Importing The Required Libraries

In [1]:
```
import warnings
warnings.simplefilter(action='ignore', category=FutureWarning)
warnings.simplefilter(action='ignore', category=UserWarning)
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
```

Importing the Datasets: We have 2 sets of datasets which we will merge to form one dataset on which we will work

In [2]:
```
credit_df = pd.read_csv(r'D:\Dataset for ML\credits.csv')
movie_df = pd.read_csv(r'D:\Dataset for ML\movies.csv')
```

In [3]:
```
credit_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]:
```
1 movie_df.head()
```

Out[4]:

	budget	genres	homepage	id	keywords	original_language	original_title	overview	pop
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...	150.4
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.(
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...	107.3
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...	112.3
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...	43.9

In [5]:
```
# Checking the Info and datatypes of the features
credit_df.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 [6]:
```
movie_df.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 
```

In [7]:
```
# Checking for the null values in the datasets
credit_df.isnull().sum()
```

Out[7]:
```
movie_id    0
title       0
cast        0
crew        0
dtype: int64
```

In [8]:
```
movie_df.isnull().sum()
```

Out[8]:
```
budget                     0
genres                     0
homepage                3091
id                         0
keywords                   0
original_language          0
original_title             0
overview                   3
popularity                 0
production_companies       0
production_countries       0
release_date               1
revenue                    0
runtime                    2
spoken_languages           0
status                     0
tagline                  844
title                      0
vote_average               0
vote_count                 0
dtype: int64
```

In [9]:
```
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)
```

In [10]:
```
1 credit_df
```

Out[10]:

	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...
5	559	Spider-Man 3	[{"cast_id": 30, "character": "Peter Parker / ...	[{"credit_id": "52fe4252c3a36847f80151a5", "de...
6	38757	Tangled	[{"cast_id": 34, "character": "Flynn Rider (vo...	[{"credit_id": "52fe46db9251416c91062101", "de...

In [11]:
```
1 movie_df
```

Out[11]:

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

The code uses Pandas' merge function to combine two DataFrames, movie_df and credit_df, based on the 'title' column, resulting in an updated movie_df with combined information from both DataFrames.

In [12]:
```
1 movie_df = movie_df.merge(credit_df, on = 'title')
```

In [13]:
```
1 movie_df
```

Out[13]:

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

In [14]:
```
1 # Let's check the shape of the dataset
2 movie_df.shape
```

Out[14]: (4808, 23)

This piece of code extracts columns ('movie_id' 'title' 'overview' 'genres 'keywords 'cast' 'crew'), from the DataFrame. By doing it filters the DataFrame to include these specific columns. The outcome is then reassigned back, to the variable.

In [15]:
```
movie_df = movie_df[['movie_id', 'title', 'overview', 'genres', 'keywords', 'cast', 'crew']]
movie_df
```

Out[15]:

	movie_id	title	overview	genres	keywords	cast	
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", "...	["52fe48009251416c
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...	["52fe4232c3a36847
2	206647	Spectre	A cryptic message from Bond’s past sends him o...	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	[{"id": 470, "name": "spy"}, {"id": 818, "name...	[{"cast_id": 1, "character": "James Bond", "cr...	["54805967c3a36829
3	49026	The Dark Knight Rises	Following the death of District Attorney Harve...	[{"id": 28, "name": "Action"}, {"id": 80, "nam...	[{"id": 849, "name": "dc comics"}, {"id": 853,...	[{"cast_id": 2, "character": "Bruce Wayne / Ba...	["52fe4781c3a3684

In [16]:
```
movie_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 4808 entries, 0 to 4807
Data columns (total 7 columns):
 #   Column    Non-Null Count  Dtype 
---  ------    --------------  ----- 
 0   movie_id  4808 non-null   int64 
 1   title     4808 non-null   object
 2   overview  4805 non-null   object
 3   genres    4808 non-null   object
 4   keywords  4808 non-null   object
 5   cast      4808 non-null   object
 6   crew      4808 non-null   object
dtypes: int64(1), object(6)
memory usage: 300.5+ KB
```

In [17]:
```
1 movie_df.isnull().sum()
```

Out[17]:
```
movie_id    0
title       0
overview    3
genres      0
keywords    0
cast        0
crew        0
dtype: int64
```

This code eliminates any rows, in the DataFrame movie_df that contain missing values (NaN). The changes are made directly to the DataFrame itself so there is no requirement, for reassigning it.

In [18]:
```
1 movie_df.dropna(inplace= True)
```

```
C:\Users\sweet\AppData\Local\Temp\ipykernel_8788\3108332569.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#returning-a-view-versus-a-copy)
  movie_df.dropna(inplace= True)
```

In [19]:
```
1 movie_df.duplicated().sum()
```

Out[19]:
```
0
```

#To retrieve the value, in the 'genres column for the row (index 0) of the DataFrame `movie_df` we are using the code `movie_df.iloc[0].genres` . #This assumes that the 'genres column contains data that can be iterated upon, like a list or a string representing genres.

In [20]:
```
1 movie_df.iloc[0].genres
```

Out[20]:
```
'[{"id": 28, "name": "Action"}, {"id": 12, "name": "Adventure"}, {"id": 14, "name": "Fantasy"}, {"id": 878, "name": "Science Fiction"}]'
```

The `import ast` statement is used to import Abstract Syntax Trees modules into Python. It is usually used where real words in a string form have to be carefully evaluated. For our previous code, it can be used to convert string representations of characters (e.g., " [action, drama]") in the 'genres' column to actual Python characters using `ast. literal_eval` .

In [21]:
```
1 import ast
```

The function "convert" transforms a string representation of a list of dictionaries, where each dictionary has a 'name' key, into a list of names. It uses ast.literal_eval to safely evaluate the string as a list and then extracts the 'name' values, creating a more readable list of names from the original data.

In [22]:
```
1 def convert(obj):
2     L= []
3     for i in ast.literal_eval(obj):
4         L.append(i['name'])
5     return L
```

In [23]:
```
movie_df['genres']= movie_df['genres'].apply(convert)
movie_df['keywords'] = movie_df['keywords'].apply(convert)
movie_df.head()
```

```
C:\Users\sweet\AppData\Local\Temp\ipykernel_8788\2475596741.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)
  movie_df['genres']= movie_df['genres'].apply(convert)
C:\Users\sweet\AppData\Local\Temp\ipykernel_8788\2475596741.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)
  movie_df['keywords'] = movie_df['keywords'].apply(convert)
```

Out[23]:

	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...

The function convert3 processes a string representation of a list of dictionaries, extracting the names from the first three dictionaries and returning them in a list. The loop stops after encountering the third dictionary.

In [24]:

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

In [25]:
```
movie_df['cast'] = movie_df['cast'].apply(convert3)
movie_df.head()
```

```
C:\Users\sweet\AppData\Local\Temp\ipykernel_8788\4156296963.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)
  movie_df['cast'] = movie_df['cast'].apply(convert3)
```

Out[25]:

	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]	[{"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 ...	[Johnny Depp]	[{"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...	[Daniel Craig]	[{"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...	[Christian Bale]	[{"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...	[Taylor Kitsch]	[{"credit_id": "52fe479ac3a36847f813eaa3", "de...

The fetch_director function extracts the director's name from a list of dictionaries in string representation, where each dictionary contains information about the employee's job and name It returns through the list, adding the director's name to the line about when the task is 'guided', and stops the iteration. The program returns a list of directory names.

In [26]:
```
def fetch_director(obj):
    L = []
    for i in ast.literal_eval(obj):
        if i['job'] == 'Director':
            L.append(i['name'])
            break
    return L
```

In [27]:
```
movie_df['crew'] = movie_df['crew'].apply(fetch_director)
movie_df
```

```
C:\Users\sweet\AppData\Local\Temp\ipykernel_8788\4260810410.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)
  movie_df['crew'] = movie_df['crew'].apply(fetch_director)
```

Out[27]:

	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]	[James Cameron]
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 ...	[Johnny Depp]	[Gore Verbinski]
			A cryptic message from	[Action,	[spy, based on novel,		

The code movie_df['overview'][0] retrieves the content of the 'overview' column for the first row (index 0) in the DataFrame movie_df. This provides the overview or summary of the first movie in the dataset.

In [28]:
```
movie_df['overview'][0]
```

Out[28]:
```
'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.'
```

This code takes each entry in the 'overview' column of the DataFrame movie_df and applies a lambda function. The lambda function uses the split() method to split the string into a list of words. The result is then assigned back to the 'overview' column, transforming each overview into a list of words.

In [29]:
```
1 movie_df['overview'] = movie_df['overview'].apply(lambda x:x.split())
```

```
C:\Users\sweet\AppData\Local\Temp\ipykernel_8788\599775510.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)
  movie_df['overview'] = movie_df['overview'].apply(lambda x:x.split())
```

Take a look at overview column

In [30]:
```
1 movie_df.head()
```

Out[30]:

	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]	[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]	[Gore Verbinski]
2	206647	Spectre	[A, cryptic, message, from, Bond's, past, send...	[Action, Adventure, Crime]	[spy, based on novel, secret agent, sequel, mi...	[Daniel Craig]	[Sam Mendes]
3	49026	The Dark Knight Rises	[Following, the, death, of, District, Attorney...	[Action, Crime, Drama, Thriller]	[dc comics, crime fighter, terrorist, secret i...	[Christian Bale]	[Christopher Nolan]
4	49529	John Carter	[John, Carter, is, a, war-weary,, former, mili...	[Action, Adventure, Science Fiction]	[based on novel, mars, medallion, space travel...	[Taylor Kitsch]	[Andrew Stanton]

In [31]:

```
## Remove spaces from strings in the 'genres' column
movie_df['genres'] = movie_df['genres'].apply(lambda x: [i.replace(" ", "") for i in x])

# Remove spaces from strings in the 'keywords' column
movie_df['keywords'] = movie_df['keywords'].apply(lambda x: [i.replace(" ", "") for i in x])

# Check for None values in 'cast' column
movie_df['cast'] = movie_df['cast'].apply(lambda x: [i.replace(" ", "") for i in x] if x is not None else

# Check for None values in 'crew' column
movie_df['crew'] = movie_df['crew'].apply(lambda x: [i.replace(" ", "") for i in x] if x is not None else

```

```
C:\Users\sweet\AppData\Local\Temp\ipykernel_8788\4052144616.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)
  movie_df['genres'] = movie_df['genres'].apply(lambda x: [i.replace(" ", "") for i in x])
C:\Users\sweet\AppData\Local\Temp\ipykernel_8788\4052144616.py:5: 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)
  movie_df['keywords'] = movie_df['keywords'].apply(lambda x: [i.replace(" ", "") for i in x])
C:\Users\sweet\AppData\Local\Temp\ipykernel_8788\4052144616.py:8: 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)
  movie_df['cast'] = movie_df['cast'].apply(lambda x: [i.replace(" ", "") for i in x] if x is not None else
[])
C:\Users\sweet\AppData\Local\Temp\ipykernel_8788\4052144616.py:11: 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)
  movie_df['crew'] = movie_df['crew'].apply(lambda x: [i.replace(" ", "") for i in x] if x is not None else
[])
```

Take a look at other feature column

In [32]:
```
1 movie_df.head()
```

Out[32]:

	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]	[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]	[GoreVerbinski]
2	206647	Spectre	[A, cryptic, message, from, Bond's, past, send...	[Action, Adventure, Crime]	[spy, basedonnovel, secretagent, sequel, mi6, ...	[DanielCraig]	[SamMendes]
3	49026	The Dark Knight Rises	[Following, the, death, of, District, Attorney...	[Action, Crime, Drama, Thriller]	[dccomics, crimefighter, terrorist, secretiden...	[ChristianBale]	[ChristopherNolan]
4	49529	John Carter	[John, Carter, is, a, war-weary,, former, mili...	[Action, Adventure, ScienceFiction]	[basedonnovel, mars, medallion, spacetravel, p...	[TaylorKitsch]	[AndrewStanton]

In [33]:
```
1 # Concatenate the 'overview', 'keywords', 'cast', and 'genres' columns to create a new 'tags' column
2 movie_df['tags'] = movie_df['overview'] + movie_df['keywords'] + movie_df['cast'] + movie_df['genres']
```

```
C:\Users\sweet\AppData\Local\Temp\ipykernel_8788\449457524.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)
  movie_df['tags'] = movie_df['overview'] + movie_df['keywords'] + movie_df['cast'] + movie_df['genres']
```

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

Out[34]:

	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]	[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]	[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]	[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]	[dccomics, crimefighter, terrorist, secretiden...	[ChristianBale]	[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]	[AndrewStanton]	[John, Carter, is, a, war-weary,, former, mili...

In [35]:
```
1 # Create a new DataFrame 'new_df' with selected columns
2 new_df = movie_df[['movie_id', 'title','tags']]
3 new_df.head()
```

Out[35]:

	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 [36]:
```
# Join the lists of tags into a single string for each row in the 'tags' column
new_df['tags'] = new_df['tags'].apply(lambda x: ' '.join(x))
```

```
C:\Users\sweet\AppData\Local\Temp\ipykernel_8788\4000229333.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 [37]:
```
new_df.head()
```

Out[37]:

	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 [38]:
```
new_df['tags'][0]
```

Out[38]:
```
'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. cultureclash future spacewar spacecolony society spacetravel futuristic romance space alien tribe alienplanet cgi marine soldier battle loveaffair antiwar powerrelations mindandsoul 3d SamWorthington Action Adventure Fantasy ScienceFiction'
```

In [39]:
```
# Convert all text in the 'tags' column to lowercase
new_df['tags'] = new_df['tags'].apply(lambda X:X.lower())
new_df.head()
```

```
C:\Users\sweet\AppData\Local\Temp\ipykernel_8788\2116046987.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:X.lower())
```

Out[39]:

	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 [40]:
```
from sklearn.feature_extraction.text import CountVectorizer

# Initialize the CountVectorizer with maximum features and stop words
cv = CountVectorizer(max_features = 5000, stop_words = 'english')

# Fit and transform the 'tags' column into a bag-of-words representation
cv.fit_transform(new_df['tags']).toarray().shape
```

Out[40]: (4805, 5000)

In [41]:
```
# Apply fit_transform to convert 'tags' column into a bag-of-words representation
vectors = cv.fit_transform(new_df['tags']).toarray()
```

In [42]:
```
vectors[0]
```

Out[42]: array([0, 0, 0, ..., 0, 0, 0], dtype=int64)

In [43]:
```
# Get feature names from the CountVectorizer
feature_names = cv.get_feature_names_out()

# Print the number of feature names (vocabulary size)
print(len(feature_names))
```

```
5000
```

In [44]:
```
import nltk
```

In [45]:
```
# Create an instance of the PorterStemmer
from nltk.stem.porter import PorterStemmer
ps = PorterStemmer()
```

In [46]:
```
def stem(text):
    y = []
    for i in text.split():
        y.append(ps.stem(i))
    return " ".join(y)
```

In [47]:
```
# Apply the stem function to each element in the 'tags' column
new_df['tags'] = new_df['tags'].apply(stem)
```

```
C:\Users\sweet\AppData\Local\Temp\ipykernel_8788\162247528.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(stem)
```

In [48]:
```
# Compute cosine similarity between rows (movies) in the 'vectors' matrix
from sklearn.metrics.pairwise import cosine_similarity
cosine_similarity(vectors)
```

Out[48]:
```
array([[1.        , 0.09416472, 0.06172134, ..., 0.02465568, 0.0270666 ,
        0.        ],
       [0.09416472, 1.        , 0.06780635, ..., 0.02708645, 0.        ,
        0.        ],
       [0.06172134, 0.06780635, 1.        , ..., 0.02663118, 0.        ,
        0.        ],
       ...,
       [0.02465568, 0.02708645, 0.02663118, ..., 1.        , 0.07007128,
        0.04671418],
       [0.0270666 , 0.        , 0.        , ..., 0.07007128, 1.        ,
        0.05128205],
       [0.        , 0.        , 0.        , ..., 0.04671418, 0.05128205,
        1.        ]])
```

In [49]:
```
# Get the shape of the cosine similarity matrix
cosine_similarity(vectors).shape
```

Out[49]:
```
(4805, 4805)
```

In [50]:
```
# Get the shape of the first row of the similarity matrix
similarity = cosine_similarity(vectors)
```

In [51]:
```
# Get the first row of the similarity matrix
similarity[0]
```

Out[51]:
```
array([1.        , 0.09416472, 0.06172134, ..., 0.02465568, 0.0270666 ,
       0.        ])
```

In [52]:
```
similarity[0].shape
```

Out[52]:
```
(4805,)
```

In [53]:
```
# Retrieve the similarity scores and movie indices for the first movie
first_movie_similarity = similarity[0]

# Enumerate pairs of movie indices and similarity scores
enumerated_similarity = list(enumerate(first_movie_similarity))

# Sort the pairs based on similarity scores in descending order
sorted_similarity = sorted(enumerated_similarity, reverse=True, key=lambda x: x[1])

# Retrieve the top 5 most similar movies (excluding the first one)
top_similar_movies = sorted_similarity[1:6]

# Display the result (optional)
print(top_similar_movies)
```

```
[(539, 0.26892643710023856), (1194, 0.264575131106459), (507, 0.25539990311691463), (1216, 0.248069469178416
9), (260, 0.24688535993934702)]
```

In [54]:
```
def recommend(movie):
    # Find the index of the specified movie
    movie_index = new_df[new_df['title'] == movie].index[0]

    # Get the similarity scores between the specified movie and all other movies
    distances = similarity[movie_index]

    # Retrieve the indices and similarity scores of the top 5 most similar movies
    movie_list = sorted(list(enumerate(distances)), reverse=True, key=lambda x: x[1])[1:6]

    # Print the titles of the recommended movies
    for i in movie_list:
        print(new_df.iloc[i[0]].title)
```

In [55]:
```
recommend('Avatar')
```

```
Titan A.E.
Small Soldiers
Independence Day
Aliens vs Predator: Requiem
Ender's Game
```

In [56]:
```
1 recommend('Iron Man')
```

```
Iron Man 2
Iron Man 3
Avengers: Age of Ultron
The Avengers
Ant-Man
```

In [57]:
```
1 recommend('Liar Liar')
```

```
13 Going on 30
Heartbreakers
A Simple Wish
A Good Year
21 & Over
```

In [58]:
```
1 recommend('The Dark Knight Rises')
```

```
The Dark Knight
Batman Begins
Batman
Batman Returns
Batman
```