In [1]:

```
import os

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

In [2]:

```
os.getcwd()
```

Out[2]:

```
'C:\\Users\\user\\Documents\\ML Projects\\Recommender Systems Projects\\Movie Recommendation System'
```

In [3]:

```
movies = pd.read_csv("Datasets/tmdb_5000_movies.csv")
credits = pd.read_csv("Datasets/tmdb_5000_credits.csv")
```

In [4]:

```
print(movies.shape)
print(credits.shape)
```

```
(4803, 20)
(4803, 4)
```

In [5]:
```
movies.head()
```

Out[5]:

	budget	genres	homepage	id	keywords	original_language	original_title	overview	popularity	production_companies	production_cou
0	237000000	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	http://www.avatarmovie.com/	19995	[{"id": 1463, "name": "culture clash"}, {"id":...	en	Avatar	In the 22nd century, a paraplegic Marine is di...	150.437577	[{"name": "Ingenious Film Partners", "id": 289...	[{"iso_3166_1": "name": "United
1	300000000	[{"id": 12, "name": "Adventure"}, {"id": 14, "...	http://disney.go.com/disneypictures/pirates/	285	[{"id": 270, "name": "ocean"}, {"id": 726, "na...	en	Pirates of the Caribbean: At World's End	Captain Barbossa, long believed to be dead, ha...	139.082615	[{"name": "Walt Disney Pictures", "id": 2}, {"...	[{"iso_3166_1": "name": "United
2	245000000	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	http://www.sonypictures.com/movies/spectre/	206647	[{"id": 470, "name": "spy"}, {"id": 818, "name...	en	Spectre	A cryptic message from Bond’s past sends him o...	107.376788	[{"name": "Columbia Pictures", "id": 5}, {"nam...	[{"iso_3166_1": "name": "U Kingd
3	250000000	[{"id": 28, "name": "Action"}, {"id": 80, "nam...	http://www.thedarkknightrises.com/	49026	[{"id": 849, "name": "dc comics"}, {"id": 853,...	en	The Dark Knight Rises	Following the death of District Attorney Harve...	112.312950	[{"name": "Legendary Pictures", "id": 923}, {"...	[{"iso_3166_1": "name": "United
4	260000000	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	http://movies.disney.com/john-carter	49529	[{"id": 818, "name": "based on novel"}, {"id":...	en	John Carter	John Carter is a war-weary, former military ca...	43.926995	[{"name": "Walt Disney Pictures", "id": 2}]	[{"iso_3166_1": "name": "United

In [6]:

```
credits.head()
```

Out[6]:

	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 [7]:

```
movies.duplicated().sum()
```

Out[7]:

```
0
```

In [8]:

```
credits.duplicated().sum()
```

Out[8]:

```
0
```

No Duplicate rows in **movies** and **credits**.

In [9]:

```
movies.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4803 entries, 0 to 4802
Data columns (total 20 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   budget                4803 non-null   int64  
 1   genres                4803 non-null   object 
 2   homepage              1712 non-null   object 
 3   id                    4803 non-null   int64  
 4   keywords              4803 non-null   object 
 5   original_language     4803 non-null   object 
 6   original_title        4803 non-null   object 
 7   overview              4800 non-null   object 
 8   popularity            4803 non-null   float64
 9   production_companies  4803 non-null   object 
 10  production_countries  4803 non-null   object 
 11  release_date          4802 non-null   object 
 12  revenue               4803 non-null   int64  
 13  runtime               4801 non-null   float64
 14  spoken_languages      4803 non-null   object 
 15  status                4803 non-null   object 
 16  tagline               3959 non-null   object 
 17  title                 4803 non-null   object 
 18  vote_average          4803 non-null   float64
 19  vote_count            4803 non-null   int64  
dtypes: float64(3), int64(4), object(13)
memory usage: 750.6+ KB
```

In [10]:
```
isnull_ser_movies = movies.isna().sum()
print(isnull_ser_movies/len(movies)*100)
print(isnull_ser_movies[isnull_ser_movies != 0].index)
```

```
budget                   0.000000
genres                   0.000000
homepage                64.355611
id                       0.000000
keywords                 0.000000
original_language        0.000000
original_title           0.000000
overview                 0.062461
popularity               0.000000
production_companies     0.000000
production_countries     0.000000
release_date             0.020820
revenue                  0.000000
runtime                  0.041641
spoken_languages         0.000000
status                   0.000000
tagline                 17.572351
title                    0.000000
vote_average             0.000000
vote_count               0.000000
dtype: float64
Index(['homepage', 'overview', 'release_date', 'runtime', 'tagline'], dtype='object')
```

In movies dataframe: **'homepage'**, **'overview'**, **'release_date'**, **'runtime'**, **'tagline'** columns contain null values.

In [11]:
```
credits.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4803 entries, 0 to 4802
Data columns (total 4 columns):
 #   Column    Non-Null Count  Dtype 
---  ------    --------------  ----- 
 0   movie_id  4803 non-null   int64 
 1   title     4803 non-null   object
 2   cast      4803 non-null   object
 3   crew      4803 non-null   object
dtypes: int64(1), object(3)
memory usage: 150.2+ KB
```

In [12]:

```
# movie_id is the common column b/w movies and credits that has no null values.
movies.rename(columns={'id': 'movie_id'}, inplace=True)
```

In [13]:

```
movies['movie_id'].nunique()
```

Out[13]:

```
4803
```

In [14]:

```
credits['movie_id'].nunique()
```

Out[14]:

```
4803
```

In [15]:

```
movies['title'].nunique()
```

Out[15]:

```
4800
```

In [16]:

```
credits['title'].nunique()
```

Out[16]:

```
4800
```

In [17]:

```
movies['status'].value_counts()
```

Out[17]:

```
status
Released           4795
Rumored               5
Post Production       3
Name: count, dtype: int64
```

In [18]:

```
movies['original_language'].value_counts()
```

Out[18]:

```
original_language
en    4505
fr      70
es      32
zh      27
de      27
hi      19
ja      16
it      14
cn      12
ru      11
ko      11
pt       9
da       7
sv       5
nl       4
fa       4
th       3
he       3
ta       2
cs       2
ro       2
id       2
ar       2
vi       1
sl       1
ps       1
no       1
ky       1
hu       1
pl       1
af       1
nb       1
tr       1
is       1
xx       1
te       1
el       1
Name: count, dtype: int64
```

```
In [19]: movies['release_date'].describe()
```

```
Out[19]: count           4802
         unique          3280
         top       2006-01-01
         freq              10
         Name: release_date, dtype: object
```

```
In [20]: cols_to_keep = ['original_title', 'title']
         movies[movies['original_title'] != movies['title']].loc[:, cols_to_keep]
```

Out[20]:

	original_title	title
97	シン・ゴジラ	Shin Godzilla
215	4: Rise of the Silver Surfer	Fantastic 4: Rise of the Silver Surfer
235	Astérix aux Jeux Olympiques	Asterix at the Olympic Games
317	金陵十三釵	The Flowers of War
474	Évolution	Evolution
...	...	...
4699	Lumea e a mea	The World Is Mine
4719	Une femme mariée: Suite de fragments d'un film...	The Married Woman
4751	Gabriela, Cravo e Canela	Gabriela
4790	دایره	The Circle
4792	キュア	Cure

261 rows × 2 columns

In [21]:
```
credits['cast'][0]
```

Out[21]:
```
'[{"cast_id": 242, "character": "Jake Sully", "credit_id": "5602a8a7c3a3685532001c9a", "gender": 2, "id": 65731, "name": "Sam Worthington",
"order": 0}, {"cast_id": 3, "character": "Neytiri", "credit_id": "52fe48009251416c750ac9cb", "gender": 1, "id": 8691, "name": "Zoe Saldana",
"order": 1}, {"cast_id": 25, "character": "Dr. Grace Augustine", "credit_id": "52fe48009251416c750aca39", "gender": 1, "id": 10205, "name":
"Sigourney Weaver", "order": 2}, {"cast_id": 4, "character": "Col. Quaritch", "credit_id": "52fe48009251416c750ac9cf", "gender": 2, "id": 327
47, "name": "Stephen Lang", "order": 3}, {"cast_id": 5, "character": "Trudy Chacon", "credit_id": "52fe48009251416c750ac9d3", "gender": 1, "i
d": 17647, "name": "Michelle Rodriguez", "order": 4}, {"cast_id": 8, "character": "Selfridge", "credit_id": "52fe48009251416c750ac9e1", "gend
er": 2, "id": 1771, "name": "Giovanni Ribisi", "order": 5}, {"cast_id": 7, "character": "Norm Spellman", "credit_id": "52fe48009251416c750ac9
dd", "gender": 2, "id": 59231, "name": "Joel David Moore", "order": 6}, {"cast_id": 9, "character": "Moat", "credit_id": "52fe48009251416c750
ac9e5", "gender": 1, "id": 30485, "name": "CCH Pounder", "order": 7}, {"cast_id": 11, "character": "Eytukan", "credit_id": "52fe48009251416c7
50ac9ed", "gender": 2, "id": 15853, "name": "Wes Studi", "order": 8}, {"cast_id": 10, "character": "Tsu\'Tey", "credit_id": "52fe48009251416c
750ac9e9", "gender": 2, "id": 10964, "name": "Laz Alonso", "order": 9}, {"cast_id": 12, "character": "Dr. Max Patel", "credit_id": "52fe48009
251416c750ac9f1", "gender": 2, "id": 95697, "name": "Dileep Rao", "order": 10}, {"cast_id": 13, "character": "Lyle Wainfleet", "credit_id":
"52fe48009251416c750ac9f5", "gender": 2, "id": 98215, "name": "Matt Gerald", "order": 11}, {"cast_id": 32, "character": "Private Fike", "cred
it_id": "52fe48009251416c750aca5b", "gender": 2, "id": 154153, "name": "Sean Anthony Moran", "order": 12}, {"cast_id": 33, "character": "Cryo
Vault Med Tech", "credit_id": "52fe48009251416c750aca5f", "gender": 2, "id": 397312, "name": "Jason Whyte", "order": 13}, {"cast_id": 34, "ch
aracter": "Venture Star Crew Chief", "credit_id": "52fe48009251416c750aca63", "gender": 2, "id": 42317, "name": "Scott Lawrence", "order": 1
4}, {"cast_id": 35, "character": "Lock Up Trooper", "credit_id": "52fe48009251416c750aca67", "gender": 2, "id": 986734, "name": "Kelly Kilgou
r", "order": 15}, {"cast_id": 36, "character": "Shuttle Pilot", "credit_id": "52fe48009251416c750aca6b", "gender": 0, "id": 1207227, "name":
"James Patrick Pitt", "order": 16}, {"cast_id": 37, "character": "Shuttle Co-Pilot", "credit_id": "52fe48009251416c750aca6f", "gender": 0, "i
```

In [22]:
```
credits['crew'][0]
```

Out[22]:
```
'[{"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 Boye
s"}, {"credit_id": "54491cb70e0a267480001bd0", "department": "Sound", "gender": 0, "id": 900, "job": "Supervising Sound Editor", "name": "Chr
istopher Boyes"}, {"credit_id": "539c4a4cc3a36810c9002101", "department": "Production", "gender": 1, "id": 1262, "job": "Casting", "name": "M
ali Finn"}, {"credit_id": "5544ee3b925141499f0008fc", "department": "Sound", "gender": 2, "id": 1729, "job": "Original Music Composer", "nam
e": "James Horner"}, {"credit_id": "52fe48009251416c750ac9c3", "department": "Directing", "gender": 2, "id": 2710, "job": "Director", "name":
"James Cameron"}, {"credit_id": "52fe48009251416c750ac9d9", "department": "Writing", "gender": 2, "id": 2710, "job": "Writer", "name": "James
Cameron"}, {"credit_id": "52fe48009251416c750aca17", "department": "Editing", "gender": 2, "id": 2710, "job": "Editor", "name": "James Camero
n"}, {"credit_id": "52fe48009251416c750aca29", "department": "Production", "gender": 2, "id": 2710, "job": "Producer", "name": "James Camero
n"}, {"credit_id": "52fe48009251416c750aca3f", "department": "Writing", "gender": 2, "id": 2710, "job": "Screenplay", "name": "James Camero
n"}, {"credit_id": "539c4987c3a36810ba0021a4", "department": "Art", "gender": 2, "id": 7236, "job": "Art Direction", "name": "Andrew Menzie
s"}, {"credit_id": "549598c3c3a3686ae9004383", "department": "Visual Effects", "gender": 0, "id": 6690, "job": "Visual Effects Producer", "na
me": "Jill Brooks"}, {"credit_id": "52fe48009251416c750aca4b", "department": "Production", "gender": 1, "id": 6347, "job": "Casting", "name":
"Margery Simkin"}, {"credit_id": "570b6f419251417da70032fe", "department": "Art", "gender": 2, "id": 6878, "job": "Supervising Art Director",
"name": "Kevin Ishioka"}, {"credit_id": "5495a0fac3a3686ae9004468", "department": "Sound", "gender": 0, "id": 6883, "job": "Music Editor", "n
ame": "Dick Bernstein"}, {"credit_id": "54959706c3a3686af3003e81", "department": "Sound", "gender": 0, "id": 8159, "job": "Sound Effects Edit
or", "name": "Shannon Mills"}, {"credit_id": "54491d58c3a3680fb1001ccb", "department": "Sound", "gender": 0, "id": 8160, "job": "Foley", "nam
e": "Dennie Thorpe"}, {"credit_id": "54491d6cc3a3680fa5001b2c", "department": "Sound", "gender": 0, "id": 8163, "job": "Foley", "name": "Jana
```

```
In [23]: movies[movies['original_language'] == 'fr']['genres'][235]
```

```
Out[23]: '[{"id": 14, "name": "Fantasy"}, {"id": 12, "name": "Adventure"}, {"id": 35, "name": "Comedy"}, {"id": 10751, "name": "Family"}]'
```

```
In [24]: movies['genres'][0]
```

```
Out[24]: '[{"id": 28, "name": "Action"}, {"id": 12, "name": "Adventure"}, {"id": 14, "name": "Fantasy"}, {"id": 878, "name": "Science Fiction"}]'
```

```
In [25]: movies['keywords'][0]
```

```
Out[25]: '[{"id": 1463, "name": "culture clash"}, {"id": 2964, "name": "future"}, {"id": 3386, "name": "space war"}, {"id": 3388, "name": "space colony"},
         {"id": 3679, "name": "society"}, {"id": 3801, "name": "space travel"}, {"id": 9685, "name": "futuristic"}, {"id": 9840, "name": "romance"}, {"i
         d": 9882, "name": "space"}, {"id": 9951, "name": "alien"}, {"id": 10148, "name": "tribe"}, {"id": 10158, "name": "alien planet"}, {"id": 10987,
         "name": "cgi"}, {"id": 11399, "name": "marine"}, {"id": 13065, "name": "soldier"}, {"id": 14643, "name": "battle"}, {"id": 14720, "name": "love a
         ffair"}, {"id": 165431, "name": "anti war"}, {"id": 193554, "name": "power relations"}, {"id": 206690, "name": "mind and soul"}, {"id": 209714,
         "name": "3d"}]'
```

```
In [26]: movies['original_language'][0]
```

```
Out[26]: 'en'
```

```
In [27]: movies['spoken_languages'][0]
```

```
Out[27]: '[{"iso_639_1": "en", "name": "English"}, {"iso_639_1": "es", "name": "Espa\\u00f1ol"}]'
```

```
In [28]: movies['movie_id'][0]
```

```
Out[28]: 19995
```

```
In [29]: credits['movie_id'][0]
```

```
Out[29]: 19995
```

```
In [30]: movies['overview'][0]
```

```
Out[30]: 'In the 22nd century, a paraplegic Marine is dispatched to the moon Pandora on a unique mission, but becomes torn between following orders and pr
         otecting an alien civilization.'
```

```
In [31]: movies['tagline'][0]
```

```
Out[31]: 'Enter the World of Pandora.'
```

```
In [32]: movies['title'][0]
```

```
Out[32]: 'Avatar'
```

Building a recommender system that uses **Content Based filtering**:

Significant variable for Content based filtering:

'genres', 'id', 'keywords', 'original_language', 'overview', 'release_date', 'runtime', 'spoken_languages', 'tagline', 'title', 'cast', 'crew'

But since we do not have user profile information in this project therefore we would not use 'original_language', 'spoken_languages', 'runtime'.

We are dropping 'tagline' because over 17% missing values are there. And imputing them based on 'overview' means increased complexity of the recommender system.

```
In [33]: isnull_ser_movies.index
```

```
Out[33]: 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', 'vote_average',
               'vote_count'],
              dtype='object')
```

```
In [34]: movies1 = movies.copy()
         movies1.drop(columns='title', inplace=True)
         movies1 = movies1.merge(credits, on="movie_id")
```

In [35]:
```
movies1 = movies1[['genres', 'movie_id', 'keywords', 'overview',
          'release_date', 'title', 'cast', 'crew']]
movies1.head()
```

Out[35]:

	genres	movie_id	keywords	overview	release_date	title	cast	crew
0	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	19995	[{"id": 1463, "name": "culture clash"}, {"id":...	In the 22nd century, a paraplegic Marine is di...	2009-12-10	Avatar	[{"cast_id": 242, "character": "Jake Sully", "...	[{"credit_id": "52fe48009251416c750aca23", "de...
1	[{"id": 12, "name": "Adventure"}, {"id": 14, "...	285	[{"id": 270, "name": "ocean"}, {"id": 726, "na...	Captain Barbossa, long believed to be dead, ha...	2007-05-19	Pirates of the Caribbean: At World's End	[{"cast_id": 4, "character": "Captain Jack Spa...	[{"credit_id": "52fe4232c3a36847f800b579", "de...
2	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	206647	[{"id": 470, "name": "spy"}, {"id": 818, "name...	A cryptic message from Bond's past sends him o...	2015-10-26	Spectre	[{"cast_id": 1, "character": "James Bond", "cr...	[{"credit_id": "54805967c3a36829b5002c41", "de...
3	[{"id": 28, "name": "Action"}, {"id": 80, "nam...	49026	[{"id": 849, "name": "dc comics"}, {"id": 853,...	Following the death of District Attorney Harve...	2012-07-16	The Dark Knight Rises	[{"cast_id": 2, "character": "Bruce Wayne / Ba...	[{"credit_id": "52fe4781c3a36847f81398c3", "de...
4	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	49529	[{"id": 818, "name": "based on novel"}, {"id":...	John Carter is a war-weary, former military ca...	2012-03-07	John Carter	[{"cast_id": 5, "character": "John Carter", "c...	[{"credit_id": "52fe479ac3a36847f813eaa3", "de...

Dropping the rows having null value for **'release_date'** and **'overview'**.

In [36]:
```
movies1[movies1['release_date'].isnull()]
```

Out[36]:

	genres	movie_id	keywords	overview	release_date	title	cast	crew
4553	[]	380097	[]	1971 post civil rights San Francisco seemed li...	NaN	America Is Still the Place	[]	[]

In [37]:
```
movies1[movies1['overview'].isnull()]
```

Out[37]:

	genres	movie_id	keywords	overview	release_date	title	cast	crew
2656	[{"id": 18, "name": "Drama"}]	370980	[{"id": 717, "name": "pope"}, {"id": 5565, "na...	NaN	2015-12-03	Chiamatemi Francesco - Il Papa della gente	[{"cast_id": 5, "character": "Jorge Mario Berg...	[{"credit_id": "5660019ac3a36875f100252b", "de...
4140	[{"id": 99, "name": "Documentary"}]	459488	[{"id": 6027, "name": "music"}, {"id": 225822,...	NaN	2015-12-12	To Be Frank, Sinatra at 100	[{"cast_id": 0, "character": "Narrator", "cred...	[{"credit_id": "592b25e4c3a368783e065a2f", "de...
4431	[{"id": 99, "name": "Documentary"}]	292539	[]	NaN	2014-04-26	Food Chains	[]	[{"credit_id": "5470c3b1c3a368085e000abd", "de...

In [38]:

```
movies1.dropna(inplace=True)
```

In [39]:

```
movies1.isnull().sum()
```

Out[39]:

```
genres          0
movie_id        0
keywords        0
overview        0
release_date    0
title           0
cast            0
crew            0
dtype: int64
```

In [40]:

```
# Coverting the string into the right data type, extracting the value
# corresponding to 'name' key and adding it into a list.
def extract_name(s):
    list1 = []
    for el in eval(s):
        list1.append(el["name"])
    return list1
```

In [41]:

```
def extract_director_name(s):
    list1 = []
    for el in eval(s):
        if el["job"] == "Director":
            list1.append(el["name"])
    return list1
```

In [42]:

```
# Extracting the top 4 main casts.
# Note: According to the meta data casts are listed in the order
# they appear in the credits.
def extract_main_cast_info(s):

    list1 = []
    count = 1

    for el in eval(s):
        list1.append(el["character"])
        list1.append(el["name"])
        if count == 4:
            break

        count += 1
    return list1
```

In [43]:

```
from datetime import datetime
```

In [44]:

```
def extract_release_year(s):
    return [str(datetime.strptime(s, '%Y-%m-%d').year)]
```

In [45]:

```
import nltk
nltk.download('punkt')

# Stopwords.
from nltk.corpus import stopwords
nltk.download('stopwords')

from nltk.stem import PorterStemmer

import string
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\user\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\user\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

In [46]:

```
translator = str.maketrans('', '', string.punctuation)
porter_stemmer = PorterStemmer()
```

In [47]:

```
sentence = "In the 22nd century, a paraplegic Marine is dispatched to the moon Pandora on a unique mission, but becomes torn between following orde
words = nltk.word_tokenize(sentence.lower().translate(translator))
words = [porter_stemmer.stem(word) for word in words if word not in stopwords.words('english')]
print(words)
```

```
['22nd', 'centuri', 'parapleg', 'marin', 'dispatch', 'moon', 'pandora', 'uniqu', 'mission', 'becom', 'torn', 'follow', 'order', 'protect', 'alie
n', 'civil']
```

In [48]:

```
def extract_normalized_words(s):
    words = nltk.word_tokenize(s.lower().translate(translator))
    words = [porter_stemmer.stem(word) for word in words if word not in stopwords.words('english')]
    return words
```

In [49]:

```
movies1['genres'] = movies1['genres'].apply(extract_name)
```

In [50]:

```
movies1['keywords'] = movies1['keywords'].apply(extract_name)
```

In [51]:

```
movies1['cast'] = movies1['cast'].apply(extract_main_cast_info)
```

In [52]:

```
movies1['crew'] = movies1['crew'].apply(extract_director_name)
```

In [53]:

```
movies1['release_date'] = movies1['release_date'].apply(extract_release_year)
```

In [54]:

```
movies1['overview'] = movies1['overview'].apply(extract_normalized_words)
```

In [55]:
```
movies1.isna().sum()
```

Out[55]:
```
genres          0
movie_id        0
keywords        0
overview        0
release_date    0
title           0
cast            0
crew            0
dtype: int64
```

In [56]:
```
type(movies1['genres'][0])
```

Out[56]:
```
list
```

In [57]:
```
movies1['genres'][0]
```

Out[57]:
```
['Action', 'Adventure', 'Fantasy', 'Science Fiction']
```

In [58]:
```
movies1['keywords'][0]
```

Out[58]:
```
['culture clash',
 'future',
 'space war',
 'space colony',
 'society',
 'space travel',
 'futuristic',
 'romance',
 'space',
 'alien',
 'tribe',
 'alien planet',
 'cgi',
 'marine',
 'soldier',
 'battle',
 'love affair',
 'anti war',
 'power relations',
 'mind and soul',
 '3d']
```

```
In [59]: movies1['overview'][0]
```

```
Out[59]: ['22nd',
          'centuri',
          'parapleg',
          'marin',
          'dispatch',
          'moon',
          'pandora',
          'uniqu',
          'mission',
          'becom',
          'torn',
          'follow',
          'order',
          'protect',
          'alien',
          'civil']
```

```
In [60]: movies1['cast'][0]
```

```
Out[60]: ['Jake Sully',
          'Sam Worthington',
          'Neytiri',
          'Zoe Saldana',
          'Dr. Grace Augustine',
          'Sigourney Weaver',
          'Col. Quaritch',
          'Stephen Lang']
```

```
In [61]: movies1.columns
```

```
Out[61]: Index(['genres', 'movie_id', 'keywords', 'overview', 'release_date', 'title',
                'cast', 'crew'],
               dtype='object')
```

Creating a column named **'Context'** which contains the attributes of each corresponding movie. These attributes are: **'genres'**, **'keywords'**, **'overview'**, **'release_date'**, **'cast'**, **'crew'**

```
In [62]: movies1['context'] = (movies1['genres'] + movies1['keywords'] + movies1['overview']
                              + movies1['release_date'] + movies1['cast'] + movies1['crew'])
```

```
In [63]: movies1['context'][0]
```

```
Out[63]: ['Action',
          'Adventure',
          'Fantasy',
          'Science Fiction',
          'culture clash',
          'future',
          'space war',
          'space colony',
          'society',
          'space travel',
          'futuristic',
          'romance',
          'space',
          'alien',
          'tribe',
          'alien planet',
          'cgi',
          'marine',
          'soldier',
          'battle',
          'love affair',
          'anti war',
          'power relations',
          'mind and soul',
          '3d',
          '22nd',
          'centuri',
          'parapleg',
          'marin',
          'dispatch',
          'moon',
          'pandora',
          'uniqu',
          'mission',
          'becom',
          'torn',
          'follow',
          'order',
          'protect',
          'alien',
          'civil',
          '2009',
          'Jake Sully',
          'Sam Worthington',
          'Neytiri',
          'Zoe Saldana',
          'Dr. Grace Augustine',
          'Sigourney Weaver',
          'Col. Quaritch',
          'Stephen Lang',
          'James Cameron']
```

In [64]:

```
movies2 = movies1.copy()
movies2.drop(columns=['genres', 'keywords', 'overview', 'release_date',
       'cast', 'crew'], inplace=True)
```

In [65]:

```
movies2['context'] = movies2['context'].apply(lambda x: " ".join(x))
```

In [66]:

```
movies2.head()
```

Out[66]:

	movie_id	title	context
0	19995	Avatar	Action Adventure Fantasy Science Fiction cultu...
1	285	Pirates of the Caribbean: At World's End	Adventure Fantasy Action ocean drug abuse exot...
2	206647	Spectre	Action Adventure Crime spy based on novel secr...
3	49026	The Dark Knight Rises	Action Crime Drama Thriller dc comics crime fi...
4	49529	John Carter	Action Adventure Science Fiction based on nove...

In [67]:

```
movies2['context'][0]
```

Out[67]:

```
'Action Adventure Fantasy Science Fiction culture clash future space war space colony society space travel futuristic romance space alien tribe a
lien planet cgi marine soldier battle love affair anti war power relations mind and soul 3d 22nd centuri parapleg marin dispatch moon pandora uni
qu mission becom torn follow order protect alien civil 2009 Jake Sully Sam Worthington Neytiri Zoe Saldana Dr. Grace Augustine Sigourney Weaver C
ol. Quaritch Stephen Lang James Cameron'
```

In [68]:

```
from sklearn.feature_extraction.text import CountVectorizer
```

In [69]:

```
cv = CountVectorizer(stop_words = 'english')
cv
```

Out[69]:

```
▾            CountVectorizer
CountVectorizer(stop_words='english')
```

In [70]:

```
# Learning the vocabulary and creating a token count matrix in one go.
# By default each token would be a word n-gram.

words = cv.fit_transform(movies2['context'])
words
```

Out[70]:

```
<4799x33621 sparse matrix of type '<class 'numpy.int64'>'
        with 255042 stored elements in Compressed Sparse Row format>
```

In [71]:

```
# Shape of the token count matrix.
words.shape
```

Out[71]:

```
(4799, 33621)
```

In [72]:

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

In [73]:

```
similarity_matrix = cosine_similarity(words)
```

In [74]:

```
similarity_matrix
```

Out[74]:

```
array([[1.        , 0.04868645, 0.05229764, ..., 0.04546629, 0.03892495,
        0.        ],
       [0.04868645, 1.        , 0.02864459, ..., 0.03735437, 0.        ,
        0.        ],
       [0.05229764, 0.02864459, 1.        , ..., 0.02675002, 0.0763381 ,
        0.        ],
       ...,
       [0.04546629, 0.03735437, 0.02675002, ..., 1.        , 0.0132733 ,
        0.02381628],
       [0.03892495, 0.        , 0.0763381 , ..., 0.0132733 , 1.        ,
        0.01359318],
       [0.        , 0.        , 0.        , ..., 0.02381628, 0.01359318,
        1.        ]])
```

In [75]:

```
similarity_matrix.shape
```

Out[75]:

```
(4799, 4799)
```

In [76]:

```
similarity_matrix[0]
```

Out[76]:

```
array([1.        , 0.04868645, 0.05229764, ..., 0.04546629, 0.03892495,
       0.        ])
```

In [77]:
```
movies2[movies2['title'] == 'Avatar'].index[0]
```

Out[77]:
```
0
```

In [78]:
```
a = np.random.randint(1,31, (3,6))
e = enumerate(a[0])
e
```

Out[78]:
```
<enumerate at 0x18a390e1800>
```

In [79]:
```
list1 = list(e)
list1
sorted(list1, reverse=True, key=lambda x: x[1])
```

Out[79]:
```
[(3, 29), (4, 29), (0, 28), (1, 23), (5, 16), (2, 11)]
```

In [80]:
```
len(list1)
```

Out[80]:
```
6
```

In [81]:
```
# movie_name: a valid movie name that should be there in the dataset

# count: specifies the number of similar movies to be recommended as
# per the movie_name

# Print movies similar to the movie_name
def search_related_movies_by_name(movie_name, count):
    try:
        movie_index = movies2[movies2['title'].apply(lambda title: title.lower()) ==
                movie_name.lower()].index[0]

        distances = sorted(list(enumerate(similarity_matrix[movie_index])),
                           reverse=True, key = lambda x :x[1])

        for t in distances[1:count+1]:
          # print(f"movie: {movies2['title'][t[0]]}")
          print(f"movie: {movies2['title'][t[0]]}, similarity score: {t[1]}")
    except IndexError:
        print(f"'{movie_name}', was not found! Try some other movie name")
```

In [83]:

```
# query: specifies the tpe of movies to be recommended
# count: specifies the number of similar movies to be recommended as per the query

def search_related_movies_by_query(query, count):
    normalized_query = " ".join(extract_normalized_words(query))
    print(f'normalized_query: {normalized_query}\n')
    matrix = cv.transform([query])
    result = cosine_similarity(matrix, words)[0]
    sorted_result = sorted(list(enumerate(result)), reverse=True, key=lambda x: x[1])

    # print(sorted_result)

    for t in sorted_result[0:count]:
        if t[1] == 0:
            print(f'\nCould not find {count} related movies because the similarity scores of the query with the remaining movies are 0.')
            break
        # print(f"movie: {movies2['title'][t[0]]}")
        print(f"movie: {movies2['title'][t[0]]}, similarity score: {t[1]}")
```

In [84]:

```
def movie_recommender(query, is_movie_name=True, count=5):
    if is_movie_name:
        search_related_movies_by_name(query, count)
    else:
        search_related_movies_by_query(query, count)
```

In [85]:

```
movie_recommender('Avatar', is_movie_name=True)
```

```
movie: Aliens, similarity score: 0.3531887144873645
movie: Lifeforce, similarity score: 0.3409971697352367
movie: Moonraker, similarity score: 0.3167762968124701
movie: Lockout, similarity score: 0.3105295017040594
movie: Mission to Mars, similarity score: 0.30988989340045614
```

In [86]:

```
movie_recommender('Avatar', is_movie_name=False)
```

```
normalized_query: avatar

movie: The Last Airbender, similarity score: 0.25
movie: Bronson, similarity score: 0.1270001270001905
movie: 16 to Life, similarity score: 0.125

Could not find 5 related movies because the similarity scores of the query with the remaining movies are 0.
```

```
In [87]: movie_recommender('batman', is_movie_name=True)
```

```
movie: The Dark Knight, similarity score: 0.41640438621500747
movie: Batman & Robin, similarity score: 0.3960590171906697
movie: Batman Returns, similarity score: 0.39088155249705214
movie: Batman Begins, similarity score: 0.38118124993124386
movie: The Dark Knight Rises, similarity score: 0.3765330442186538
```

```
In [88]: movie_recommender('batman', is_movie_name=False)
```

```
normalized_query: batman

movie: Batman Returns, similarity score: 0.4029114820126901
movie: 2:13, similarity score: 0.36650833306891567
movie: The Dark Knight Rises, similarity score: 0.33567254331867563
movie: The Dark Knight, similarity score: 0.26413527189768715
movie: Batman Forever, similarity score: 0.2508726030021272
```

```
In [89]: movie_recommender('the godfather', is_movie_name=True)
```

```
movie: Center Stage, similarity score: 0.24123649045751025
movie: Act of Valor, similarity score: 0.21774229673875098
movie: Step Up 2: The Streets, similarity score: 0.2098217272655633
movie: Take the Lead, similarity score: 0.203738641308575
movie: Gone with the Wind, similarity score: 0.20191135200894694
```

```
In [90]: movie_recommender('the godfather', is_movie_name=False)
```

```
normalized_query: godfath


Could not find 5 related movies because the similarity scores of the query with the remaining movies are 0.
```

```
In [91]: movie_recommender('Sherlock Holmes', is_movie_name=True)
```

```
movie: Sherlock Holmes: A Game of Shadows, similarity score: 0.44150314702736076
movie: Young Sherlock Holmes, similarity score: 0.40422604172722154
movie: Shaft, similarity score: 0.18845713897261818
movie: Midnight in the Garden of Good and Evil, similarity score: 0.16643851232744458
movie: Die Hard: With a Vengeance, similarity score: 0.16141167423097186
```

In [92]:

```
movie_recommender('Sherlock Holmes', is_movie_name=False)
```

```
normalized_query: sherlock holm

movie: Sherlock Holmes, similarity score: 0.42874646285627205
movie: Young Sherlock Holmes, similarity score: 0.39283710065919303
movie: Sherlock Holmes: A Game of Shadows, similarity score: 0.19802950859533483
movie: The Claim, similarity score: 0.18107149208503706
movie: Eulogy, similarity score: 0.12126781251816646
```

In [93]:

```
movie_recommender('Detective genre', is_movie_name=False)
```

```
normalized_query: detect genr

movie: Se7en, similarity score: 0.29851115706299675
movie: The Eclipse, similarity score: 0.2857142857142857
movie: Bad Boys, similarity score: 0.2822162605150792
movie: Girl 6, similarity score: 0.26037782196164777
movie: In the Valley of Elah, similarity score: 0.24433888871261045
```

In [94]:

```
movie_recommender('action and suspense', is_movie_name=False)
```

```
normalized_query: action suspens

movie: The Transporter Refueled, similarity score: 0.32232918561015206
movie: Special, similarity score: 0.3162277660168379
movie: Better Luck Tomorrow, similarity score: 0.26726124191242434
movie: The Blair Witch Project, similarity score: 0.254000254000381
movie: The Man with the Golden Gun, similarity score: 0.2357022603955158
```

In [95]:

```
movie_recommender('comedy movies', is_movie_name=False)
```

```
normalized_query: comedi movi

movie: There's Something About Mary, similarity score: 0.32232918561015206
movie: Special, similarity score: 0.3162277660168379
movie: Manito, similarity score: 0.3
movie: Twin Falls Idaho, similarity score: 0.2721655269759087
movie: American Hero, similarity score: 0.25537695922762454
```

```
In [96]: movie_recommender('movies involving magic', is_movie_name=False)
```

```
normalized_query: movi involv magic

movie: Last Action Hero, similarity score: 0.21650635094610965
movie: Practical Magic, similarity score: 0.21213203435596423
movie: The Oogieloves in the Big Balloon Adventure, similarity score: 0.18749999999999997
movie: Harry Potter and the Chamber of Secrets, similarity score: 0.18731716231633877
movie: Oz: The Great and Powerful, similarity score: 0.18569533817705183
```

```
In [97]: movie_recommender('war movies', is_movie_name=False)
```

```
normalized_query: war movi

movie: The Hunting Party, similarity score: 0.429197537639476
movie: Unbroken, similarity score: 0.4029114820126901
movie: Saving Private Ryan, similarity score: 0.37947331922020544
movie: Awake, similarity score: 0.36901248321155405
movie: The Great Raid, similarity score: 0.3646624787447363
```

```
In [98]: movie_recommender('adventure movies', is_movie_name=False)
```

```
normalized_query: adventur movi

movie: Inkheart, similarity score: 0.20628424925175867
movie: Year One, similarity score: 0.19611613513818402
movie: 30 Minutes or Less, similarity score: 0.19611613513818402
movie: The Adventures of Rocky & Bullwinkle, similarity score: 0.16222142113076252
movie: Puss in Boots, similarity score: 0.159111456835146
```

```
In [99]: movie_recommender('science fiction movies', is_movie_name=False)
```

```
normalized_query: scienc fiction movi

movie: Special, similarity score: 0.5163977794943223
movie: Timecop, similarity score: 0.30323921743156135
movie: Ender's Game, similarity score: 0.25819888974716115
movie: Red Planet, similarity score: 0.24743582965269678
movie: An Ideal Husband, similarity score: 0.24618298195866548
```

In [100]:

```
movie_recommender('kung fu', is_movie_name=False)
```

```
normalized_query: kung fu

movie: City of Life and Death, similarity score: 0.6135719910778963
movie: Bulletproof Monk, similarity score: 0.3563483225498991
movie: Kung Fu Panda 2, similarity score: 0.3475706678180953
movie: Kung Fu Panda, similarity score: 0.24164883733207076
movie: The Grandmaster, similarity score: 0.2332847374079217
```

In [101]:

```
movie_recommender('mummy', is_movie_name=False)
```

```
normalized_query: mummi

movie: The Forest, similarity score: 0.13245323570650439
movie: Hotel Transylvania 2, similarity score: 0.11043152607484653
movie: Hotel Transylvania, similarity score: 0.10721125348377948
movie: Home Run, similarity score: 0.09166984970282113

Could not find 5 related movies because the similarity scores of the query with the remaining movies are 0.
```

In [102]:

```
movie_recommender('spiderman', is_movie_name=False)
```

```
normalized_query: spiderman

movie: Spider-Man 3, similarity score: 0.211999576001272
movie: Spider-Man 2, similarity score: 0.10314212462587934
movie: The Amazing Spider-Man 2, similarity score: 0.09053574604251853
movie: The Amazing Spider-Man, similarity score: 0.08137884587711594

Could not find 5 related movies because the similarity scores of the query with the remaining movies are 0.
```

In [103]:

```
movies2['title'][:40]
```

Out[103]:

```
0                                          Avatar
1        Pirates of the Caribbean: At World's End
2                                         Spectre
3                           The Dark Knight Rises
4                                     John Carter
5                                    Spider-Man 3
6                                         Tangled
7                         Avengers: Age of Ultron
8          Harry Potter and the Half-Blood Prince
9              Batman v Superman: Dawn of Justice
10                               Superman Returns
11                              Quantum of Solace
12     Pirates of the Caribbean: Dead Man's Chest
13                                The Lone Ranger
14                                   Man of Steel
15       The Chronicles of Narnia: Prince Caspian
16                                   The Avengers
17    Pirates of the Caribbean: On Stranger Tides
18                                 Men in Black 3
19      The Hobbit: The Battle of the Five Armies
20                         The Amazing Spider-Man
21                                     Robin Hood
22            The Hobbit: The Desolation of Smaug
23                             The Golden Compass
24                                      King Kong
25                                        Titanic
26                     Captain America: Civil War
27                                     Battleship
28                                 Jurassic World
29                                        Skyfall
30                                   Spider-Man 2
31                                     Iron Man 3
32                            Alice in Wonderland
33                          X-Men: The Last Stand
34                            Monsters University
35            Transformers: Revenge of the Fallen
36                Transformers: Age of Extinction
37                     Oz: The Great and Powerful
38                       The Amazing Spider-Man 2
39                                   TRON: Legacy
Name: title, dtype: object
```