```
import numpy as np
import pandas as pd
```

```
chunk_iter = pd.read_csv('movies.csv', chunksize=1000)

data = pd.concat(chunk_iter)
```

```
data.shape
```

```
(4803, 24)
```

```
selected_features = ['genres','keywords','tagline','cast','director']
print(selected_features)
```

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

```
for feature in selected_features:
  print(data[feature].isnull().sum())
```

```
0
0
0
0
0
```

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

```
comb_features = data['genres']+' '+data['keywords']+' '+data['tagline']+' '+data['cast']+' '+data['director']
```

```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
vectorizer = TfidfVectorizer()
```

```
feature_vectors = vectorizer.fit_transform(comb_features)
print(feature_vectors)
```

```
<Compressed Sparse Row sparse matrix of dtype 'float64'
        with 124266 stored elements and shape (4803, 17318)>
  Coords        Values
  (0, 201)      0.07860022416510505
  (0, 274)      0.09021200873707368
  (0, 5274)     0.11108562744414445
  (0, 13599)    0.1036413987316636
  (0, 5437)     0.1036413987316636
  (0, 3678)     0.21392179219912877
  (0, 3065)     0.22208377802661425
  (0, 5836)     0.1646750903586285
  (0, 14378)    0.33962752210959823
  (0, 16587)    0.12549432354918996
  (0, 3225)     0.24960162956997736
  (0, 14271)    0.21392179219912877
  (0, 4945)     0.24025852494110758
  (0, 15261)    0.07095833561276566
  (0, 16998)    0.1282126322850579
  (0, 11192)    0.09049319826481456
  (0, 11503)    0.27211310056983656
  (0, 13349)    0.15021264094167086
  (0, 17007)    0.23643326319898797
  (0, 17290)    0.20197912553916567
  (0, 13319)    0.2177470539412484
  (0, 14064)    0.20596090415084142
  (0, 16668)    0.19843263965100372
  (0, 14608)    0.15150672398763912
  (0, 8756)     0.22709015857011816
  :     :
  (4801, 403)   0.17727585190343229
  (4801, 4835)  0.24713765026964
  (4801, 17266) 0.28860981849329476
  (4801, 13835) 0.27870029291200094
  (4801, 13175) 0.28860981849329476
  (4801, 17150) 0.3025765103586468
  (4801, 3511)  0.3025765103586468
  (4801, 13948) 0.3025765103586468
  (4801, 7269)  0.3025765103586468
```

```
  (4802, 11161) 0.17867407682173203
  (4802, 4518)  0.16784466610624255
  (4802, 2129)  0.3099656128577656
  (4802, 4980)  0.16078053641367315
  (4802, 6155)  0.18056463596934083
  (4802, 3436)  0.21753405888348784
  (4802, 4528)  0.19504460807622875
  (4802, 1316)  0.1960747079005741
  (4802, 12989) 0.1696476532191718
  (4802, 4371)  0.1538239182675544
  (4802, 6417)  0.21753405888348784
  (4802, 4608)  0.24002350969074696
  (4802, 2425)  0.24002350969074696
  (4802, 3654)  0.262512960498006
  (4802, 5367)  0.22969114490410403
  (4802, 6996)  0.5700048226105303
```

```
co_sim = cosine_similarity(feature_vectors)
```

```
print(co_sim)
```

```
[[1.         0.07219487 0.037733   ... 0.         0.         0.        ]
 [0.07219487 1.         0.03281499 ... 0.03575545 0.         0.        ]
 [0.037733   0.03281499 1.         ... 0.         0.05389661 0.        ]
 ...
 [0.         0.03575545 0.         ... 1.         0.         0.02651502]
 [0.         0.         0.05389661 ... 0.         1.         0.        ]
 [0.         0.         0.         ... 0.02651502 0.         1.        ]]
```

```
co_sim.shape
```

```
(4803, 4803)
```

```
movie_name = input(' Enter your favourite movie name : ')
```

```
all_titles = data['title'].tolist()
print(all_titles)
```

```
['Avatar', "Pirates of the Caribbean: At World's End", 'Spectre', 'The Dark Knight Rises', 'John Carter', 'Spider-Man 3', 'Tangled', 'Avengers: Age of Ultron', 'Harry
```

```
import difflib
```

```
close_match = difflib.get_close_matches(movie_name, all_titles)
print(close_match)
```

```
['Iron Man', 'Iron Man 3', 'Iron Man 2']
```

```
closest_match = close_match[0]
print(closest_match)
```

```
Iron Man
```

```
index_of_the_movie = data[data.title == closest_match]['index'].values[0]
print(index_of_the_movie)
```

```
68
```

```
similarity_score = list(enumerate(co_sim[index_of_the_movie]))
print(similarity_score)
```

```
[(0, np.float64(0.033570748780675445)), (1, np.float64(0.0546448279236134)), (2, np.float64(0.013735500604224325)), (3, np.float64(0.006468756104392058)), (4, np.floa
```

```
len(similarity_score)
```

```
4803
```

```
sorted_similar_movies = sorted(similarity_score, key = lambda x:x[1], reverse = True)
print(sorted_similar_movies)
```

```
[(68, np.float64(1.0)), (79, np.float64(0.40890433998005965)), (31, np.float64(0.3146705244947752)), (7, np.float64(0.23944423963486416)), (16, np.float64(0.227044037
```

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

i = 1

for movie in sorted_similar_movies:
  index = movie[0]
```

```
  title_from_index = data[data.index==index]['title'].values[0]
  if (i<30):
    print(i, '.',title_from_index)
    i+=1
```

```
Movies suggested for you :

 1 . Iron Man
 2 . Iron Man 2
 3 . Iron Man 3
 4 . Avengers: Age of Ultron
 5 . The Avengers
 6 . Captain America: Civil War
 7 . Captain America: The Winter Soldier
 8 . Ant-Man
 9 . X-Men
 10 . Made
 11 . X-Men: Apocalypse
 12 . X2
 13 . The Incredible Hulk
 14 . The Helix... Loaded
 15 . X-Men: First Class
 16 . X-Men: Days of Future Past
 17 . Captain America: The First Avenger
 18 . Kick-Ass 2
 19 . Guardians of the Galaxy
 20 . Deadpool
 21 . Thor: The Dark World
 22 . G-Force
 23 . X-Men: The Last Stand
 24 . Duets
 25 . Mortdecai
 26 . The Last Airbender
 27 . Southland Tales
 28 . Zathura: A Space Adventure
 29 . Sky Captain and the World of Tomorrow
```

```
movie_name = input(' Enter your favourite movie name : ')
all_titles = data['title'].tolist()

close_match = difflib.get_close_matches(movie_name, all_titles)

closest_match = close_match[0]
```

```
index_of_the_movie = data[data.title == closest_match]['index'].values[0]

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

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

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

i = 1

for movie in sorted_similar_movies:
  index = movie[0]
  title_from_index = data[data.index==index]['title'].values[0]
  if (i<30):
    print(i, '.',title_from_index)
    i+=1
```

```
Enter your favourite movie name : batman
Movies suggested for you :

1 . Batman
2 . Batman Returns
3 . Batman & Robin
4 . The Dark Knight Rises
5 . Batman Begins
6 . The Dark Knight
7 . A History of Violence
8 . Superman
9 . Beetlejuice
10 . Bedazzled
11 . Mars Attacks!
12 . The Sentinel
13 . Planet of the Apes
14 . Man of Steel
15 . Suicide Squad
16 . The Mask
17 . Salton Sea
18 . Spider-Man 3
19 . The Postman Always Rings Twice
20 . Hang 'em High
21 . Spider-Man 2
22 . Dungeons & Dragons: Wrath of the Dragon God
23 . Superman Returns
24 . Jonah Hex
25 . Exorcist II: The Heretic
```

```
26 . Superman II
27 . Green Lantern
28 . Superman III
29 . Something's Gotta Give
```