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

In [2]:
```
movies_df = pd.read_csv('movies.csv')
```

In [3]:
```
movies_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9742 entries, 0 to 9741
Data columns (total 3 columns):
 #   Column   Non-Null Count  Dtype 
---  ------   --------------  ----- 
 0   movieId  9742 non-null   int64 
 1   title    9742 non-null   object
 2   genres   9742 non-null   object
dtypes: int64(1), object(2)
memory usage: 228.5+ KB
```

In [4]:
```
action_df = movies_df[movies_df['genres'] == 'Action'].reset_index()#selecting only acti
```

In [5]:
```
action_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 60 entries, 0 to 59
Data columns (total 4 columns):
 #   Column   Non-Null Count  Dtype 
---  ------   --------------  ----- 
 0   index    60 non-null     int64 
 1   movieId  60 non-null     int64 
 2   title    60 non-null     object
 3   genres   60 non-null     object
dtypes: int64(2), object(2)
memory usage: 2.0+ KB
```

In [6]:
```
action_df.head(5)
```

Out[6]:

	index	movieId	title	genres
0	8	9	Sudden Death (1995)	Action
1	63	71	Fair Game (1995)	Action
2	172	204	Under Siege 2: Dark Territory (1995)	Action
3	215	251	Hunted, The (1995)	Action
4	555	667	Bloodsport 2 (a.k.a. Bloodsport II: The Next K...	Action

In [7]:
```
action_df.drop('genres', axis= 1, inplace= True)
```

In [8]:
```
action_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 60 entries, 0 to 59
Data columns (total 3 columns):
 #   Column   Non-Null Count  Dtype 
---  ------   --------------  ----- 
 0   index    60 non-null     int64 
 1   movieId  60 non-null     int64 
 2   title    60 non-null     object
dtypes: int64(2), object(1)
memory usage: 1.5+ KB
```

In [9]:
```
ratings_df = pd.read_csv('ratings.csv')
ratings_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100836 entries, 0 to 100835
Data columns (total 4 columns):
 #   Column     Non-Null Count   Dtype  
---  ------     --------------   -----  
 0   userId     100836 non-null  int64  
 1   movieId    100836 non-null  int64  
 2   rating     100836 non-null  float64
 3   timestamp  100836 non-null  int64  
dtypes: float64(1), int64(3)
memory usage: 3.1 MB
```

In [10]:
```
ratings_df.drop('timestamp', axis= 1, inplace= True)
```

In [11]:
```
movie_df = ratings_df.pivot(index= 'movieId', columns= 'userId',
                                values= 'rating').reset_index(drop= True)
```

In [12]:
```
movie_df.fillna(0, inplace= True)
```

In [13]:
```
movie_df.index = ratings_df.movieId.unique()
```

In [14]:
```
movie_df.iloc[0:5, 0:5]
```

Out[14]:

userId	1	2	3	4	5
1	4.0	0.0	0.0	0.0	4.0
3	0.0	0.0	0.0	0.0	0.0
6	4.0	0.0	0.0	0.0	0.0
47	0.0	0.0	0.0	0.0	0.0
50	0.0	0.0	0.0	0.0	0.0

In [15]:
```
movie_df.fillna(0, inplace= True)
```

In [16]:
```
from sklearn.metrics import pairwise_distances
from scipy.spatial.distance import cosine, correlation
```

In [17]:
```
movie_sim = 1 - pairwise_distances(movie_df.values, metric= 'correlation')
```

In [18]:
```
movie_sim_df = pd.DataFrame(movie_sim)
```

In [19]:
```
movie_sim_df.iloc[0:5, 0:5]
```

Out[19]:

	0	1	2	3	4
0	1.000000	0.231327	0.173213	-0.028917	0.192474
1	0.231327	1.000000	0.191945	0.071269	0.200526
2	0.173213	0.191945	1.000000	0.067143	0.370171
3	-0.028917	0.071269	0.067143	1.000000	0.167910
4	0.192474	0.200526	0.370171	0.167910	1.000000

In [20]:
```
def get_similar_movies(movieid, topn = 5):
    movieidx = action_df[action_df.movieId == movieid].index[0]
```

```
    action_df['similarity'] = movie_sim_df.iloc[movieidx]
    top_n = action_df.sort_values(['similarity'], ascending = False)[0:topn]
    return top_n
```

In [21]:
```
get_similar_movies(71, topn= 5)
```

Out[21]:

	index	movieId	title	similarity
1	63	71	Fair Game (1995)	1.000000
18	3345	4531	Red Heat (1988)	0.412808
44	7208	72874	Peacekeeper, The (1997)	0.333349
32	4379	6417	Live Wire (1992)	0.332326
9	1910	2534	Avalanche (1978)	0.285086

In [22]:
```
animation_children_df = movies_df[(movies_df['genres'] == 'Animation') |
                                  (movies_df['genres'] == 'Children')]
animation_children_df.head(5)
```

Out[22]:

	movieId	title	genres
301	343	Baby-Sitters Club, The (1995)	Children
1097	1426	Zeus and Roxanne (1997)	Children
6973	66335	Afro Samurai: Resurrection (2009)	Animation
7059	69469	Garfield's Pet Force (2009)	Animation
7195	72603	Merry Madagascar (2009)	Animation

In [23]:
```
def get_similar_movies(movieid, topn= 5):
    movieidx = animation_children_df[animation_children_df.movieId == movieid].index[0]
    animation_children_df['similarity'] = movie_sim_df.iloc[movieidx]
    top_n = animation_children_df.sort_values(['similarity'], ascending= False)[0:topn]
    return top_n
```

In [24]:
```
get_similar_movies(66335)
```

```
/var/folders/3q/2bh9znv97bl09382p2y_xj9m0000gn/T/ipykernel_3544/2739994426.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  animation_children_df['similarity'] = movie_sim_df.iloc[movieidx]
```

Out[24]:

	movieId	title	genres	similarity
6973	66335	Afro Samurai: Resurrection (2009)	Animation	1.000000
8593	117545	Asterix: The Land of the Gods (Astérix: Le dom...	Animation	1.000000
9539	172587	Vacations in Prostokvashino (1980)	Animation	0.490280
9601	176051	LEGO DC Super Hero Girls: Brain Drain (2017)	Animation	0.418148
7279	74791	Town Called Panic, A (Panique au village) (2009)	Animation	0.223780

In [25]:
```
action_df_merged = ratings_df.merge(action_df, on= 'movieId', how= 'inner')
action_df_merged.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
Int64Index: 186 entries, 0 to 185
Data columns (total 6 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   userId      186 non-null    int64  
 1   movieId     186 non-null    int64  
 2   rating      186 non-null    float64
 3   index       186 non-null    int64  
 4   title       186 non-null    object 
 5   similarity  186 non-null    float64
dtypes: float64(2), int64(3), object(1)
memory usage: 10.2+ KB
```

In [24]:
```
#animation_children_df = ratings_df.merge(animation_df, on= 'movieId', how= 'inner')
#animation_df.head(5)
```

Out[24]:

	userId	movieId	rating	title	genres
0	50	66335	2.5	Afro Samurai: Resurrection (2009)	Animation
1	50	81018	3.0	Illusionist, The (L'illusionniste) (2010)	Animation
2	298	81018	2.0	Illusionist, The (L'illusionniste) (2010)	Animation
3	318	81018	4.0	Illusionist, The (L'illusionniste) (2010)	Animation
4	89	69469	5.0	Garfield's Pet Force (2009)	Animation