NAME : NACHIKET DHAMPALWAR

ROLL NO : TEAD21225

SUBJECT : SL III

CLASS : TE

BRANCH : AI&DS

Mini Project

Problem Statment : Develop a movie recommendation model using the scikit-learn library in python.

Code & Output:

```
💡 Click here to ask Blackbox to help you code faster
import numpy as np
import pandas as pd
import difflib
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
```
[4] Python

Data Collection and Pre-Processing

```
💡 Click here to ask Blackbox to help you code faster
#Loading Data from the csv file
movies_data= pd.read_csv('movie_dataset.csv')
```
[9] Python

```
💡 Click here to ask Blackbox to help you code faster
#printing first 5 rows of dataset
movies_data.head()
```
[10] Python

	index	budget	genres	homepage	id	keywords	original_language	original_title	overview	popularity	...	runtime	spoken_languages	status	tagline	title	vote_average	vote_count
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	...	162.0	[{"iso_639_1": "en", "name": "English"}, {"iso...	Released	Enter the World of Pandora.	Avatar	7.2	11800
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	...	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

	index	budget	genres	homepage	id	keywords	original_language	original_title	overview	popularity	...	runtime	spoken_languages	status	tagline	title	vote_average	vote_count
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	...	162.0	[{"iso_639_1": "en", "name": "English"}, {"iso...	Released	Enter the World of Pandora.	Avatar	7.2	11800
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	...	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
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	...	148.0	[{"iso_639_1": "fr", "name": "Fran\u00e7ais"},...	Released	A Plan No One Escapes	Spectre	6.3	4466
3	3	250000000	Action Crime Drama Thriller	http://www.thedarkknightrises.com/	49026	dc comics crime fighter terrorist secret ident...	en	The Dark Knight Rises	Following the death of District Attorney Harve...	112.312950	...	165.0	[{"iso_639_1": "en", "name": "English"}]	Released	The Legend Ends	The Dark Knight Rises	7.6	9106
4	4	260000000	Action Adventure Science Fiction	http://movies.disney.com/john-carter	49529	based on novel mars medallion space travel pri...	en	John Carter	John Carter is a war-weary, former military ca...	43.926995	...	132.0	[{"iso_639_1": "en", "name": "English"}]	Released	Lost in our world, found in another.	John Carter	6.1	2124

5 rows × 24 columns

[13]
```
#No. of Rows & Columns
movies_data.shape
```

```
(4803, 24)
```

[14]
💡 Click here to ask Blackbox to help you code faster
```
#selecting the relevant features for recommendation
selected_features = ['genres','keywords','tagline','cast','director']
print(selected_features)
```

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

[19]
💡 Click here to ask Blackbox to help you code faster
```
# replacing the null valuess with null string
for feature in selected_features:
  movies_data[feature] = movies_data[feature].fillna('')
```

[20]
💡 Click here to ask Blackbox to help you code faster
```
movies_data.isnull().sum()
```

```
index                   0
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
```

```
cast                          0
crew                          0
director                      0
dtype: int64
```

💡 Click here to ask Blackbox to help you code faster

```python
# combining all the 5 selected features
combined_features = movies_data['genres']+' '+movies_data['keywords']+' '+movies_data['tagline']+' '+movies_data['cast']+' '+movies_data['director']
```

[21]

💡 Click here to ask Blackbox to help you code faster

```python
print(combined_features)
```

[22]

```
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
```

💡 Click here to ask Blackbox to help you code faster

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

vectorizer = TfidfVectorizer()
```

[23]

💡 Click here to ask Blackbox to help you code faster

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

[24]

💡 Click here to ask Blackbox to help you code faster

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

[24]

💡 Click here to ask Blackbox to help you code faster

```python
print(feature_vectors)
```

[25]

```
  (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
  :	:
  (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)
```

[26]


```
print(similarity)
```

[27]

```
[[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.        ]]
```



```
similarity.shape
```

[30]

```
(4803, 4803)
```

Getting Movie Name from the user


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

[31]

```
 Enter your favourite movie name : batman
```



```
#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)
```

[32]

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



```
# 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)
```

[33]

```
['Batman', 'Batman', 'Catwoman']
```



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

[34]

```
Batman
```



```
# 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)
```

[35]

```
1359
```

```
# getting a list of similar movies

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

```
[(0, 0.02531512269737111), (1, 0.04983293064399152), (2, 0.013599520029326722), (3, 0.20438773732168222), (4, 0.024929726723526918), (5, 0.11533013884014888), (6, 0.0), (7, 0.
```

Click here to ask Blackbox to help you code faster

```
len(similarity_score)
```

```
4803
```

Click here to ask Blackbox to help you code faster

```
# sorting the movies based on their similarity score

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

```
[(1359, 1.0), (428, 0.4311643836232694), (210, 0.25737999820859625), (3, 0.20438773732168222), (119, 0.19262528757150407), (65, 0.1775581506611392), (1512, 0.14705162654306442
```

Click here to ask Blackbox to help you code faster

```
# 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<20):
    print(i, '.',title_from_index)
    i+=1
```

```
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
```

Click here to ask Blackbox to help you code faster

```
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
```

```
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<=20):
    print(i, '.',title_from_index)
    i+=1
```

[47]

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

1 . Vampires
2 . Contact
3 . BloodRayne
4 . Dudley Do-Right
5 . Priest
6 . Ghosts of Mississippi
7 . Julia
8 . Stranded
9 . Me and Orson Welles
10 . The Shadow
11 . Wal-Mart: The High Cost of Low Price
12 . The Girl with the Dragon Tattoo
13 . Salvador
14 . The Lord of the Rings: The Two Towers
15 . xXx
16 . High Anxiety
17 . Vampire in Brooklyn
18 . The Black Hole
19 . How to Be Single
20 . Judgment at Nuremberg
```