# CODING

## Importing the dependencies

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

## Data Collection and Pre-Processing

```
# loading the data from the csv file to apandas dataframe
movies_data = pd.read_csv('/content/movies.csv')
```

```
# printing the first 5 rows of the dataframe
movies_data.head()
```

| | index | budget | genres | homepage | id | keywords | original_language | original_title | overview | popularity | production_companies | production_countri |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 237000000 | Action Adventure Fantasy Science Fiction | http://www.avatarmovie.com/ | 19995 | culture clash future space war space colony so... | en | Avatar | In the 22nd century, a paraplegic Marine is di... | 150.437577 | [{"name": "Ingenious Film Partners", "id": 289... | [{"iso_3166_1": "U "name": "United Stat |
| 1 | 1 | 300000000 | Adventure Fantasy Action | http://disney.go.com/disneypictures/pirates/ | 285 | ocean drug abuse exotic island east india trad... | en | Pirates of the Caribbean: At World's End | Captain Barbossa, long believed to be dead, ha... | 139.082615 | [{"name": "Walt Disney Pictures", "id": 2}, {"... | [{"iso_3166_1": "U "name": "United Stat |
| 2 | 2 | 245000000 | Action Adventure Crime | http://www.sonypictures.com/movies/spectre/ | 206647 | spy based on novel secret agent sequel mi6 | en | Spectre | A cryptic message from Bond's past sends him o... | 107.376788 | [{"name": "Columbia Pictures", "id": 5}, {"nam... | [{"iso_3166_1": "G "name": "Unit Kingdom |
| 3 | 3 | 250000000 | Action Crime Drama Thriller | http://www.thedarkknightrises.com/ | 49026 | dc comics crime fighter terrorist secret | en | The Dark Knight Rises | Following the death of District Attorney Harve... | 112.312950 | [{"name": "Legendary Pictures", "id": 923}, {"... | [{"iso_3166_1": "U "name": "United Stat |

| production_countries | release_date | revenue | runtime | spoken_languages | status | tagline | title | vote_average | vote_count | cast | crew | director |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| [{"iso_3166_1": "US", "name": "United States o... | 2009-12-10 | 2787965087 | 162.0 | [{"iso_639_1": "en", "name": "English"}, {"iso... | Released | Enter the World of Pandora. | Avatar | 7.2 | 11800 | Sam Worthington Zoe Saldana Sigourney Weaver S... | [{'name': 'Stephen E. Rivkin', 'gender': 0, 'd... | James Cameron |
| [{"iso_3166_1": "US", "name": "United States o... | 2007-05-19 | 961000000 | 169.0 | [{"iso_639_1": "en", "name": "English"}] | Released | At the end of the world, the adventure begins. | Pirates of the Caribbean: At World's End | 6.9 | 4500 | Johnny Depp Orlando Bloom Keira Knightley Stel... | [{'name': 'Dariusz Wolski', 'gender': 2, 'depa... | Gore Verbinski |
| [{"iso_3166_1": "GB", "name": "United Kingdom"... | 2015-10-26 | 880674609 | 148.0 | [{"iso_639_1": "fr", "name": "Fran\u00e7ais"},... | Released | A Plan No One Escapes | Spectre | 6.3 | 4466 | Daniel Craig Christoph Waltz L\u00e9a Seydoux ... | [{'name': 'Thomas Newman', 'gender': 2, 'depar... | Sam Mendes |
| [{"iso_3166_1": "US", "name": "United States o... | 2012-07-16 | 1084939099 | 165.0 | [{"iso_639_1": "en", "name": "English"}] | Released | The Legend Ends | The Dark Knight Rises | 7.6 | 9106 | Christian Bale Michael Caine Gary Oldman Anne ... | [{'name': 'Hans Zimmer', 'gender': 2, 'departm... | Christopher Nolan |

```
# number of rows and columns in the data frame

movies_data.shape
```

```
(4803, 24)
```

```
# selecting the relevant features for recommendation

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

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

```
# replacing the null valuess with null string

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

```
# combining all the 5 selected features

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

```python
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...
                                ...
4798    Action Crime Thriller united states\u2013mexic...
4799    Comedy Romance  A newlywed couple's honeymoon ...
4800    Comedy Drama Romance TV Movie date love at fir...
4801      A New Yorker in Shanghai Daniel Henney Eliza...
4802    Documentary obsession camcorder crush dream gi...
Length: 4803, dtype: object
```

```python
# converting the text data to feature vectors

vectorizer = TfidfVectorizer()
```

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

```python
print(feature_vectors)
```

```
(0, 2432)     0.17272411194153
(0, 7755)     0.1128035714854756
(0, 13024)    0.1942362060108871
(0, 10229)    0.16058685400095302
(0, 8756)     0.22709015857011816
(0, 14608)    0.15150672398763912
(0, 16668)    0.19843263965100372
(0, 14064)    0.20596090415084142
(0, 13319)    0.2177470539412484
(0, 17290)    0.20197912553916567
(0, 17007)    0.23643326319898797
(0, 13349)    0.15021264094167086
(0, 11503)    0.27211310056983656
(0, 11192)    0.09049319826481456
(0, 16998)    0.1282126322850579
(0, 15261)    0.07095833561276566
(0, 4945)     0.24025852494110758
(0, 14271)    0.21392179219912877
(0, 3225)     0.24960162956997736
(0, 16587)    0.12549432354918996
(0, 14378)    0.33962752210959823
(0, 5836)     0.1646750903586285
(0, 3065)     0.22208377802661425
(0, 3678)     0.21392179219912877
(0, 5437)     0.1036413987316636
:     :
(4801, 17266) 0.2886098184932947
(4801, 4835)  0.24713765026963996
(4801, 403)   0.17727585190343226
(4801, 6935)  0.2886098184932947
(4801, 11663) 0.21557500762727902
(4801, 1672)  0.1564793427630879
(4801, 10929) 0.13504166990041588
```


```
(0, 5437)     0.1036413987316636
:     :
(4801, 17266) 0.2886098184932947
(4801, 4835)  0.24713765026963996
(4801, 403)   0.17727585190343226
(4801, 6935)  0.2886098184932947
(4801, 11663) 0.21557500762727902
(4801, 1672)  0.1564793427630879
(4801, 10929) 0.13504166990041588
```


```
(4801, 7474)  0.11307961713172225
(4801, 3796)  0.3342808988877418
(4802, 6996)  0.5700048226105303
(4802, 5367)  0.22969114490410403
(4802, 3654)  0.262512960498006
(4802, 2425)  0.24002350969074696
(4802, 4608)  0.24002350969074696
(4802, 6417)  0.21753405888348784
(4802, 4371)  0.1538239182675544
(4802, 12989) 0.1696476532191718
(4802, 1316)  0.1960747079005741
(4802, 4528)  0.19504460807622875
(4802, 3436)  0.21753405888348784
(4802, 6155)  0.18056463596934083
(4802, 4980)  0.16078053641367315
(4802, 2129)  0.3099656128577656
(4802, 4518)  0.16784466610624255
(4802, 11161) 0.17867407682173203
```

## Cosine Similarity

```
# getting the similarity scores using cosine similarity

similarity = cosine_similarity(feature_vectors)
```

```
print(similarity)
```

```
[[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.        ]]
```

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

```
(4803, 4803)
```

## Getting the movie name from the user

```
# getting the movie name from the user

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

```
 Enter your favourite movie name : iron man
```

```
# creating a list with all the movie names given in the dataset

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

```
['Avatar', "Pirates of the Caribbean: At World's End", 'Spectre', 'The Dark Knight Rises', 'John Carter',
```

```
', 'Blackhat', 'Sky Captain and the World of Tomorrow', 'Basic Instinct 2', 'Escape Plan', 'The Legend of Hercules', 'The Sum of All Fears',
```

```python
# finding the close match for the movie name given by the user

find_close_match = difflib.get_close_matches(movie_name, list_of_all_titles)
print(find_close_match)
```

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

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

```
Iron Man
```

```python
# finding the index of the movie with title

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

```
68
```

```python
# getting a list of similar movies

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

```
[(0, 0.033570748780675445), (1, 0.0546448279236134), (2, 0.013735500604224323), (3, 0.006468756104392058), (4, 0.03268943310073386),
```

```
40364272462), (3952, 0.012151496446083307), (3250, 0.012147477967543958), (3846, 0.012128241991713645), (1943, 0.012127214318469933),
```

```python
# print the name of similar movies based on the index

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
```

## Movie Recommendation System

```python
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
```

# SAMPLE OUTPUT