```
import numpy as np
import pandas as pd
import difflib
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
```

```
movies_data = pd.read_csv('/content/movies.csv', engine='python', on_bad_lines='skip')
```

```
movies_data.head()
```

	l_language	original_title	ov
	en	Avatar	ce pa M
	en	Pirates of the Caribbean: At World's End	Ba t
	en	Spectre	A m se
	en	The Dark Knight Rises	Fo th of A
	en	John Carter	Ca

```
movies_data.shape
```

```
(1535, 24)
```

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

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

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

```
combined_features = movies_data['genres']+' '+movies_data['keywords']+' '+movies_data['tagline']+' '+movies_data['cast']+'
```

```
print(combined_features)
```

```
0       Action Adventure Fantasy Science Fiction cultu...
1       Adventure Fantasy Action ocean drug abuse exot...
2       Action Adventure Crime spy based on novel secr...
3       Action Crime Drama Thriller dc comics crime fi...
4       Action Adventure Science Fiction based on nove...
                              ...
1530    Drama confession airplane f word hangover airp...
1531    Action Adventure Thriller Science Fiction veni...
1532    Comedy Drama hotel painting wartime gunfight t...
```

```
1533    Drama Mystery american football billard baseba...
1534    Comedy Horror small town outbreak exterminator...
Length: 1535, dtype: object
```

```
vectorizer = TfidfVectorizer() # converting the text data to feature vectors
```

```
feature_vectors = vectorizer.fit_transform(combined_features)
```

```
print(feature_vectors)
```

```
<Compressed Sparse Row sparse matrix of dtype 'float64'
        with 42639 stored elements and shape (1535, 7969)>
  Coords        Values
  (0, 80)       0.0653413222967886
  (0, 115)      0.07314438029564296
  (0, 2432)     0.09653626312273343
  (0, 6234)     0.09420318257178323
  (0, 2508)     0.0943287664876066
  (0, 1675)     0.2513429681236611
  (0, 1375)     0.24135352032853655
  (0, 2725)     0.1641058814945882
  (0, 6600)     0.3161784179893521
  (0, 7609)     0.12354121036094529
  (0, 1450)     0.23360510264760642
  (0, 6551)     0.2131948485303838
  (0, 2274)     0.24135352032853655
  (0, 7013)     0.0694516904454956
  (0, 7830)     0.11994108940139879
  (0, 5118)     0.09483569395599635
  (0, 5264)     0.2513429681236611
  (0, 6130)     0.15296270913952276
  (0, 7834)     0.21728473663433562
  (0, 7956)     0.21728473663433562
  (0, 6116)     0.21728473663433562
  (0, 6448)     0.21728473663433562
  (0, 7652)     0.2131948485303838
  (0, 6715)     0.16227788497429332
  (0, 4028)     0.24135352032853655
  :     :
  (1534, 461)   0.18281158406254464
  (1534, 3604)  0.15721891502084806
  (1534, 7257)  0.13785880826415098
  (1534, 6633)  0.18790395824017128
  (1534, 3742)  0.17850604304359063
  (1534, 6517)  0.15212654084322141
  (1534, 7153)  0.1537230170679259
  (1534, 2662)  0.14185571929193413
  (1534, 1748)  0.1658817685842534
  (1534, 1630)  0.18790395824017128
  (1534, 783)   0.17148664759991852
  (1534, 4112)  0.20217169081924172
  (1534, 3211)  0.19413652052517072
  (1534, 2470)  0.20217169081924172
  (1534, 6146)  0.20217169081924172
  (1534, 5206)  0.21349662728186783
  (1534, 2379)  0.21349662728186783
  (1534, 2182)  0.21349662728186783
  (1534, 2426)  0.21349662728186783
  (1534, 3106)  0.21349662728186783
  (1534, 3957)  0.21349662728186783
  (1534, 2772)  0.21349662728186783
  (1534, 5750)  0.21349662728186783
  (1534, 4456)  0.21349662728186783
  (1534, 3814)  0.21349662728186783
```

```
similarity = cosine_similarity(feature_vectors) # getting the similarity scores using cosine similarity
```

```
print(similarity)
```

```
[[1.         0.06276354 0.03480607 ... 0.         0.00784426 0.        ]
 [0.06276354 1.         0.02724654 ... 0.         0.03738678 0.        ]
 [0.03480607 0.02724654 1.         ... 0.07446964 0.0131692  0.        ]
 ...
 [0.         0.         0.07446964 ... 1.         0.00416241 0.00411252]
 [0.00784426 0.03738678 0.0131692  ... 0.00416241 1.         0.        ]
 [0.         0.         0.         ... 0.00411252 0.         1.        ]]
```

```
print(similarity.shape)
```

```
(1535, 1535)
```

```
movie_name = input(' Enter your favourite movie name : ') # getting the movie name from the user
```

```
 Enter your favourite movie name : Spectre
```

```
list_of_all_titles = movies_data['title'].tolist() # creating a list with all the movie names given in the dataset
print(list_of_all_titles)
```

```
['Avatar', "Pirates of the Caribbean: At World's End", 'Spectre', 'The Dark Knight Rises', 'John Carter', 'Spider-Man 3', 'T
```

```
find_close_match = difflib.get_close_matches(movie_name, list_of_all_titles) # finding the close match for the movie name g
print(find_close_match)
```

```
['Spectre', 'Sphere', 'Species']
```

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

```
Spectre
```

```
index_of_the_movie = movies_data[movies_data.title == close_match]['index'].values[0] # finding the index of the movie witl
print(index_of_the_movie)
```

```
2
```

```
similarity_score = list(enumerate(similarity[index_of_the_movie])) # getting a list of similar movies
print(similarity_score)
```

```
[(0, np.float64(0.034806065306741324)), (1, np.float64(0.027246536242224284)), (2, np.float64(1.0000000000000002)), (3, np.f
```

```
len(similarity_score)
```

```
1535
```

```
sorted_similar_movies = sorted(similarity_score, key = lambda x:x[1], reverse = True) # sorting the movies based on their s
print(sorted_similar_movies)
```

```
238643228)), (256, np.float64(0.07717456074029956)), (1137, np.float64(0.07713025444835594)), (1474, np.float64(0.07664738976
```

```
print('Movies suggested for you : \n') # print the name of similar movies based on the index

i = 1

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

```
Movies suggested for you :

1 . Spectre
2 . Skyfall
3 . Mission: Impossible - Ghost Protocol
4 . Johnny English Reborn
5 . The Incredibles
6 . The Sorcerer's Apprentice
7 . Quantum of Solace
8 . Red Dragon
9 . The Green Hornet
10 . The Legend of Tarzan
11 . Water for Elephants
12 . Harry Potter and the Order of the Phoenix
13 . Tears of the Sun
14 . The Girl with the Dragon Tattoo
15 . Clash of the Titans
16 . Road to Perdition
17 . Django Unchained
18 . Wrath of the Titans
19 . Harry Potter and the Goblet of Fire
20 . GoldenEye
21 . Inglourious Basterds
22 . The Spirit
23 . The English Patient
24 . Insurgent
25 . Sherlock Holmes
26 . RED 2
27 . The Prince of Egypt
28 . Safe House
29 . Fun with Dick and Jane
```

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

```
list_of_all_titles = movies_data['title'].tolist()

find_close_match = difflib.get_close_matches(movie_name, list_of_all_titles)

close_match = find_close_match[0]

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

similarity_score = list(enumerate(similarity[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 = movies_data[movies_data.index==index]['title'].values[0]
  if (i<30):
    print(i, '.',title_from_index)
    i+=1
```

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

1 . Titanic
2 . The Dilemma
3 . The Day the Earth Stood Still
4 . Revolutionary Road
5 . Primary Colors
6 . Stuck on You
7 . Flushed Away
8 . The Blind Side
9 . Contagion
10 . Almost Famous
11 . The Great Gatsby
12 . The Host
13 . The Phantom
14 . Gangs of New York
15 . This Is the End
16 . Dragonfly
17 . Insurgent
18 . Shutter Island
19 . The Aviator
20 . Something Borrowed
21 . Body of Lies
22 . Laws of Attraction
23 . Terminator 2: Judgment Day
24 . All the King's Men
25 . The Holiday
26 . The Beach
27 . Blood Diamond
28 . The Departed
29 . The Life of David Gale
```