In [28]:
```python
import pandas as pd
import numpy as np
```

In [29]:
```python
movies = pd.read_csv('tmdb_5000_movies.csv')
credits = pd.read_csv('tmdb_5000_credits.csv')
```

In [30]:
```python
movies.head(1)
```

Out[30]:

| | budget | genres | homepage | id | keywords | original_language | original_title | overview | popularity | production_ |
|---|---|---|---|---|---|---|---|---|---|---|
| **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 Film Pa |

In [31]:
```python
credits.head(1)
```

Out[31]:

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

In [33]:
```python
movies = movies.merge(credits, on='title')
```

```
In [34]: movies.head(1)
```

Out[34]:

| | budget | genres | homepage | id | keywords | original_language | original_title | overview | popularity | production_ |
|---|---|---|---|---|---|---|---|---|---|---|
| **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 Film P |

1 rows × 23 columns

```
In [35]: movies=movies[['movie_id' , 'title' , 'overview' , 'genres' ,'keywords','cast','crew']]
```

```
In [36]: movies.head(1)
```

Out[36]:

| | 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... |

```
In [37]: movies.isnull().sum()
```

```
Out[37]:
movie_id    0
title       0
overview    3
genres      0
keywords    0
cast        0
crew        0
dtype: int64
```

In [38]:
```
movies.dropna(inplace=True)
```

```
C:\Users\ramas\AppData\Local\Temp\ipykernel_21148\3786870272.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

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#return
ing-a-view-versus-a-copy)
  movies.dropna(inplace=True)
```

In [39]:
```
movies.isnull().sum()
```

Out[39]:
```
movie_id    0
title       0
overview    0
genres      0
keywords    0
cast        0
crew        0
dtype: int64
```

In [46]:
```
import ast
ast.literal_eval('[{"id": 28, "name": "Action"}, {"id": 12, "name": "Adventure"}, {"id": 14, "name": "Fantasy
)
```

Out[46]:
```
[{'id': 28, 'name': 'Action'},
 {'id': 12, 'name': 'Adventure'},
 {'id': 14, 'name': 'Fantasy'},
 {'id': 878, 'name': 'Science Fiction'}]
```

In [45]:
```
movies['genres'][0]
```

Out[45]:
```
'[{"id": 28, "name": "Action"}, {"id": 12, "name": "Adventure"}, {"id": 14, "name": "Fantasy"}, {"id": 878,
"name": "Science Fiction"}]'
```

In [49]:
```
def convert(obj):
    l=[]
    for i  in ast.literal_eval(obj):
        l.append(i['name'])
    return l
```

In [50]:
```
movies['genres'] = movies['genres'].apply(convert)
```

In [51]:
```
movies.head(2)
```

Out[51]:

| | 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] | [{"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... | [Adventure, Fantasy, Action] | [{"id": 270, "name": "ocean"}, {"id": 726, "na... | [{"cast_id": 4, "character": "Captain Jack Spa... | [{"credit_id": "52fe4232c3a36847f800b579", "de... |

In [ ]:
```
def convert(obj):
    l=[]
    for i  in ast.literal_eval(obj):
        l.append(i['name'])
    return l
```

In [52]:
```
movies['keywords'] = movies['keywords'].apply(convert)
```

In [54]:
```
movies.head(1)
```

Out[54]:

| | 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... |

In [55]:
```
def convertor3(obj):
    l=[]
    counter=0
    for i in ast.literal_eval(obj):
        if counter !=3:
            l.append(i['name'])
        else:
            break
    return l
```

In [56]:
```
movies['cast'] = movies['cast'].apply(convertor3)
```

In [57]:
```
movies.head(1)
```

Out[57]:

| | 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... | [Sam Worthington, Zoe Saldana, Sigourney Weave... | [{"credit_id": "52fe48009251416c750aca23", "de... |

In [60]:
```
def fetch_director(obj):
    l=[]
    for i in ast.literal_eval(obj):
        if i['job'] =='Director':
            l.append(i['name'])
            break
    return l
```

In [61]:
```
movies['crew'] = movies['crew'].apply(fetch_director)
```

In [62]:
```
movies.head(1)
```

Out[62]:

| | 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... | [Sam Worthington, Zoe Saldana, Sigourney Weave... | [James Cameron] |

```
In [65]: movies['overview']=movies['overview'].apply( lambda x:x.split())
```

```
In [66]: movies.head(1)
```

Out[66]:

| | 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 Weave... | [James Cameron] |

```
In [71]: movies['genres']=movies['genres'].apply(lambda x:[i.replace(" ","") for i in x])
```

```
In [72]: 
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['crew']=movies['crew'].apply(lambda x:[i.replace(" ","") for i in x])
```

```
In [73]: movies.head(1)
```

Out[73]:

| | 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] |

```
In [ ]: 
```

In [74]:
```
movies['tags'] = movies['overview'] + movies['genres'] +movies['keywords'] + movies['cast']+ movies['crew']
```

In [76]:
```
movies['tags'][0]
```

Out[76]:
```
['In',
 'the',
 '22nd',
 'century,',
 'a',
 'paraplegic',
 'Marine',
 'is',
 'dispatched',
 'to',
 'the',
 'moon',
 'Pandora',
 'on',
 'a',
 'unique',
 'mission,',
 'but',
 'becomes',
```

In [77]:
```
new_df = movies[['movie_id','title','tags']]
```

In [78]:
```
new_df.head()
```

Out[78]:

| | movie_id | title | tags |
|---|---|---|---|
| **0** | 19995 | Avatar | [In, the, 22nd, century,, a, paraplegic, Marin... |
| **1** | 285 | Pirates of the Caribbean: At World's End | [Captain, Barbossa,, long, believed, to, be, d... |
| **2** | 206647 | Spectre | [A, cryptic, message, from, Bond’s, past, send... |
| **3** | 49026 | The Dark Knight Rises | [Following, the, death, of, District, Attorney... |
| **4** | 49529 | John Carter | [John, Carter, is, a, war-weary,, former, mili... |

In [80]:
```
new_df['tags']=new_df['tags'].apply(lambda x:" ".join(x))
```

```
C:\Users\ramas\AppData\Local\Temp\ipykernel_21148\487797088.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#return
ing-a-view-versus-a-copy)
  new_df['tags']=new_df['tags'].apply(lambda x:" ".join(x))
```

In [82]:
```
new_df.head()
```

Out[82]:

| | 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... |
| **2** | 206647 | Spectre | A cryptic message from Bond's past sends him o... |
| **3** | 49026 | The Dark Knight Rises | Following the death of District Attorney Harve... |
| **4** | 49529 | John Carter | John Carter is a war-weary, former military ca... |

In [85]:
```
new_df['tags']=new_df['tags'].apply(lambda x:x.lower())
```

```
C:\Users\ramas\AppData\Local\Temp\ipykernel_21148\4224080999.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#return
ing-a-view-versus-a-copy)
  new_df['tags']=new_df['tags'].apply(lambda x:x.lower())
```

In [86]:
```
new_df.head()
```

Out[86]:

| | 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... |
| **2** | 206647 | Spectre | a cryptic message from bond’s past sends him o... |
| **3** | 49026 | The Dark Knight Rises | following the death of district attorney harve... |
| **4** | 49529 | John Carter | john carter is a war-weary, former military ca... |

In [120]:
```
from sklearn.feature_extraction.text import CountVectorizer
```

In [121]:
```
cv = CountVectorizer(max_features=5000,stop_words='english')
```

In [122]:
```
vector =cv.fit_transform(new_df['tags']).toarray()
```

In [125]:
```
vector
```

Out[125]:
```
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 [130]:
```
cv.get_feature_names_out()
```

Out[130]:
```
array(['000', '007', '10', ..., 'zoo', 'zooeydeschanel', 'zoëkravitz'],
      dtype=object)
```

In [108]:
```
import nltk
```

In [109]:
```
from nltk.stem.porter import PorterStemmer
```

In [111]:
```
ps= PorterStemmer()
```

In [116]:
```
def stem(text):
    y =[]

    for i in text.split():
        y.append(ps.stem(i))
    return " ".join(y)
```

In [117]:
```
ps.stem('loves')
```

Out[117]:
```
'love'
```

In [119]:
```
 new_df['tags']=new_df['tags'].apply(stem)
```

```
C:\Users\ramas\AppData\Local\Temp\ipykernel_21148\1865199325.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#return
ing-a-view-versus-a-copy)
  new_df['tags']=new_df['tags'].apply(stem)
```

In [ ]:

In [131]:
```
from sklearn.metrics.pairwise import cosine_similarity
```

In [134]:
```
similarity = cosine_similarity(vector)
```

```
In [140]: similarity[1]

Out[140]: array([0.06818182, 1.        , 0.07644708, ..., 0.02153652, 0.        ,
                 0.02383656])
```

```
In [155]: def recommend(movies):
              movie_index =new_df[new_df['title']=='Batman Begins'].index[0]
              distance = similarity[movie_index]
              movie_list =sorted(list(enumerate(distance)),reverse=True ,key=lambda x:x[1])[1:6]
              for i in movie_list:
                  print(i[0])
```

```
In [157]: recommend('Avarat') # giving the nearest movies name

          65
          1363
          3
          1362
          4148
```

```
In [ ]:
```

```
In [146]: new_df[new_df['title']=='Batman Begins'].index[0]

Out[146]: 119
```

```
In [166]: sorted(list(enumerate(similarity[0])),reverse=True ,key=lambda x:x[1])
```

```
Out[166]: [(0, 0.9999999999999998),
           (1216, 0.27028123880866767),
           (582, 0.23162743094465488),
           (2333, 0.22996655275195),
           (3730, 0.22498852128662872),
           (507, 0.22438727760202976),
           (539, 0.22305671869347438),
           (1920, 0.22010219462003333),
           (2409, 0.2175970699446223),
           (4048, 0.20751433915982237),
           (2786, 0.20695933859617893),
           (322, 0.20691022044226628),
           (577, 0.20100756305184242),
           (778, 0.19968076595771794),
           (1204, 0.19790977371009408),
           (61, 0.19596237883454903),
           (2971, 0.19462473604038075),
           (74, 0.1938287215142766),
           (151, 0.1928473039599675),
```

```
In [2]: def recommend(movies):
            movie_index =new_df[new_df['title']=='Batman Begins'].index[0]
            distance = similarity[movie_index]
            movie_list =sorted(list(enumerate(distance)),reverse=True ,key=lambda x:x[1])[1:6]
            for i in movie_list:
                print(new_df.iloc[i[0]].title)
```

```
In [3]: recommend('Batman Begins')
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[3], line 1
----> 1 recommend('Batman Begins')

Cell In[2], line 2, in recommend(movies)
      1 def recommend(movies):
----> 2     movie_index =new_df[new_df['title']=='Batman Begins'].index[0]
      3     distance = similarity[movie_index]
      4     movie_list =sorted(list(enumerate(distance)),reverse=True ,key=lambda x:x[1])[1:6]

NameError: name 'new_df' is not defined
```

```
In [167]: new_df.iloc[1363].title
```

```
Out[167]: 'Batman'
```

```
In [168]: import pickle
```

```
In [169]: pickle.dump(new_df , open('movies.pkl','wb')) # not send
```

```
In [170]: pickle.dump(new_df.to_dict(),open('movie_dict.pkl','wb'))
```

```
In [171]: pickle.dump(similarity ,open('similarity.pkl','wb'))
```

```
In [ ]:
```