	Giovanni	...	Piper	Tat
1011	Perrier's Bounty	11.126842	7.311811	62.068836	45.519560	5.388472	-0.183642	3.405082	38.400844	16.221819	...	-3.571746	-9.254
531	Igby Goes Down	-13.336688	5.937199	50.483889	-0.313320	6.917036	6.073350	10.439721	8.938729	4.999677	...	-0.174297	4.706
1088	Dysfunctional Friends	-21.911179	-32.704348	48.969347	5.439767	4.793710	12.557240	1.807911	-3.322193	21.385147	...	7.151401	-12.316
260	9	10.210398	-15.145165	47.506042	19.839395	-2.075837	7.359908	2.377434	28.780532	-4.296487	...	18.698933	19.716
1894	ATL	-34.789057	22.723915	44.893895	22.030965	6.008132	7.043894	-6.920832	21.885799	10.001234	...	8.734572	-12.466
...	...	...	...	...	...	...	...	...	...	...	...	...	
1196	Witness	12.666926	-4.139098	-31.144399	2.888026	11.118038	0.474637	0.941553	5.817365	16.653652	...	4.231884	0.302
906	Nicholas Nickleby	-27.593797	4.601114	-31.669415	-2.762903	-0.050177	3.577785	6.183976	15.541223	20.616716	...	10.032698	10.054
630	Chill Factor	48.262631	18.839505	-32.512322	7.516827	11.109922	3.474476	2.759099	-4.660102	8.891461	...	2.513013	8.156
174	Travelers and Magicians	4.138958	9.978952	-36.744796	-2.606986	-0.107878	2.603020	6.148495	11.790512	5.410577	...	2.288824	7.660
1064	Munich	-14.616556	5.533201	-41.388407	-0.941481	-1.500404	-0.034148	1.203699	2.291204	1.820862	...	-2.614988	0.530

2000 rows × 601 columns

Predictions for Marlee

In [35]:
```
df_predicted_ratings_transposed = df_predicted_ratings_transposed.sort_values(by = ['Marlee'], ascending = [False
```

In [36]:
```
df_predicted_ratings_transposed
```

Out[36]:

User	Movie	Vincent	Edgar	Addilyn	Marlee	Javier	Marcus	Mary	Rosalie	Giovanni	...	Piper	Tat
1011	Perrier's Bounty	11.126842	7.311811	62.068836	45.519560	5.388472	-0.183642	3.405082	38.400844	16.221819	...	-3.571746	-9.254
1613	Zero Effect	12.596832	21.289138	7.410321	35.668306	10.285115	5.489499	-0.242420	-1.807571	4.317554	...	-4.703345	-7.983
1694	Adventureland	-7.223958	-32.661174	32.659379	33.822747	1.530436	5.558817	2.488765	24.868207	-7.487804	...	3.471969	2.037
1916	Dear John	40.672042	-19.322420	3.457344	29.473072	0.191969	0.289261	6.218792	1.901697	15.934935	...	8.386134	8.348
709	Interstellar	-0.580475	24.625557	17.226100	28.359578	-0.795702	10.366033	5.318886	4.509847	10.432297	...	-1.444384	4.190
...	...	...	...	...	...	...	...	...	...	...	...	...	
258	Yes	-10.129188	4.641759	-4.921987	-17.088900	8.839980	-0.514103	5.035454	22.589809	1.301784	...	7.312132	2.641
1302	28 Days Later...	12.329021	1.127112	-2.562601	-17.938608	-0.796590	2.539203	6.815771	-0.204672	4.395216	...	2.228970	6.096
439	I Am Legend	-14.053111	-2.270626	11.410074	-18.833739	0.849657	7.537474	5.532393	-10.855944	8.510744	...	0.021403	-2.293
554	The Glass House	24.288539	21.155030	-7.334714	-20.584111	-0.000860	2.881643	1.516308	-1.647002	-0.462067	...	-0.424695	-1.411
524	Hard to Be a God	21.895897	-5.717386	5.294962	-25.670528	5.350300	0.997910	15.463998	-6.177409	11.430341	...	-0.073118	9.114

2000 rows × 601 columns

Predictions for Javier

Predictions for Javier

In [37]:
```
df_predicted_ratings_transposed = df_predicted_ratings_transposed.sort_values(by = ['Javier'], ascending = [False
```

In [38]:
```
df_predicted_ratings_transposed
```

Out[38]:

User	Movie	Vincent	Edgar	Addilyn	Marlee	Javier	Marcus	Mary	Rosalie	Giovanni	...	Piper	Ta
116	Now You See Me 2	35.579890	11.439955	44.337113	-11.966714	30.180310	9.138184	3.521516	-23.921751	-8.618598	...	-3.897557	7.62
1821	Homefront	-15.285193	13.819875	-1.640086	-16.295338	27.819034	-3.678753	-1.198782	7.016834	3.335706	...	10.440197	7.42
1097	Flyboys	-34.655883	8.006143	0.667566	1.189215	24.292088	7.276421	6.565342	10.486078	19.162952	...	12.163400	3.25
623	Sonny with a Chance	3.364106	26.607714	2.690162	10.015549	20.650167	-1.885893	1.057009	10.107105	13.241942	...	7.630997	6.41
271	Bran Nue Dae	20.902673	0.193784	12.033087	8.205228	19.443464	0.178272	4.932976	-19.421265	5.608477	...	1.393711	6.12
...	...	...	...	...	...	...	...	...	...	...	...	...	
1093	Black Hawk Down	13.621866	4.122097	15.515581	17.675400	-9.054545	5.909087	6.202958	13.337591	7.112159	...	1.420174	12.54
228	Little Miss Sunshine	-6.324866	-20.712909	5.834867	3.510988	-9.366164	2.203066	2.616416	2.051124	1.797260	...	0.832406	5.18
84	The Stewardesses	1.934006	-3.112297	-1.132853	10.596504	-9.640044	1.183614	-1.904850	7.189367	4.909263	...	9.568754	-1.14
262	Vampire Killers	-11.465996	-9.772650	16.776944	9.386893	-9.889289	4.311788	5.617011	14.851140	9.803218	...	-7.657621	7.90
717	Time Bandits	7.448104	-1.486973	-15.497349	-9.037279	-12.415851	0.756819	4.553289	5.134548	-6.598058	...	6.674508	12.18

2000 rows × 601 columns

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: