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

from sklearn.feature_extraction.text import CountVectorizer
from sklearn.metrics.pairwise import cosine_similarity
```

In [2]:
```
credits_df = pd.read_csv("tmdb_5000_credits.csv")
movies_df = pd.read_csv("tmdb_5000_movies.csv")
```

In [3]:
```
credits_df.head()
```

Out[3]:

	movie_id	title	cast	crew
0	19995	Avatar	[{"cast_id": 242, "character": "Jake Sully", "...	[{"credit_id": "52fe48009251416c750aca23", "de...
1	285	Pirates of the Caribbean: At World's End	[{"cast_id": 4, "character": "Captain Jack Spa...	[{"credit_id": "52fe4232c3a36847f800b579", "de...
2	206647	Spectre	[{"cast_id": 1, "character": "James Bond", "cr...	[{"credit_id": "54805967c3a36829b5002c41", "de...
3	49026	The Dark Knight Rises	[{"cast_id": 2, "character": "Bruce Wayne / Ba...	[{"credit_id": "52fe4781c3a36847f81398c3", "de...
4	49529	John Carter	[{"cast_id": 5, "character": "John Carter", "c...	[{"credit_id": "52fe479ac3a36847f813eaa3", "de...

In [4]:
```
movies_df.head()
```

Out[4]:

	budget	genres	homepage	id	keywords	original_
0	237000000	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	http://www.avatarmovie.com/	19995	[{"id": 1463, "name": "culture clash"}, {"id":...	
1	300000000	[{"id": 12, "name": "Adventure"}, {"id": 14, "...	http://disney.go.com/disneypictures/pirates/	285	[{"id": 270, "name": "ocean"}, {"id": 726, "na...	
2	245000000	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	http://www.sonypictures.com/movies/spectre/	206647	[{"id": 470, "name": "spy"}, {"id": 818, "name...	
3	250000000	[{"id": 28, "name": "Action"}, {"id": 80, "nam...	http://www.thedarkknightrises.com/	49026	[{"id": 849, "name": "dc comics"}, {"id": 853,...	
4	260000000	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	http://movies.disney.com/john-carter	49529	[{"id": 818, "name": "based on novel"}, {"id":...	

In [5]:
```
# extract only ID, TITLE, CAST, CREW ,and merge with ID

credits_df.columns = ['id','title','cast','crew']
movies_df= movies_df.merge(credits_df,on = 'id')
movies_df.head()
```

Out[5]:

	budget	genres	homepage	id	keywords	original_
0	237000000	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	http://www.avatarmovie.com/	19995	[{"id": 1463, "name": "culture clash"}, {"id":...	
1	300000000	[{"id": 12, "name": "Adventure"}, {"id": 14, "...	http://disney.go.com/disneypictures/pirates/	285	[{"id": 270, "name": "ocean"}, {"id": 726, "na...	
2	245000000	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	http://www.sonypictures.com/movies/spectre/	206647	[{"id": 470, "name": "spy"}, {"id": 818, "name...	
3	250000000	[{"id": 28, "name": "Action"}, {"id": 80, "nam...	http://www.thedarkknightrises.com/	49026	[{"id": 849, "name": "dc comics"}, {"id": 853,...	
4	260000000	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	http://movies.disney.com/john-carter	49529	[{"id": 818, "name": "based on novel"}, {"id":...	

5 rows × 23 columns

In [6]:
```
movies_df.columns
```

Out[6]:
```
Index(['budget', 'genres', 'homepage', 'id', 'keywords', 'original_language',
       'original_title', 'overview', 'popularity', 'production_companies',
       'production_countries', 'release_date', 'revenue', 'runtime',
       'spoken_languages', 'status', 'tagline', 'title_x', 'vote_average',
       'vote_count', 'title_y', 'cast', 'crew'],
      dtype='object')
```

In [7]:
```
features = ['cast','crew','keywords','genres']

for feature in features:
    movies_df[feature] = movies_df[feature].apply(literal_eval)

movies_df[features].head()
```

Out[7]:

	cast	crew	keywords	genres
0	[{'cast_id': 242, 'character': 'Jake Sully', '...	[{'credit_id': '52fe48009251416c750aca23', 'de...	[{'id': 1463, 'name': 'culture clash'}, {'id':...	[{'id': 28, 'name': 'Action'}, {'id': 12, 'nam...
1	[{'cast_id': 4, 'character': 'Captain Jack Spa...	[{'credit_id': '52fe4232c3a36847f800b579', 'de...	[{'id': 270, 'name': 'ocean'}, {'id': 726, 'na...	[{'id': 12, 'name': 'Adventure'}, {'id': 14, '...
2	[{'cast_id': 1, 'character': 'James Bond', 'cr...	[{'credit_id': '54805967c3a36829b5002c41', 'de...	[{'id': 470, 'name': 'spy'}, {'id': 818, 'name...	[{'id': 28, 'name': 'Action'}, {'id': 12, 'nam...
3	[{'cast_id': 2, 'character': 'Bruce Wayne / Ba...	[{'credit_id': '52fe4781c3a36847f81398c3', 'de...	[{'id': 849, 'name': 'dc comics'}, {'id': 853,...	[{'id': 28, 'name': 'Action'}, {'id': 80, 'nam...
4	[{'cast_id': 5, 'character': 'John Carter', 'c...	[{'credit_id': '52fe479ac3a36847f813eaa3', 'de...	[{'id': 818, 'name': 'based on novel'}, {'id':...	[{'id': 28, 'name': 'Action'}, {'id': 12, 'nam...

In [8]:
```
movies_df['crew'][0]
```

Out[8]:
```
[{'credit_id': '52fe48009251416c750aca23',
  'department': 'Editing',
  'gender': 0,
  'id': 1721,
  'job': 'Editor',
  'name': 'Stephen E. Rivkin'},
 {'credit_id': '539c47ecc3a36810e3001f87',
  'department': 'Art',
  'gender': 2,
  'id': 496,
  'job': 'Production Design',
  'name': 'Rick Carter'},
 {'credit_id': '54491c89c3a3680fb4001cf7',
  'department': 'Sound',
  'gender': 0,
  'id': 900,
  'job': 'Sound Designer',
  'name': 'Christopher Boyes'},
 {'credit_id': '54491cb70e0a267480001bd0',
```

In [9]:
```
# Creates a function to extract director name

def get_director(x):
    for i in x:
        if i['job']=='Director':
            return i['name']
    return np.nan
```

In [10]:
```
def get_list(x):
    if isinstance(x,list):
        names = [i['name'] for i in x]

        if len(names) > 3:
            names = names[:3]
        return names
    return []
```

In [11]:
```
# lets apply above both function on dataset

movies_df['director'] = movies_df["crew"].apply(get_director)
features = ['cast','keywords','genres']

for feature in features:
    movies_df[feature] = movies_df[feature].apply(get_list)
```

In [12]:
```
movies_df.columns
```

Out[12]:
```
Index(['budget', 'genres', 'homepage', 'id', 'keywords', 'original_language',
       'original_title', 'overview', 'popularity', 'production_companies',
       'production_countries', 'release_date', 'revenue', 'runtime',
       'spoken_languages', 'status', 'tagline', 'title_x', 'vote_average',
       'vote_count', 'title_y', 'cast', 'crew', 'director'],
      dtype='object')
```

In [13]:
```
def clean_data(row):
    if isinstance (row,list):
        return (str.lower(i.replace(" ",""))for i in row)
    else:
        if isinstance (row,str):
            return str.lower(row.replace(" ",""))
        else:
            return ""


features = ["cast","keywords","director","genres"]

for feature in features:
    movies_df[feature]=movies_df[feature].apply(clean_data)
```

In [14]:
```
movies_df[["cast","keywords","director","genres"]].head()
```

Out[14]:

	cast	keywords	director	genres
0	<generator object clean_data.<locals>.<genexpr...	<generator object clean_data.<locals>.<genexpr...	jamescameron	<generator object clean_data.<locals>.<genexpr...
1	<generator object clean_data.<locals>.<genexpr...	<generator object clean_data.<locals>.<genexpr...	goreverbinski	<generator object clean_data.<locals>.<genexpr...
2	<generator object clean_data.<locals>.<genexpr...	<generator object clean_data.<locals>.<genexpr...	sammendes	<generator object clean_data.<locals>.<genexpr...
3	<generator object clean_data.<locals>.<genexpr...	<generator object clean_data.<locals>.<genexpr...	christophernolan	<generator object clean_data.<locals>.<genexpr...
4	<generator object clean_data.<locals>.<genexpr...	<generator object clean_data.<locals>.<genexpr...	andrewstanton	<generator object clean_data.<locals>.<genexpr...

In [15]:
```
def create_group (features):
    return " ".join(features["keywords"]) + " "+ " ".join(features["cast"]) +

movies_df["group"] = movies_df.apply(create_group,axis=1)
print(movies_df["group"].head())
```

```
0    cultureclash future spacewar samworthington zo...
1    ocean drugabuse exoticisland johnnydepp orland...
2    spy basedonnovel secretagent danielcraig chris...
3    dccomics crimefighter terrorist christianbale ...
4    basedonnovel mars medallion taylorkitsch lynnc...
Name: group, dtype: object
```

In [16]:
```
print(movies_df["group"].head(10))
```

```
0    cultureclash future spacewar samworthington zo...
1    ocean drugabuse exoticisland johnnydepp orland...
2    spy basedonnovel secretagent danielcraig chris...
3    dccomics crimefighter terrorist christianbale ...
4    basedonnovel mars medallion taylorkitsch lynnc...
5    dualidentity amnesia sandstorm tobeymaguire ki...
6    hostage magic horse zacharylevi mandymoore don...
7    marvelcomic sequel superhero robertdowneyjr. c...
8    witch magic broom danielradcliffe rupertgrint ...
9    dccomics vigilante superhero benaffleck henryc...
Name: group, dtype: object
```

In [17]:
```
count_vect=CountVectorizer(stop_words="english")
count_matrix=count_vect.fit_transform(movies_df["group"])
print(count_matrix.shape)
```

```
(4803, 9290)
```

In [18]:
```
cosine_sim=cosine_similarity(count_matrix, count_matrix)
cosine_sim.shape
```

Out[18]:
```
(4803, 4803)
```

In [19]:
```
movies_df = movies_df.reset_index()
indices = pd.Series(movies_df.index , index = movies_df['original_title'])
```

In [20]:
```
indices.head()
```

Out[20]:
```
original_title
Avatar                                         0
Pirates of the Caribbean: At World's End       1
Spectre                                        2
The Dark Knight Rises                          3
John Carter                                    4
dtype: int64
```

In [21]:
```
def get_recommendation(title,cosine_sim = cosine_sim):
    idx = indices[title]
    similarity_score = list(enumerate(cosine_sim[idx]))
    similarity_score = sorted(similarity_score,key = lambda x : x[1],reverse=T
    similarity_score = similarity_score[1:11]
    movies_indices = [ind[0] for ind in similarity_score]
    movies = movies_df['original_title'].iloc[movies_indices]
    return movies
```

In [22]:
```
print(get_recommendation("The Avengers"),cosine_sim)
```

```
7                 Avengers: Age of Ultron
26              Captain America: Civil War
79                              Iron Man 2
169     Captain America: The First Avenger
174                    The Incredible Hulk
85     Captain America: The Winter Soldier
31                              Iron Man 3
33                   X-Men: The Last Stand
68                                 Iron Man
94                 Guardians of the Galaxy
Name: original_title, dtype: object [[1.         0.33333333 0.22222222 ... 0.
0.         0.        ]
 [0.33333333 1.         0.22222222 ... 0.         0.         0.        ]
 [0.22222222 0.22222222 1.         ... 0.         0.         0.        ]
 ...
 [0.         0.         0.         ... 1.         0.         0.        ]
 [0.         0.         0.         ... 0.         1.         0.        ]
 [0.         0.         0.         ... 0.         0.         1.        ]]
```

In [23]:
```
print(get_recommendation("The Dark Knight"),cosine_sim)
```

```
3          The Dark Knight Rises
4638    Amidst the Devil's Wings
119                Batman Begins
2398                     Hitman
1720                   Kick-Ass
1986                     Faster
3326               Black November
1740                   Kick-Ass 2
1503                     Takers
303                     Catwoman
Name: original_title, dtype: object [[1.         0.33333333 0.22222222 ... 0.
0.         0.        ]
 [0.33333333 1.         0.22222222 ... 0.         0.         0.        ]
 [0.22222222 0.22222222 1.         ... 0.         0.         0.        ]
 ...
 [0.         0.         0.         ... 1.         0.         0.        ]
 [0.         0.         0.         ... 0.         1.         0.        ]
 [0.         0.         0.         ... 0.         0.         1.        ]]
```

In [24]:
```
print(get_recommendation("Spectre"),cosine_sim)
```

```
29                      Skyfall
11             Quantum of Solace
1084             The Glimmer Man
1234              The Art of War
2156                  Nancy Drew
4638    Amidst the Devil's Wings
62          The Legend of Tarzan
3373    The Other Side of Heaven
4                    John Carter
72                 Suicide Squad
Name: original_title, dtype: object [[1.         0.33333333 0.22222222 ... 0.
0.         0.        ]
 [0.33333333 1.         0.22222222 ... 0.         0.         0.        ]
 [0.22222222 0.22222222 1.         ... 0.         0.         0.        ]
 ...
 [0.         0.         0.         ... 1.         0.         0.        ]
 [0.         0.         0.         ... 0.         1.         0.        ]
 [0.         0.         0.         ... 0.         0.         1.        ]]
```

In [25]:
```
print(get_recommendation("Salt"),cosine_sim)
```

```
372                    Spy Game
677     Clear and Present Danger
1425                  Abduction
1848           Agent Cody Banks
3077                      Malone
282                   True Lies
337       A Good Day to Die Hard
392                  Safe House
914         Central Intelligence
1092            The Ghost Writer
Name: original_title, dtype: object [[1.         0.33333333 0.22222222 ... 0.
0.         0.        ]
 [0.33333333 1.         0.22222222 ... 0.         0.         0.        ]
 [0.22222222 0.22222222 1.         ... 0.         0.         0.        ]
 ...
 [0.         0.         0.         ... 1.         0.         0.        ]
 [0.         0.         0.         ... 0.         1.         0.        ]
 [0.         0.         0.         ... 0.         0.         1.        ]]
```

In [26]:
```
print(get_recommendation("Friends with Benefits"),cosine_sim)
```

```
4247        Me You and Five Bucks
3116                The Open Road
272                Town & Country
682                 The Love Guru
2513                      Tootsie
2854           How to Be a Player
3122              Blonde Ambition
3791                Among Giants
1572    Forgetting Sarah Marshall
4121      24 7: Twenty Four Seven
Name: original_title, dtype: object [[1.         0.33333333 0.22222222 ... 0.
0.         0.        ]
 [0.33333333 1.         0.22222222 ... 0.         0.         0.        ]
 [0.22222222 0.22222222 1.         ... 0.         0.         0.        ]
 ...
 [0.         0.         0.         ... 1.         0.         0.        ]
 [0.         0.         0.         ... 0.         1.         0.        ]
 [0.         0.         0.         ... 0.         0.         1.        ]]
```

In [27]:
```
print(get_recommendation("Unfaithful"),cosine_sim)
```

```
2864             Arbitrage
593            The Dilemma
1081    Revolutionary Road
2040       The Glass House
2703    A Walk on the Moon
3009                Swimfan
3587               Addicted
4647            The Canyons
1018       The Cotton Club
2151          The Bank Job
Name: original_title, dtype: object [[1.         0.33333333 0.22222222 ... 0.
0.         0.        ]
 [0.33333333 1.         0.22222222 ... 0.         0.         0.        ]
 [0.22222222 0.22222222 1.         ... 0.         0.         0.        ]
 ...
 [0.         0.         0.         ... 1.         0.         0.        ]
 [0.         0.         0.         ... 0.         1.         0.        ]
 [0.         0.         0.         ... 0.         0.         1.        ]]
```

In [ ]: