# untitled

October 29, 2023

## 1 importing libraries

[1]:
```
import warnings
warnings.filterwarnings('ignore')
```

[2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

[3]:
```
df= pd.read_csv('movies.csv')
```

[4]:
```
# closest movies name
import difflib
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
```

## 2 Pre processing

[5]:
```
df.head()
```

[5]:
```
   index     budget                                     genres  \
0      0  237000000  Action Adventure Fantasy Science Fiction
1      1  300000000                  Adventure Fantasy Action
2      2  245000000                    Action Adventure Crime
3      3  250000000               Action Crime Drama Thriller
4      4  260000000          Action Adventure Science Fiction

                                       homepage      id  \
0                   http://www.avatarmovie.com/   19995
1  http://disney.go.com/disneypictures/pirates/     285
2   http://www.sonypictures.com/movies/spectre/  206647
3             http://www.thedarkknightrises.com/   49026
4          http://movies.disney.com/john-carter   49529

                                            keywords original_language  \
```

```
0  culture clash future space war space colony so…                en
1  ocean drug abuse exotic island east india trad…                en
2         spy based on novel secret agent sequel mi6                 en
3  dc comics crime fighter terrorist secret ident…                en
4  based on novel mars medallion space travel pri…                en

                           original_title  \
0                                  Avatar
1  Pirates of the Caribbean: At World's End
2                                 Spectre
3                   The Dark Knight Rises
4                             John Carter

                                        overview  popularity  … runtime  \
0  In the 22nd century, a paraplegic Marine is di…  150.437577  …   162.0
1  Captain Barbossa, long believed to be dead, ha…  139.082615  …   169.0
2  A cryptic message from Bond’s past sends him o…  107.376788  …   148.0
3  Following the death of District Attorney Harve…  112.312950  …   165.0
4  John Carter is a war-weary, former military ca…   43.926995  …   132.0

                                  spoken_languages    status  \
0  [{"iso_639_1": "en", "name": "English"}, {"iso…  Released
1          [{"iso_639_1": "en", "name": "English"}]  Released
2  [{"iso_639_1": "fr", "name": "Fran\u00e7ais"},…  Released
3          [{"iso_639_1": "en", "name": "English"}]  Released
4          [{"iso_639_1": "en", "name": "English"}]  Released

                                     tagline  \
0                   Enter the World of Pandora.
1  At the end of the world, the adventure begins.
2                         A Plan No One Escapes
3                               The Legend Ends
4           Lost in our world, found in another.

                                   title vote_average vote_count  \
0                                 Avatar          7.2      11800
1  Pirates of the Caribbean: At World's End          6.9       4500
2                                 Spectre          6.3       4466
3                   The Dark Knight Rises          7.6       9106
4                             John Carter          6.1       2124

                                               cast  \
0  Sam Worthington Zoe Saldana Sigourney Weaver S…
1  Johnny Depp Orlando Bloom Keira Knightley Stel…
2  Daniel Craig Christoph Waltz L\u00e9a Seydoux …
3  Christian Bale Michael Caine Gary Oldman Anne …
4  Taylor Kitsch Lynn Collins Samantha Morton Wil…
```

```
                                                  crew           director
0  [{'name': 'Stephen E. Rivkin', 'gender': 0, 'd…      James Cameron
1  [{'name': 'Dariusz Wolski', 'gender': 2, 'depa…     Gore Verbinski
2  [{'name': 'Thomas Newman', 'gender': 2, 'depar…         Sam Mendes
3  [{'name': 'Hans Zimmer', 'gender': 2, 'departm…  Christopher Nolan
4  [{'name': 'Andrew Stanton', 'gender': 2, 'depa…     Andrew Stanton

[5 rows x 24 columns]
```

[6]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4803 entries, 0 to 4802
Data columns (total 24 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   index                 4803 non-null   int64
 1   budget                4803 non-null   int64
 2   genres                4775 non-null   object
 3   homepage              1712 non-null   object
 4   id                    4803 non-null   int64
 5   keywords              4391 non-null   object
 6   original_language     4803 non-null   object
 7   original_title        4803 non-null   object
 8   overview              4800 non-null   object
 9   popularity            4803 non-null   float64
 10  production_companies  4803 non-null   object
 11  production_countries  4803 non-null   object
 12  release_date          4802 non-null   object
 13  revenue               4803 non-null   int64
 14  runtime               4801 non-null   float64
 15  spoken_languages      4803 non-null   object
 16  status                4803 non-null   object
 17  tagline               3959 non-null   object
 18  title                 4803 non-null   object
 19  vote_average          4803 non-null   float64
 20  vote_count            4803 non-null   int64
 21  cast                  4760 non-null   object
 22  crew                  4803 non-null   object
 23  director              4773 non-null   object
dtypes: float64(3), int64(5), object(16)
memory usage: 900.7+ KB
```

[7]:
```
df.isnull().sum()
```

[7]:
```
index                     0
budget                    0
genres                   28
homepage               3091
id                        0
keywords                412
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
cast                     43
crew                      0
director                 30
dtype: int64
```

[8]:
```
df.shape
```

[8]:
```
(4803, 24)
```

[9]:
```
selected_features = ['genres','keywords','overview','director','tagline','cast']

print(selected_features)
```

```
['genres', 'keywords', 'overview', 'director', 'tagline', 'cast']
```

[10]:
```
# replacing the null values with null string

for feature in selected_features:
    df[feature]=df[feature].fillna('')
```

[11]:
```
df[selected_features].shape
```

[11]:
```
(4803, 6)
```

[12]:
```
# combining all the 5 features columns
```

```
combined_features = df['genres']+' '+df['keywords']+' '+df['overview']+'␣
↪'+df['director']+' '+df['tagline']+' '+df['cast']
```

[13]:
```
# print(combined_features)
```

[14]:
```
# converting the text data to features vectors
vectorizer = TfidfVectorizer()
```

[15]:
```
feature_vectors = vectorizer.fit_transform(combined_features)
```

[16]:
```
# print(feature_vectors)
```

## 3 cosine similarity

[17]:
```
# getting the similarity scores using cosine similarity

similarity = cosine_similarity(feature_vectors)
```

[18]:
```
# print(similarity)
```

[19]:
```
print(similarity.shape)
```

```
(4803, 4803)
```

[20]:
```
# getting the movie name from the user

movie_name  = input('Enter the fav Movie name: ')
```

```
Enter the fav Movie name: ironman
```

[21]:
```
# creating a list with all the movie names given in then dataset
list_of_all_movies = df['title'].tolist()

# print(list_of_all_movies)
```

[22]:
```
# finding the close match for the movie name given by the user

find_close_match = difflib.get_close_matches(movie_name,list_of_all_movies)

print(find_close_match)
```

```
['Birdman', 'Iron Man', 'Hitman']
```

[23]:
```
close_match = find_close_match[0]
print(close_match)
```

```
Birdman
```

[24]:
```
# finding the index of the movie title

index_of_the_movie = df[df.title==close_match]['index'].values[0]
print(index_of_the_movie)
```

2334

[25]:
```
# getting a list of similar movies

similarity_score = list(enumerate(similarity[index_of_the_movie]))

# print(similarity_score)
```

[26]:
```
len(similarity_score)
```

[26]: 4803

[27]:
```
# sorting the movie based on their similarity score

sorted_similar_score = sorted(similarity_score,key = lambda x:x[1],reverse=True)

# print(sorted_similar_score)
```

[28]:
```
# print the name of similar movies based on the index

print('Movies suggested for you : \n')

i=1

for movie in sorted_similar_score:
    index=movie[0]
    title_from_index = df[df.index==index]['title'].values[0]
    if (i<20):
        print(i,'.',title_from_index)
        i+=1
```

```
Movies suggested for you :

1 . Birdman
2 . The Revenant
3 . 21 Grams
4 . Babel
5 . Synecdoche, New York
6 . Biutiful
7 . Amores perros
8 . Fair Game
9 . Quinceañera
10 . Keeping the Faith
```

```
11 . Griff the Invisible
12 . House Party 2
13 . Last I Heard
14 . The Specials
15 . The Brothers McMullen
16 . Death to Smoochy
17 . Wish I Was Here
18 . Special
19 . Private Benjamin
```

[29]:
```
movie_name  = input('Enter the fav Movie name: ')

list_of_all_movies = df['title'].tolist()

find_close_match = difflib.get_close_matches(movie_name,list_of_all_movies)

close_match = find_close_match[0]

index_of_the_movie = df[df.title==close_match]['index'].values[0]

similarity_score = list(enumerate(similarity[index_of_the_movie]))

sorted_similar_score = sorted(similarity_score,key = lambda x:x[1],reverse=True)

print('Movies suggested for you : \n')

i=1

for movie in sorted_similar_score:
    index=movie[0]
    title_from_index = df[df.index==index]['title'].values[0]
    if (i<21):
        print(i,'.',title_from_index)
        i+=1
```

```
Enter the fav Movie name: x-men
Movies suggested for you :

1 . X-Men
2 . X2
3 . X-Men: The Last Stand
4 . X-Men: Days of Future Past
5 . X-Men: Apocalypse
6 . The Wolverine
7 . X-Men Origins: Wolverine
8 . X-Men: First Class
9 . Iron Man 2
10 . The Avengers
```

```
11 . Man of Steel
12 . The Incredible Hulk
13 . Avengers: Age of Ultron
14 . The Shadow
15 . Kick-Ass
16 . Deadpool
17 . Ant-Man
18 . Captain America: Civil War
19 . Thor: The Dark World
20 . Superhero Movie
```

[ ]: