



```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import os

movies=pd.read_csv('/content/Movie_Recommendar/dataset/tmdb_5000_movies.csv')
credits=pd.read_csv('/content/Movie_Recommendar/dataset/tmdb_5000_credits.csv')
```

```
movies.head(2)
```





	budget	genres	homepage	id	keywords	origin:
0	237000000	{{"id": 28, "name": "Action"}, {"id": 12, "nam...	http://www.avatarmovie.com/	1995	{{"id": 1463, "name": "culture clash"}, {"id": ...	
1	300000000	{{"id": 12, "name": "Adventure"}, {"id": 14, "...	http://disney.go.com/disneypictures/pirates/	285	{{"id": 270, "name": "ocean"}, {"id": 726, "na...	

Next steps:

[Generate code with movies](#)

☒ [View recommended plots](#)

```
credits.head(2)
```

	movie_id	title	cast	crew	
0	1995	Avatar	{{"cast_id": 242, "character": "Jake Sully", "...	{{"credit_id": "52fe48009251416c750aca23", "de...	 

Next steps:

[Generate code with credits](#)

☒ [View recommended plots](#)

```
movies.shape
```

(4803, 20)

```
credits.shape
```

(4803, 4)

```
#merging dataset
movies=movies.merge(credits,on="title")
```

```
movies.head(2)
```

	budget	genres	homepage	id	keywords	origin:
0	237000000	{{"id": 28, "name": "Action"}, {"id": 12, "nam...	http://www.avatarmovie.com/	1995	{{"id": 1463, "name": "culture clash"}, {"id": ...	
1	300000000	{{"id": 12, "name": "Adventure"}, {"id": 14, "...	http://disney.go.com/disneypictures/pirates/	285	{{"id": 270, "name": "ocean"}, {"id": 726, "na...	

2 rows × 23 columns



```
movies.shape
```

```
(4809, 23)
```

```
# node need of budget,homepage,original language,popularity,
movies.columns
```

```
Index(['budget', 'genres', 'homepage', 'id', 'keywords', 'original_language',
       'original_title', 'overview', 'popularity', 'production_companies',
       'production_countries', 'release_date', 'revenue', 'runtime',
       'spoken_languages', 'status', 'tagline', 'title', 'vote_average',
       'vote_count', 'movie_id', 'cast', 'crew'],
      dtype='object')
```

```
movies=movies[['movie_id','title','overview','genres','keywords',"cast","crew"]];
```

```
movies.head(2)
```

	movie_id	title	overview	genres	keywords	cast
0	19995	Avatar	In the 22nd century, a paraplegic Marine is di...	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	[{"id": 1463, "name": "culture clash"}, {"id": ...	[{"cast_id": 242, "character": "Jake Sully", "...

Next steps:

[Generate code with movies](#)

[View recommended plots](#)

```
movies.shape
```

```
(4809, 7)
```

```
movies.isnull().sum()
```

```
movie_id    0
title       0
overview    3
genres      0
keywords    0
cast        0
crew        0
dtype: int64
```

```
movies.dropna(inplace=True)
```

```
movies.isnull().sum()
```

```
movie_id    0
title       0
overview    0
genres      0
keywords    0
cast        0
crew        0
dtype: int64
```

```
movies.shape
```

```
(4806, 7)
```

```
movies.duplicated().sum()
```

```
0
```

```
movies.iloc[0]['genres']
```

```
'[{"id": 28, "name": "Action"}, {"id": 12, "name": "Adventure"}, {"id": 14, "name": "Fantasy"}, {"id": 878, "name": "Science Fiction"}]'
```




```
2 [{"cast_id": 1, "character": "James Bond", "cr...
3 [{"cast_id": 2, "character": "Bruce Wayne / Ba...
4 [{"cast_id": 5, "character": "John Carter", "c...
Name: cast, dtype: object
```

```
# movies['cast']=movies['cast'].apply(convert)
import ast
```

```
def convert_cast(text):
    l = []
    counter=0;
    for i in ast.literal_eval(text):
        if counter < 3:
            l.append(i['name'])
            counter +=1
    return l
```

```
movies['cast'] = movies['cast'].apply(convert_cast)
```

```
movies.head(2)
```

	movie_id	title	overview	genres	keywords	cast
0	19995	Avatar	In the 22nd century, a paraplegic Marine is di...	[Action, Adventure, Fantasy, Science Fiction]	[culture clash, future, space war, space colon...	[Sam Worthington, Zoe Saldana, Sigourney Weaver...]

Next steps:

[Generate code with movies](#)

☒ [View recommended plots](#)

```
movies.iloc[0]['crew']

'{"credit_id": "52fe48009251416c750aca23", "department": "Editing", "gender": 0, "id": 1721, "job": "Editor", "name": "Stephen E. Rivkin"}, {"credit_id": "539c47ecc3a36810e3001f87", "department": "Art", "gender": 2, "id": 496, "job": "Production Design", "name": "Rick Carter"}, {"credit_id": "54491c89c3a3680fb4001cf7", "department": "Sound", "gender": 0, "id": 900, "job": "Sound Designer", "name": "Christopher Boyes"}, {"credit_id": "54491cb70e0a267480001bd0", "department": "Sound", "gender": 0, "id": 900, "job": "Supervising Sound Editor", "name": "Christopher Boyes"}, {"credit_id": "539c4a4cc3a36810c9002101", "department": "Production", "gender": 1, "id": 1262, "job": "Cast
```

```
import ast

def fetch_director(text):
    l=[]
    for i in ast.literal_eval(text):
        if i['job'] == "Director":
            l.append(i['name'])
            break;
    return l

movies['crew']=movies['crew'].apply(fetch_director)
```

```
movies.head(2)
```

	movie_id	title	overview	genres	keywords	cast	crew
0	19995	Avatar	In the 22nd century, a paraplegic Marine is di...	[Action, Adventure, Fantasy, Science Fiction]	[culture clash, future, space war, space colon...	[Sam Worthington, Zoe Saldana, Sigourney Weaver...]	[James Cameron]

Next steps:

[Generate code with movies](#)

☒ [View recommended plots](#)

```
movies.iloc[0]['overview']
```



'In the 22nd century, a paraplegic Marine is dispatched to the moon Pandora on a unique mission, but becomes torn between following orders and protecting an alien civilization'

```
movies['overview'] = movies['overview'].apply(lambda x:x.split())
```

```
movies.head()
```

	movie_id	title	overview	genres	keywords	cast	crew
0	19995	Avatar	[In, the, 22nd, century,, a, paraplegic, Marin...	[Action, Adventure, Fantasy, Science Fiction]	[culture clash, future, space war, space colon...	[Sam Worthington, Zoe Saldana, Sigourney Weaver]	[James Cameron]
1	285	Pirates of the Caribbean: At World's End	[Captain, Barbossa,, long, believed, to, be, d...	[Adventure, Fantasy, Action]	[ocean, drug abuse, exotic island, east india ...	[Johnny Depp, Orlando Bloom, Keira Knightley]	[Gore Verbinski]
			[A, cryptic, message,	[Action	[spy, based on novel	[Daniel Craig	

Next steps:

Generate code with movies

☒ View recommended plots

```
movies.iloc[0]['overview']
```

['In',
'the',
'22nd',
'century',
'a',
'paraplegic',
'Marine',
'is',
'dispatched',
'to',
'the',
'moon',
'Pandora',
'on',
'a',
'unique',
'mission',
'but',
'becomes',
'torn',
'between',
'following',
'orders',
'and',
'protecting',
'an',
'alien',
'civilization.']

```
def removespace(word):
```

```
l = []  
for i in word:  
    l.append(i.replace(" ", ""))  
return l;
```

```
movies['cast'] = movies['cast'].apply(removespace)  
movies['crew'] = movies['crew'].apply(removespace)  
movies['genres'] = movies['genres'].apply(removespace)  
movies['keywords'] = movies['keywords'].apply(removespace)  
# movies['cast'] = movies['cast'].apply(removespace)
```

```
movies.head()
```



	movie_id	title	overview	genres	keywords	cast
0	19995	Avatar	[In, the, 22nd, century,, a, paraplegic, Marin...	[Action, Adventure, Fantasy, ScienceFiction]	[cultureclash, future, spacewar, spacecolony, ...	[SamWorthington, ZoeSaldana, SigourneyWeaver]
1	285	Pirates of the Caribbean: At World's End	[Captain, Barbossa,, long, believed, to, be, d...	[Adventure, Fantasy, Action]	[ocean, drugabuse, exoticisland, eastindiatrad...	[JohnnyDepp, OrlandoBloom, KeiraKnightley]
2	206647	Spectre	[A, cryptic, message, from, Bond's, past, ...	[Action, Adventure, Crime]	[spy, basedonnovel, secretagent, sequel, mi6, ...	[DanielCraig, ChristophWaltz, LéaSeydoux]

Next steps:

[Generate code with movies](#)

☒ [View recommended plots](#)

```
movies['tags']=movies['overview']+movies['genres']+movies['keywords']+movies['cast']+movies['crew']
```

```
movies.head()
```

	movie_id	title	overview	genres	keywords	cast
0	19995	Avatar	[In, the, 22nd, century,, a, paraplegic, Marin...	[Action, Adventure, Fantasy, ScienceFiction]	[cultureclash, future, spacewar, spacecolony, ...	[SamWorthington, ZoeSaldana, SigourneyWeaver]
1	285	Pirates of the Caribbean: At World's End	[Captain, Barbossa,, long, believed, to, be, d...	[Adventure, Fantasy, Action]	[ocean, drugabuse, exoticisland, eastindiatrad...	[JohnnyDepp, OrlandoBloom, KeiraKnightley]
2	206647	Spectre	[A, cryptic, message, from, Bond's, past, send...	[Action, Adventure, Crime]	[spy, basedonnovel, secretagent, sequel, mi6, ...	[DanielCraig, ChristophWaltz, LéaSeydoux]
3	49026	The Dark Knight	[Following, the, death, of, District	[Action, Crime, Drama,	[dccomics, crimefighter, terrorist	[ChristianBale, MichaelCaine, [Christop

Next steps:

[Generate code with movies](#)

☒ [View recommended plots](#)

```
movies.iloc[0]['tags']
```

```
['In',
 'the',
 '22nd',
 'century,',
 'a',
 'paraplegic',
 'Marine',
 'is',
 'dispatched',
 'to',
 'the',
 'moon',
 'Pandora',
 'on',
 'a',
 'unique',
 'mission,',
 'but',
 'becomes',
 'torn',
 'between',
 'following',
 'orders',
 'and',
```



```
'protecting',
'an',
'alien',
'civilization.',
'Action',
'Adventure',
'Fantasy',
'ScienceFiction',
'cultureclash',
'future',
'spacewar',
'spacecolony',
'society',
'spacetravel',
'futuristic',
'romance',
'space',
'alien',
'tribe',
'alienplanet',
'cgi',
'marine',
'soldier',
'battle',
'loveaffair',
'antiwar',
'powerrelations',
'mindandsoul',
'3d',
'SamWorthington',
'ZoeSaldana',
'SigourneyWeaver',
'JamesCameron']
```

```
new_df=movies[["movie_id","title","tags"]]
```

```
new_df.head()
```

	movie_id		title	tags	
0	19995		Avatar	[In, the, 22nd, century,, a, paraplegic, Marin...	
1	285	Pirates of the Caribbean: At World's End		[Captain, Barbossa,, long, believed, to, be, d...	
2	206647		Spectre	[A, cryptic, message, from, Bond's, past, send...	
3	40000	The Dark Knight Rises		[Following, the, death, of, District,	

Next steps:

Generate code with new_df

View recommended plots

```
new_df['tags']=new_df['tags'].apply(lambda x: " ".join(x))
```

<ipython-input-221-6204029b9a1d>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
new_df['tags']=new_df['tags'].apply(lambda x: " ".join(x))

```
new_df.head()
```

	movie_id		title	tags	
0	19995		Avatar	In the 22nd century, a paraplegic Marine is di...	
1	285	Pirates of the Caribbean: At World's End		Captain Barbossa, long believed to be dead, ha...	
2	206647		Spectre	A cryptic message from Bond's past sends him o...	
3	40000	The Dark Knight Rises		Following the death of District Attorney	

Next steps:

Generate code with new_df

View recommended plots



```
new_df['tags']=new_df['tags'].apply(lambda x: x.lower())
```

<ipython-input-223-d51011ffdf14>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-c
new_df['tags']=new_df['tags'].apply(lambda x: x.lower())

```
new_df.head()
```

	movie_id		title	tags	
0	19995	Avatar	in the 22nd century, a paraplegic marine is di...		
1	285	Pirates of the Caribbean: At World's End	captain barbossa, long believed to be dead, ha...		
2	206647	Spectre	a cryptic message from bond's past sends him o...		
3	40000	The Dark Knight Rises	following the death of district attorney		

Next steps:

[Generate code with new_df](#)

☒ [View recommended plots](#)

```
import nltk
from nltk.stem import PorterStemmer
```

```
ps =PorterStemmer()
```

```
def stems(text):
    l=[]
    for i in text.split():
        l.append(ps.stem(i))
    return " ".join(l)
```

```
new_df['tags']=new_df['tags'].apply(stems)
```

<ipython-input-231-2fbf24761000>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-c
new_df['tags']=new_df['tags'].apply(stems)

```
new_df.iloc[0]['tags']
```

'in the 22nd century, a paraplegic marin is dispatch to the moon pandora on a uniqu miss
ion, but becom torn between follow order and protect an alien civilization. action adv
entur fantasi sciencefict cultureclash futur spacewar spacecoloni societi spacetravel
futurist romanc space alien tribe alienplanet cgi marin soldier battl loveaffair antiw

```
from sklearn.feature_extraction.text import CountVectorizer
cv=CountVectorizer(max_features=5000,stop_words='english')
```

```
vector=cv.fit_transform(new_df['tags']).toarray()
```

```
vector
```

```
array([[0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       ...,
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0]])
```

```
vector.shape
```

```
(4806, 5000)
```




```

from sklearn.metrics.pairwise import cosine_similarity

similarity=cosine_similarity(vector)

similarity

array([[1.          , 0.08346223, 0.0860309 , ..., 0.04499213, 0.          ,
        0.          ],
       [0.08346223, 1.          , 0.06063391, ..., 0.02378257, 0.          ,
        0.02615329],
       [0.0860309 , 0.06063391, 1.          , ..., 0.02451452, 0.          ,
        0.          ],
       ...,
       [0.04499213, 0.02378257, 0.02451452, ..., 1.          , 0.03962144,
        0.04229549],
       [0.          , 0.          , 0.          , ..., 0.03962144, 1.          ,
        0.08714204],
       [0.          , 0.02615329, 0.          , ..., 0.04229549, 0.08714204,
        1.          ]])

similarity.shape

(4806, 4806)

new_df[new_df['title'] == 'Spider-Man'].index[0]

159

def recommend(movie):
    index = new_df[new_df['title'] == movie].index[0]
    distances = sorted(list(enumerate(similarity[index])), reverse=True, key=lambda x: x[1])
    for i in distances[1:10]:
        print(new_df.iloc[i[0]].title)

recommend('Spider-Man')

Spider-Man 3
Spider-Man 2
The Amazing Spider-Man 2
Arachnophobia
Kick-Ass
The Amazing Spider-Man
21 Jump Street
X-Men
Eight Legged Freaks

recommend("The Dark Knight Rises")

The Dark Knight
Batman Returns
Batman
Batman Forever
Batman Begins
Batman
Batman & Robin
Nighthawks
Slow Burn

import pickle
pickle.dump(new_df, open('/content/Movie_Recommendar/artificats/movie_list.pkl' , "wb"))
pickle.dump(new_df, open('/content/Movie_Recommendar/artificats/similarity.pkl' , "wb"))

```

Start coding or [generate](#) with AI.

Start coding or [generate](#) with AI.

