

In [3]:

Importing Datasets

In [4]:

In [5]:

Out[5]:

◀ ▶

In [6]:

Out[6]:

| | movie_id | title | cast | crew |
|---|----------|--------|---|------|
| 0 | 19995 | Avatar | {{"cast_id": 242, "character": "Jake Sully", "... [{"credit_id": "52fe48009251416c750aca23", "de... | |

Merging the two datasets

In [7]:

Out[7]:

(4803, 20)

In [8]:

Out[8]:

 $(4803, 4)$

In [9]:

Out[9]: $(4809, 23)$

Filtering the columns we required

1 genres 2id 3keywords 4title 5overview 6cast 7crew

In [10]:

```
movies = movies[['movie_id','title','overview','genres','keywords','cast','crew']]
```

In [11]:

```
movies.head(2)
```

Out[11]:

| | movie_id | title | overview | genres | keywords | cast | crew |
|---|----------|--|---|---|--|---|---|
| 0 | 19995 | Avatar | In the 22nd century, a paraplegic Marine is di... | [{"id": 28, "name": "Action"}, {"id": 12, "nam... | [{"id": 1463, "name": "culture clash"}, {"id": ... | [{"cast_id": 242, "character": "Jake Sully", "... | [{"credit_id": "52fe48009251416c750aca23", "de... |
| 1 | 285 | Pirates of the Caribbean: At World's End | Captain Barbossa, long believed to be dead, ha... | [{"id": 12, "name": "Adventure"}, {"id": 14, "... | [{"id": 270, "name": "ocean"}, {"id": 726, "na... | [{"cast_id": 4, "character": "Captain Jack Spa... | [{"credit_id": "52fe4232c3a36847f800b579", "de... |

In [12]:

```
movies.isnull().sum()
```

Out[12]:

```
movie_id    0
title       0
overview    3
genres      0
keywords    0
cast        0
crew        0
dtype: int64
```

In [13]:

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

In [14]:

```
movies.isnull().sum()
```

Out[14]:

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

In [15]:

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

Out[15]:

```
0
```

In [16]:

```
movies.iloc[0].genres
```

Out[16]:

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

In [17]:

```
def convert(obj):
    L = []
    for i in eval(obj):
        L.append(i['name'])

    return L
```

In [18]:

```
movies['genres'] = movies['genres'].apply(convert)
```

In [19]:

```
movies['keywords'] = movies['keywords'].apply(convert)
```

In [20]:

```
movies.head()
```

Out[20]:

| | movie_id | title | overview | genres | keywords | cast | crew |
|---|----------|--|---|---|---|---|---|
| 0 | 19995 | Avatar | In the 22nd century, a paraplegic Marine is di... | [Action, Adventure, Fantasy, Science Fiction] | [culture clash, future, space war, space colon... | [{"cast_id": 242, "character": "Jake Sully", "... | [{"credit_id": "52fe48009251416c750aca23", "de... |
| 1 | 285 | Pirates of the Caribbean: At World's End | Captain Barbossa, long believed to be dead, ha... | [Adventure, Fantasy, Action] | [ocean, drug abuse, exotic island, east india ... | [{"cast_id": 4, "character": "Captain Jack Spa... | [{"credit_id": "52fe4232c3a36847f800b579", "de... |
| 2 | 206647 | Spectre | A cryptic message from Bond's past sends him o... | [Action, Adventure, Crime] | [spy, based on novel, secret agent, sequel, mi... | [{"cast_id": 1, "character": "James Bond", "cr... | [{"credit_id": "54805967c3a36829b5002c41", "de... |
| 3 | 49026 | The Dark Knight Rises | Following the death of District Attorney Harve... | [Action, Crime, Drama, Thriller] | [dc comics, crime fighter, terrorist, secret i... | [{"cast_id": 2, "character": "Bruce Wayne / Ba... | [{"credit_id": "52fe4781c3a36847f81398c3", "de... |
| 4 | 49529 | John Carter | John Carter is a war-weary, former military ca... | [Action, Adventure, Science Fiction] | [based on novel, mars, medallion, space travel... | [{"cast_id": 5, "character": "John Carter", "c... | [{"credit_id": "52fe479ac3a36847f813eaa3", "de... |

In [21]:

```
def convert3(obj):
    L = []
    counter = 0
    for i in eval(obj):
        if counter != 3:
            L.append(i['name'])
            counter += 1
        else:
            break
    return L
```

In [22]:

```
movies['cast'] = movies['cast'].apply(convert3)
```

In [23]:

```
def fetch_director(obj):
    L = []
    for i in eval(obj):
        if i['job']=='Director' :
            L.append(i['name'])
            break
    return L
```

In [24]:

```
movies['crew'] = movies['crew'].apply(fetch_director)
```

In [25]:

```
movies['overview'] = movies['overview'].apply(lambda x:x.split())
```

In [26]:

```
movies.head(2)
```

Out[26]:

| | movie_id | title | overview | genres | keywords | cast | crew |
|---|----------|--|---|---|---|--|------------------|
| 0 | 19995 | Avatar | [In, the, 22nd, century,, a, paraplegic, Marin... | [Action, Adventure, Fantasy, Science Fiction] | [culture clash, future, space war, space colon... | [Sam Worthington, Zoe Saldana, Sigourney Weaver] | [James Cameron] |
| 1 | 285 | Pirates of the Caribbean: At World's End | [Captain, Barbossa,, long, believed, to, be, d... | [Adventure, Fantasy, Action] | [ocean, drug abuse, exotic island, east india ... | [Johnny Depp, Orlando Bloom, Keira Knightley] | [Gore Verbinski] |

In [27]:

```
movies['crew'] =movies['crew'].apply(lambda x: [i.replace(" ", "") for i in x])
movies['keywords'] =movies['keywords'].apply(lambda x: [i.replace(" ", "") for i in x])
movies['cast'] =movies['cast'].apply(lambda x: [i.replace(" ", "") for i in x])
movies['genres'] =movies['genres'].apply(lambda x: [i.replace(" ", "") for i in x])
```

In [28]:

```
movies.head()
```

Out[28]:

| | movie_id | title | overview | genres | keywords | cast | crew |
|---|----------|--|---|--|---|---|--------------------|
| 0 | 19995 | Avatar | [In, the, 22nd, century,, a, paraplegic, Marin... | [Action, Adventure, Fantasy, ScienceFiction] | [cultureclash, future, spacewar, spacecolony, ... | [SamWorthington, ZoeSaldana, SigourneyWeaver] | [JamesCameron] |
| 1 | 285 | Pirates of the Caribbean: At World's End | [Captain, Barbossa,, long, believed, to, be, d... | [Adventure, Fantasy, Action] | [ocean, drugabuse, exoticisland, eastindiatrad... | [JohnnyDepp, OrlandoBloom, KeiraKnightley] | [GoreVerbinski] |
| 2 | 206647 | Spectre | [A, cryptic, message, from, Bond's, past, send... | [Action, Adventure, Crime] | [spy, basedonnovel, secretagent, sequel, mi6, ... | [DanielCraig, ChristophWaltz, LéaSeydoux] | [SamMendes] |
| 3 | 49026 | The Dark Knight Rises | [Following, the, death, of, District, Attorney... | [Action, Crime, Drama, Thriller] | [dcomics, crimefighter, terrorist, secretiden... | [ChristianBale, MichaelCaine, GaryOldman] | [ChristopherNolan] |
| 4 | 49529 | John Carter | [John, Carter, is, a, war-weary,, former, mili... | [Action, Adventure, ScienceFiction] | [basedonnovel, mars, medallion, spacetravel, p... | [TaylorKitsch, LynnCollins, SamanthaMorton] | [AndrewStanton] |

In [29]:

```
movies['tags'] = movies['overview']+movies['genres']+movies['keywords']+movies['cast'] +movies['crew']
```

In [30]:

```
movies.head()
```

Out[30]:

| | movie_id | title | overview | genres | keywords | cast | crew | tags |
|---|----------|--|---|--|---|---|--------------------|---|
| 0 | 19995 | Avatar | [In, the, 22nd, century,, a, paraplegic, Marin... | [Action, Adventure, Fantasy, ScienceFiction] | [cultureclash, future, spacewar, spacecolony, ... | [SamWorthington, ZoeSaldana, SigourneyWeaver] | [JamesCameron] | [In, the, 22nd, century,, a, paraplegic, Marin... |
| 1 | 285 | Pirates of the Caribbean: At World's End | [Captain, Barbossa,, long, believed, to, be, d... | [Adventure, Fantasy, Action] | [ocean, drugabuse, exoticisland, eastindiatrad... | [JohnnyDepp, OrlandoBloom, KeiraKnightley] | [GoreVerbinski] | [Captain, Barbossa,, long, believed, to, be, d... |
| 2 | 206647 | Spectre | [A, cryptic, message, from, Bond's, past, send... | [Action, Adventure, Crime] | [spy, basedonnovel, secretagent, sequel, mi6, ... | [DanielCraig, ChristophWaltz, LéaSeydoux] | [SamMendes] | [A, cryptic, message, from, Bond's, past, send... |
| 3 | 49026 | The Dark Knight Rises | [Following, the, death, of, District, Attorney... | [Action, Crime, Drama, Thriller] | [dcomics, crimefighter, terrorist, secretiden... | [ChristianBale, MichaelCaine, GaryOldman] | [ChristopherNolan] | [Following, the, death, of, District, Attorney... |
| 4 | 49529 | John Carter | [John, Carter, is, a, war-weary,, former, mili... | [Action, Adventure, ScienceFiction] | [basedonnovel, mars, medallion, spacetravel, p... | [TaylorKitsch, LynnCollins, SamanthaMorton] | [AndrewStanton] | [John, Carter, is, a, war-weary,, former, mili... |

In [31]:

```
new_df = movies[['movie_id','title','tags']]
new_df['tags'] = new_df['tags'].apply(lambda x: " ".join(x))
```

C:\Users\prince kumar dev\AppData\Local\Temp\ipykernel_11116\2001000471.py:2: 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 (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
new_df['tags'] = new_df['tags'].apply(lambda x: " ".join(x))

In [32]:

```
new_df.head(2)
```

Out[32]:

| | movie_id | title | tags |
|---|----------|--|---|
| 0 | 19995 | Avatar | In the 22nd century, a paraplegic Marine is di... |
| 1 | 285 | Pirates of the Caribbean: At World's End | Captain Barbossa, long believed to be dead, ha... |

In [33]:

```
new_df['tags'] = new_df['tags'].apply(lambda x: x.lower())
```

C:\Users\prince kumar dev\AppData\Local\Temp\ipykernel_11116\1812721160.py:1: 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 (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
new_df['tags'] = new_df['tags'].apply(lambda x: x.lower())

In [34]:

```
new_df['tags'][0]
```

Out[34]:

'in the 22nd century, a paraplegic marine is dispatched to the moon pandora on a unique mission, but becomes torn between following orders and protecting an alien civilization. action adventure fantasy sciencefiction cultureclash future spacewar spacecolony society spacetravel futuristic romance space alien tribe alienplanet cgi marine soldier battle loveaffair antiwar powerrelations mindandsoul 3d samworthington zoesaldana sigourneyweaver jamescameron'

In [42]:

```
import nltk
```

In [43]:

```
from nltk.stem.porter import PorterStemmer  
ps = PorterStemmer()
```

In [44]:

```
def stem(text):  
    y=[]  
    for i in text.split():  
        y.append(ps.stem(i))  
    return ' '.join(y)
```

In [45]:

```
new_df['tags'] = new_df['tags'].apply(stem)
```

C:\Users\prince kumar dev\AppData\Local\Temp\ipykernel_11116\3213734980.py:1: 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 (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
new_df['tags'] = new_df['tags'].apply(stem)

In [46]:

```
new_df.head(2)
```

Out[46]:

| | movie_id | title | tags |
|---|----------|--|--|
| 0 | 19995 | Avatar | inthe22ndcentury,aparaplegmarinisdispatchtothe... |
| 1 | 285 | Pirates of the Caribbean: At World's End | captainbarbossa,longbelievetobedead,hacomebackt... |

In [47]:

```
from sklearn.feature_extraction.text import CountVectorizer  
cv = CountVectorizer(max_features=5000, stop_words='english')
```

In [48]:

```
vectors = cv.fit_transform(new_df['tags']).toarray()
```

In [49]:

```
vectors
```

Out[49]:

```
array([[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, 0],  
       [0, 0, 0, ..., 0, 0, 0]], dtype=int64)
```

In [52]:

```
cv.get_feature_names()

'noteventheromanempire',
'noteveryonintheworldisasniceasbabibink',
'noteveryonistherethroughchoice',
'noteveryonseemtoagreincludherboss',
'noteveryonshescrewoverissoquicktoforgivandforget',
'noteveryonwelcomthenewaddittothetroupasjackrabbitandmaggimousplottoevictthunder',
'noteveryonwillmakeitoutalive',
'notexactlithrilled',
'notfinished',
'notgetanisexfromhiwife',
'notgotobeeasibecausecaponhathepolicinhipocket',
'notgotobeeasiforhimtostayoutoftrouble',
'nothbutchancandheavifogcouldbringhimitothicursplace',
'nothbutwaterformiles',
'nothcanstopcasey',
'nothgetbetweenannmariandherboard',
'nothgoeasplanandbangkokistheperfectsetsetforanothadventurwiththerowdigroup',
'nothiownkids',
'nothlefttodo',
'nothmoveinthenavaglandscansavetheashonthewindandwater'
```

In [53]:

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

In [55]:

```
cosine_similarity(vectors).shape
```

Out[55]:

```
(4806, 4806)
```

In [56]:

```
similarity = cosine_similarity(vectors)
```

In [71]:

```
def recommend(movie):
    movie_index = new_df[new_df['title'] == movie].index[0]
    distances = similarity[movie_index]

    movies_list = sorted(list(enumerate(distances)),reverse =True ,key=lambda x:x[1])[1:6]

    for i in movies_list:
        print(new_df.iloc[i[0]].title)
```

In [72]:

```
recommend('Batman Begins')
```

```
Batman v Superman: Dawn of Justice
Avatar
Pirates of the Caribbean: At World's End
Spectre
The Dark Knight Rises
```

In [79]:

```
recommend('Pink Narcissus')
```

```
Man on a Ledge
Reindeer Games
Mirrors
The November Man
Machete
```

In [76]:

```
recommend('Spider-Man')
```

```
Pirates of the Caribbean: At World's End
Spectre
The Dark Knight Rises
John Carter
Spider-Man 3
```