Personalized Movie Recommendation System

Overview

This project implements a movie recommender system using Python. The system uses data from the TMDB (The Movie Database) to recommend movies based on the similarity of their features. The recommendation process involves data preprocessing, feature extraction, and similarity calculation using cosine similarity.

First Import Library

In [2]:
```
import numpy as np
import pandas as pd
```

Import Datasets

In [3]:
```
movies = pd.read_csv('tmdb_5000_movies.csv')
credits = pd.read_csv('tmdb_5000_credits.csv')
```

This code for checking datasets for information

movies.head(1)

credits.head(2)

Data Merging and Cleaning

The two datasets are merged on the 'title' column. Unnecessary columns are dropped, and null values are removed to clean the data.

In [4]:
```
# This is for merge the dataset base on title column

movies = movies.merge(credits,on='title')
```

In [5]:
```
# In this dataset we have so many column for diffrent types of data but we dont have to need lots of data so i have so go to s
# genres, # id, # keywords, # title, # overview, # cast, # crew
# So i will go these column
# Then my new dataset is,

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

In [6]:
```
# This is my new dataset

movies
```

Out[6]:

	movie_id	title	overview	genres	keywords	cast	crew
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", "...	[{"credit_id": "52fe48009251416c750aca23", "de...
1	285	Pirates of the Caribbean: At World's End	Captain Barbossa, long believed to be dead, ha...	[{"id": 12, "name": "Adventure"}, {"id": 14, "...	[{"id": 270, "name": "ocean"}, {"id": 726, "na...	[{"cast_id": 4, "character": "Captain Jack Spa...	[{"credit_id": "52fe4232c3a36847f800b579", "de...
2	206647	Spectre	A cryptic message from Bond’s past sends him o...	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	[{"id": 470, "name": "spy"}, {"id": 818, "name...	[{"cast_id": 1, "character": "James Bond", "cr...	[{"credit_id": "54805967c3a36829b5002c41", "de...
3	49026	The Dark Knight Rises	Following the death of District Attorney Harve...	[{"id": 28, "name": "Action"}, {"id": 80, "nam...	[{"id": 849, "name": "dc comics"}, {"id": 853,...	[{"cast_id": 2, "character": "Bruce Wayne / Ba...	[{"credit_id": "52fe4781c3a36847f81398c3", "de...
4	49529	John Carter	John Carter is a war-weary, former military ca...	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	[{"id": 818, "name": "based on novel"}, {"id":...	[{"cast_id": 5, "character": "John Carter", "c...	[{"credit_id": "52fe479ac3a36847f813eaa3", "de...
...	...	...	...	...	...	...	...
4804	9367	El Mariachi	El Mariachi just wants to play his guitar and ...	[{"id": 28, "name": "Action"}, {"id": 80, "nam...	[{"id": 5616, "name": "united states\u2013mexi...	[{"cast_id": 1, "character": "El Mariachi", "c...	[{"credit_id": "52fe44eec3a36847f80b280b", "de...
4805	72766	Newlyweds	A newlywed couple's honeymoon is upended by th...	[{"id": 35, "name": "Comedy"}, {"id": 10749, "...	[]	[{"cast_id": 1, "character": "Buzzy", "credit_...	[{"credit_id": "52fe487dc3a368484e0fb013", "de...
4806	231617	Signed, Sealed, Delivered	"Signed, Sealed, Delivered" introduces a dedic...	[{"id": 35, "name": "Comedy"}, {"id": 18, "nam...	[{"id": 248, "name": "date"}, {"id": 699, "nam...	[{"cast_id": 8, "character": "Oliver O\u2019To...	[{"credit_id": "52fe4df3c3a36847f8275ecf", "de...

	movie_id	title	overview	genres	keywords	cast	crew
4807	126186	Shanghai Calling	When ambitious New York attorney Sam is sent t...	[]	[]	[{"cast_id": 3, "character": "Sam", "credit_id...	[{"credit_id": "52fe4ad9c3a368484e16a36b", "de...
4808	25975	My Date with Drew	Ever since the second grade when he first saw ...	[{"id": 99, "name": "Documentary"}]	[{"id": 1523, "name": "obsession"}, {"id": 224...	[{"cast_id": 3, "character": "Herself", "credi...	[{"credit_id": "58ce021b9251415a390165d9", "de...

4809 rows × 7 columns

In [7]:
```
# For checking null values

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

Out[7]:
```
movie_id    0
title       0
overview    3
genres      0
keywords    0
cast        0
crew        0
dtype: int64
```

In [8]:
```
# Drop the null value using this function and iplace for parmanent change

movies.dropna(inplace=True)
```

In [9]:
```
# Checking this is good

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

Out[9]:
```
movie_id    0
title       0
overview    0
genres      0
keywords    0
cast        0
crew        0
dtype: int64
```

In [10]:
```
# iloc for checking specific row values

movies.iloc[0].genres
```

Out[10]:
```
'[{"id": 28, "name": "Action"}, {"id": 12, "name": "Adventure"}, {"id": 14, "name": "Fantasy"}, {"id": 878, "name": "Science Fiction"}]'
```

In [11]:
```
# In this column we have lots of details and these all of in string forment so first of all choose specific data and change in
# '[{"id": 28, "name": "Action"}, {"id": 12, "name": "Adventure"}, {"id": 14, "name": "Fantasy"}, {"id": 878, "name": "Science
# change like this -- [Action, Adventure,FFantasy,Scifi]
```

Data Transformation

The genres, keywords, cast, and crew columns contain data in string format, which needs to be transformed into a list format.

In [12]:
```
# This code for change string into a list

import ast
ast.literal_eval('[{"id": 28, "name": "Action"}, {"id": 12, "name": "Adventure"}, {"id": 14, "name": "Fantasy"}, {"id": 878, "
```

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

In [13]:
```
# The provided code defines a function convert that takes a string representation of a list of dictionaries (obj) as input.
# It uses ast.literal_eval to safely parse this string into an actual list of dictionaries.
#The function then iterates through each dictionary in the list, extracts the value associated with the key 'name', and append
#Finally, the function returns the list L containing all the extracted 'name' values.
```

```
#This code is useful for converting a JSON-like string input into a list of specific values from dictionaries.



def convert(obj):
    L = []
    for i in ast.literal_eval(obj):
        L.append(i['name'])
    return L
```

In [14]:
```
# Apply convert function to genres column in movies DataFrame.

movies['genres'] = movies['genres'].apply(convert)
```

In [15]:
```
movies.head()
```

Out[15]:

	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]	[{"id": 1463, "name": "culture clash"}, {"id":...	[{"cast_id": 242, "character": "Jake Sully", "...	[{"credit_id": "52fe48009251416c750aca23", "de...
1	285	Pirates of the Caribbean: At World's End	Captain Barbossa, long believed to be dead, ha...	[Adventure, Fantasy, Action]	[{"id": 270, "name": "ocean"}, {"id": 726, "na...	[{"cast_id": 4, "character": "Captain Jack Spa...	[{"credit_id": "52fe4232c3a36847f800b579", "de...
2	206647	Spectre	A cryptic message from Bond’s past sends him o...	[Action, Adventure, Crime]	[{"id": 470, "name": "spy"}, {"id": 818, "name...	[{"cast_id": 1, "character": "James Bond", "cr...	[{"credit_id": "54805967c3a36829b5002c41", "de...
3	49026	The Dark Knight Rises	Following the death of District Attorney Harve...	[Action, Crime, Drama, Thriller]	[{"id": 849, "name": "dc comics"}, {"id": 853,...	[{"cast_id": 2, "character": "Bruce Wayne / Ba...	[{"credit_id": "52fe4781c3a36847f81398c3", "de...
4	49529	John Carter	John Carter is a war-weary, former military ca...	[Action, Adventure, Science Fiction]	[{"id": 818, "name": "based on novel"}, {"id":...	[{"cast_id": 5, "character": "John Carter", "c...	[{"credit_id": "52fe479ac3a36847f813eaa3", "de...

In [16]:
```
# Apply convert function to keywords column in movies DataFrame.

movies['keywords'] = movies['keywords'].apply(convert)
```

In [17]:
```
movies.head()
```

Out[17]:

	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...	[{"cast_id": 242, "character": "Jake Sully", "...	[{"credit_id": "52fe48009251416c750aca23", "de...
1	285	Pirates of the Caribbean: At World's End	Captain Barbossa, long believed to be dead, ha...	[Adventure, Fantasy, Action]	[ocean, drug abuse, exotic island, east india ...	[{"cast_id": 4, "character": "Captain Jack Spa...	[{"credit_id": "52fe4232c3a36847f800b579", "de...
2	206647	Spectre	A cryptic message from Bond’s past sends him o...	[Action, Adventure, Crime]	[spy, based on novel, secret agent, sequel, mi...	[{"cast_id": 1, "character": "James Bond", "cr...	[{"credit_id": "54805967c3a36829b5002c41", "de...
3	49026	The Dark Knight Rises	Following the death of District Attorney Harve...	[Action, Crime, Drama, Thriller]	[dc comics, crime fighter, terrorist, secret i...	[{"cast_id": 2, "character": "Bruce Wayne / Ba...	[{"credit_id": "52fe4781c3a36847f81398c3", "de...
4	49529	John Carter	John Carter is a war-weary, former military ca...	[Action, Adventure, Science Fiction]	[based on novel, mars, medallion, space travel...	[{"cast_id": 5, "character": "John Carter", "c...	[{"credit_id": "52fe479ac3a36847f813eaa3", "de...

In [18]:

```
# The provided code defines a function convert3 that takes a string representation of a list of dictionaries (obj) as input.
# It uses ast.literal_eval to safely parse this string into an actual list of dictionaries.
# The function initializes an empty list L and a counter set to 0.
# It then iterates through each dictionary in the parsed list, appending the value associated with the key 'name' to L and inc
# If the counter reaches 3, the loop breaks, stopping further additions to L. The function returns L, containing up to three '



import ast

def convert3(obj):
    L = []
    counter = 0
    for i in ast.literal_eval(obj):
```

```
        if counter != 3:
            L.append(i['name'])
            counter += 1
        else:
            break
    return L
```

In [19]:
```
# The code applies the convert3 function to the cast column of the movies DataFrame, transforming each entry to a list of up t

movies['cast'] = movies['cast'].apply(convert3)
```

In [20]:
```
movies.head()
```

Out[20]:

	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]	[{"credit_id": "52fe48009251416c750aca23", "de...
1	285	Pirates of the Caribbean: At World's End	Captain Barbossa, long believed to be dead, ha...	[Adventure, Fantasy, Action]	[ocean, drug abuse, exotic island, east india ...	[Johnny Depp, Orlando Bloom, Keira Knightley]	[{"credit_id": "52fe4232c3a36847f800b579", "de...
2	206647	Spectre	A cryptic message from Bond’s past sends him o...	[Action, Adventure, Crime]	[spy, based on novel, secret agent, sequel, mi...	[Daniel Craig, Christoph Waltz, Léa Seydoux]	[{"credit_id": "54805967c3a36829b5002c41", "de...
3	49026	The Dark Knight Rises	Following the death of District Attorney Harve...	[Action, Crime, Drama, Thriller]	[dc comics, crime fighter, terrorist, secret i...	[Christian Bale, Michael Caine, Gary Oldman]	[{"credit_id": "52fe4781c3a36847f81398c3", "de...
4	49529	John Carter	John Carter is a war-weary, former military ca...	[Action, Adventure, Science Fiction]	[based on novel, mars, medallion, space travel...	[Taylor Kitsch, Lynn Collins, Samantha Morton]	[{"credit_id": "52fe479ac3a36847f813eaa3", "de...

In [21]:
```
# Retrieve the first entry of the crew column in movies.

movies['crew'][0]
```

Out[21]:
'[{"credit_id": "52fe48009251416c750aca23", "department": "Editing", "gender": 0, "id": 1721, "job": "Editor", "name": "Steph
en E. Rivkin"}, {"credit_id": "539c47ecc3a36810e3001f87", "department": "Art", "gender": 2, "id": 496, "job": "Production Des
ign", "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", "depart
ment": "Production", "gender": 1, "id": 1262, "job": "Casting", "name": "Mali Finn"}, {"credit_id": "5544ee3b925141499f0008f
c", "department": "Sound", "gender": 2, "id": 1729, "job": "Original Music Composer", "name": "James Horner"}, {"credit_id":
"52fe48009251416c750ac9c3", "department": "Directing", "gender": 2, "id": 2710, "job": "Director", "name": "James Cameron"},
{"credit_id": "52fe48009251416c750ac9d9", "department": "Writing", "gender": 2, "id": 2710, "job": "Writer", "name": "James C
ameron"}, {"credit_id": "52fe48009251416c750aca17", "department": "Editing", "gender": 2, "id": 2710, "job": "Editor", "nam
e": "James Cameron"}, {"credit_id": "52fe48009251416c750aca29", "department": "Production", "gender": 2, "id": 2710, "job":
"Producer", "name": "James Cameron"}, {"credit_id": "52fe48009251416c750aca3f", "department": "Writing", "gender": 2, "id": 2
710, "job": "Screenplay", "name": "James Cameron"}, {"credit_id": "539c4987c3a36810ba0021a4", "department": "Art", "gender":
2, "id": 7236, "job": "Art Direction", "name": "Andrew Menzies"}, {"credit_id": "549598c3c3a3686ae9004383", "department": "Vi
sual Effects", "gender": 0, "id": 6690, "job": "Visual Effects Producer", "name": "Jill Brooks"}, {"credit_id": "52fe48009251
416c750aca4b", "department": "Production", "gender": 1, "id": 6347, "job": "Casting", "name": "Margery Simkin"}, {"credit_i
d": "570b6f419251417da70032fe", "department": "Art", "gender": 2, "id": 6878, "job": "Supervising Art Director", "name": "Kev
in Ishioka"}, {"credit_id": "5495a0fac3a3686ae9004468", "department": "Sound", "gender": 0, "id": 6883, "job": "Music Edito
r", "name": "Dick Bernstein"}, {"credit_id": "54959706c3a3686af3003e81", "department": "Sound", "gender": 0, "id": 8159, "jo
b": "Sound Effects Editor", "name": "Shannon Mills"}, {"credit_id": "54491d58c3a3680fb1001ccb", "department": "Sound", "gende
r": 0, "id": 8160, "job": "Foley", "name": "Dennie Thorpe"}, {"credit_id": "54491d6cc3a3680fa5001b2c", "department": "Sound",
"gender": 0, "id": 8163, "job": "Foley", "name": "Jana Vance"}, {"credit_id": "52fe48009251416c750aca57", "department": "Cost
ume & Make-Up", "gender": 1, "id": 8527, "job": "Costume Design", "name": "Deborah Lynn Scott"}, {"credit_id": "52fe480092514
16c750aca2f", "department": "Production", "gender": 2, "id": 8529, "job": "Producer", "name": "Jon Landau"}, {"credit_id": "5
39c4937c3a36810ba002194", "department": "Art", "gender": 0, "id": 9618, "job": "Art Direction", "name": "Sean Haworth"}, {"cr
edit_id": "539c49b6c3a36810c10020e6", "department": "Art", "gender": 1, "id": 12653, "job": "Set Decoration", "name": "Kim Si
nclair"}, {"credit_id": "570b6f2f9251413a0e00020d", "department": "Art", "gender": 1, "id": 12653, "job": "Supervising Art Di
rector", "name": "Kim Sinclair"}, {"credit_id": "54491a6c0e0a26748c001b19", "department": "Art", "gender": 2, "id": 14350, "j
ob": "Set Designer", "name": "Richard F. Mays"}, {"credit_id": "56928cf4c3a3684cff0025c4", "department": "Production", "gende
r": 1, "id": 20294, "job": "Executive Producer", "name": "Laeta Kalogridis"}, {"credit_id": "52fe48009251416c750aca51", "depa
rtment": "Costume & Make-Up", "gender": 0, "id": 17675, "job": "Costume Design", "name": "Mayes C. Rubeo"}, {"credit_id": "52
fe48009251416c750aca11", "department": "Camera", "gender": 2, "id": 18265, "job": "Director of Photography", "name": "Mauro F
iore"}, {"credit_id": "5449194d0e0a26748f001b39", "department": "Art", "gender": 0, "id": 42281, "job": "Set Designer", "nam
e": "Scott Herbertson"}, {"credit_id": "52fe48009251416c750aca05", "department": "Crew", "gender": 0, "id": 42288, "job": "St
unts", "name": "Woody Schultz"}, {"credit_id": "5592aefb92514152de0010f5", "department": "Costume & Make-Up", "gender": 0, "i
d": 29067, "job": "Makeup Artist", "name": "Linda DeVetta"}, {"credit_id": "5592afa492514152de00112c", "department": "Costume
& Make-Up", "gender": 0, "id": 29067, "job": "Hairstylist", "name": "Linda DeVetta"}, {"credit_id": "54959ed592514130fc002e5
d", "department": "Camera", "gender": 2, "id": 33302, "job": "Camera Operator", "name": "Richard Bluck"}, {"credit_id": "539c
4891c3a36810ba002147", "department": "Art", "gender": 2, "id": 33303, "job": "Art Direction", "name": "Simon Bright"}, {"cred
it_id": "54959c069251417a81001f3a", "department": "Visual Effects", "gender": 0, "id": 113145, "job": "Visual Effects Supervi

sor", "name": "Richard Martin"}, {"credit_id": "54959a0dc3a3680ff5002c8d", "department": "Crew", "gender": 2, "id": 58188, "job": "Visual Effects Editor", "name": "Steve R. Moore"}, {"credit_id": "52fe48009251416c750aca1d", "department": "Editing", "gender": 2, "id": 58871, "job": "Editor", "name": "John Refoua"}, {"credit_id": "54491a4dc3a3680fc30018ca", "department": "Art", "gender": 0, "id": 92359, "job": "Set Designer", "name": "Karl J. Martin"}, {"credit_id": "52fe48009251416c750aca35", "department": "Camera", "gender": 1, "id": 72201, "job": "Director of Photography", "name": "Chiling Lin"}, {"credit_id": "52fe48009251416c750ac9ff", "department": "Crew", "gender": 0, "id": 89714, "job": "Stunts", "name": "Ilram Choi"}, {"credit_id": "54959c529251416e2b004394", "department": "Visual Effects", "gender": 2, "id": 93214, "job": "Visual Effects Supervisor", "name": "Steven Quale"}, {"credit_id": "54491edf0e0a267489001c37", "department": "Crew", "gender": 1, "id": 122607, "job": "Dialect Coach", "name": "Carla Meyer"}, {"credit_id": "539c485bc3a368653d001a3a", "department": "Art", "gender": 2, "id": 132585, "job": "Art Direction", "name": "Nick Bassett"}, {"credit_id": "539c4903c3a368653d001a74", "department": "Art", "gender": 0, "id": 132596, "job": "Art Direction", "name": "Jill Cormack"}, {"credit_id": "539c4967c3a368653d001a94", "department": "Art", "gender": 0, "id": 132604, "job": "Art Direction", "name": "Andy McLaren"}, {"credit_id": "52fe48009251416c750aca45", "department": "Crew", "gender": 0, "id": 236696, "job": "Motion Capture Artist", "name": "Terry Notary"}, {"credit_id": "54959e02c3a3680fc60027d2", "department": "Crew", "gender": 2, "id": 956198, "job": "Stunt Coordinator", "name": "Garrett Warren"}, {"credit_id": "54959ca3c3a3686ae300438c", "department": "Visual Effects", "gender": 2, "id": 957874, "job": "Visual Effects Supervisor", "name": "Jonathan Rothbart"}, {"credit_id": "570b6f519251412c74001b2f", "department": "Art", "gender": 0, "id": 957889, "job": "Supervising Art Director", "name": "Stefan Dechant"}, {"credit_id": "570b6f62c3a3680b77007460", "department": "Art", "gender": 2, "id": 959555, "job": "Supervising Art Director", "name": "Todd Cherniawsky"}, {"credit_id": "539c4a3ac3a36810da0021cc", "department": "Production", "gender": 0, "id": 1016177, "job": "Casting", "name": "Miranda Rivers"}, {"credit_id": "539c482cc3a36810c1002062", "department": "Art", "gender": 0, "id": 1032536, "job": "Production Design", "name": "Robert Stromberg"}, {"credit_id": "539c4b65c3a36810c9002125", "department": "Costume & Make-Up", "gender": 2, "id": 1071680, "job": "Costume Design", "name": "John Harding"}, {"credit_id": "54959e6692514130fc002e4e", "department": "Camera", "gender": 0, "id": 1177364, "job": "Steadicam Operator", "name": "Roberto De Angelis"}, {"credit_id": "539c49f1c3a368653d001aac", "department": "Costume & Make-Up", "gender": 2, "id": 1202850, "job": "Makeup Department Head", "name": "Mike Smithson"}, {"credit_id": "5495999ec3a3686ae100460c", "department": "Visual Effects", "gender": 0, "id": 1204668, "job": "Visual Effects Producer", "name": "Alain Lalanne"}, {"credit_id": "54959cdfc3a3681153002729", "department": "Visual Effects", "gender": 0, "id": 1206410, "job": "Visual Effects Supervisor", "name": "Lucas Salton"}, {"credit_id": "549596239251417a81001eae", "department": "Crew", "gender": 0, "id": 1234266, "job": "Post Production Supervisor", "name": "Janace Tashjian"}, {"credit_id": "54959c859251416e1e003efe", "department": "Visual Effects", "gender": 0, "id": 1271932, "job": "Visual Effects Supervisor", "name": "Stephen Rosenbaum"}, {"credit_id": "5592af28c3a368775a00105f", "department": "Costume & Make-Up", "gender": 0, "id": 1310064, "job": "Makeup Artist", "name": "Frankie Karena"}, {"credit_id": "539c4adfc3a36810e300203b", "department": "Costume & Make-Up", "gender": 1, "id": 1319844, "job": "Costume Supervisor", "name": "Lisa Lovaas"}, {"credit_id": "54959b579251416e2b004371", "department": "Visual Effects", "gender": 0, "id": 1327028, "job": "Visual Effects Supervisor", "name": "Jonathan Fawkner"}, {"credit_id": "539c48a7c3a36810b5001fa7", "department": "Art", "gender": 0, "id": 1330561, "job": "Art Direction", "name": "Robert Bavin"}, {"credit_id": "539c4a71c3a36810da0021e0", "department": "Costume & Make-Up", "gender": 0, "id": 1330567, "job": "Costume Supervisor", "name": "Anthony Almaraz"}, {"credit_id": "539c4a8ac3a36810ba0021e4", "department": "Costume & Make-Up", "gender": 0, "id": 1330570, "job": "Costume Supervisor", "name": "Carolyn M. Fenton"}, {"credit_id": "539c4ab6c3a36810da0021f0", "department": "Costume & Make-Up", "gender": 0, "id": 1330574, "job": "Costume Supervisor", "name": "Beth Koenigsberg"}, {"credit_id": "54491ab70e0a267480001ba2", "department": "Art", "gender": 0, "id": 1336191, "job": "Set Designer", "name": "Sam Page"}, {"credit_id": "544919d9c3a3680fc30018bd", "department": "Art", "gender": 0, "id": 1339441, "job": "Set Designer", "name": "Tex Kadonaga"}, {"credit_id": "54491cf50e0a267483001b0c", "department": "Editing", "gender": 0, "id": 1352422, "job": "D

ialogue Editor", "name": "Kim Foscato"}, {"credit_id": "544919f40e0a26748c001b09", "department": "Art", "gender": 0, "id": 13
52962, "job": "Set Designer", "name": "Tammy S. Lee"}, {"credit_id": "5495a115c3a3680ff5002d71", "department": "Crew", "gende
r": 0, "id": 1357070, "job": "Transportation Coordinator", "name": "Denny Caira"}, {"credit_id": "5495a12f92514130fc002e94",
"department": "Crew", "gender": 0, "id": 1357071, "job": "Transportation Coordinator", "name": "James Waitkus"}, {"credit_i
d": "5495976fc3a36811530026b0", "department": "Sound", "gender": 0, "id": 1360103, "job": "Supervising Sound Editor", "name":
"Addison Teague"}, {"credit_id": "54491837c3a3680fb1001c5a", "department": "Art", "gender": 2, "id": 1376887, "job": "Set Des
igner", "name": "C. Scott Baker"}, {"credit_id": "54491878c3a3680fb4001c9d", "department": "Art", "gender": 0, "id": 1376888,
"job": "Set Designer", "name": "Luke Caska"}, {"credit_id": "544918dac3a3680fa5001ae0", "department": "Art", "gender": 0, "i
d": 1376889, "job": "Set Designer", "name": "David Chow"}, {"credit_id": "544919110e0a267486001b68", "department": "Art", "ge
nder": 0, "id": 1376890, "job": "Set Designer", "name": "Jonathan Dyer"}, {"credit_id": "54491967c3a3680faa001b5e", "departme
nt": "Art", "gender": 0, "id": 1376891, "job": "Set Designer", "name": "Joseph Hiura"}, {"credit_id": "54491997c3a3680fb1001c
8a", "department": "Art", "gender": 0, "id": 1376892, "job": "Art Department Coordinator", "name": "Rebecca Jellie"}, {"credi
t_id": "544919ba0e0a26748f001b42", "department": "Art", "gender": 0, "id": 1376893, "job": "Set Designer", "name": "Robert An
drew Johnson"}, {"credit_id": "54491b1dc3a3680faa001b8c", "department": "Art", "gender": 0, "id": 1376895, "job": "Assistant
Art Director", "name": "Mike Stassi"}, {"credit_id": "54491b79c3a3680fbb001826", "department": "Art", "gender": 0, "id": 1376
897, "job": "Construction Coordinator", "name": "John Villarino"}, {"credit_id": "54491baec3a3680fb4001ce6", "department": "A
rt", "gender": 2, "id": 1376898, "job": "Assistant Art Director", "name": "Jeffrey Wisniewski"}, {"credit_id": "54491d2fc3a36
80fb4001d07", "department": "Editing", "gender": 0, "id": 1376899, "job": "Dialogue Editor", "name": "Cheryl Nardi"}, {"credi
t_id": "54491d86c3a3680fa5001b2f", "department": "Editing", "gender": 0, "id": 1376901, "job": "Dialogue Editor", "name": "Ma
rshall Winn"}, {"credit_id": "54491d9dc3a3680faa001bb0", "department": "Sound", "gender": 0, "id": 1376902, "job": "Supervisi
ng Sound Editor", "name": "Gwendolyn Yates Whittle"}, {"credit_id": "54491dc10e0a267486001bce", "department": "Sound", "gende
r": 0, "id": 1376903, "job": "Sound Re-Recording Mixer", "name": "William Stein"}, {"credit_id": "54491f500e0a26747c001c07",
"department": "Crew", "gender": 0, "id": 1376909, "job": "Choreographer", "name": "Lula Washington"}, {"credit_id": "54959923
9251412c4e002a2e", "department": "Visual Effects", "gender": 0, "id": 1391692, "job": "Visual Effects Producer", "name": "Chr
is Del Conte"}, {"credit_id": "54959d54c3a36831b8001d9a", "department": "Visual Effects", "gender": 2, "id": 1391695, "job":
"Visual Effects Supervisor", "name": "R. Christopher White"}, {"credit_id": "54959bdf9251412c4e002a66", "department": "Visual
Effects", "gender": 0, "id": 1394070, "job": "Visual Effects Supervisor", "name": "Dan Lemmon"}, {"credit_id": "5495971d92514
132ed002922", "department": "Sound", "gender": 0, "id": 1394129, "job": "Sound Effects Editor", "name": "Tim Nielsen"}, {"cre
dit_id": "5592b25792514152cc0011aa", "department": "Crew", "gender": 0, "id": 1394286, "job": "CG Supervisor", "name": "Micha
el Mulholland"}, {"credit_id": "54959a329251416e2b004355", "department": "Crew", "gender": 0, "id": 1394750, "job": "Visual E
ffects Editor", "name": "Thomas Nittmann"}, {"credit_id": "54959d6dc3a3686ae9004401", "department": "Visual Effects", "gende
r": 0, "id": 1394755, "job": "Visual Effects Supervisor", "name": "Edson Williams"}, {"credit_id": "5495a08fc3a3686ae300441
c", "department": "Editing", "gender": 0, "id": 1394953, "job": "Digital Intermediate", "name": "Christine Carr"}, {"credit_i
d": "55402d659251413d6d000249", "department": "Visual Effects", "gender": 0, "id": 1395269, "job": "Visual Effects Superviso
r", "name": "John Bruno"}, {"credit_id": "54959e7b9251416e1e003f3e", "department": "Camera", "gender": 0, "id": 1398970, "jo
b": "Steadicam Operator", "name": "David Emmerichs"}, {"credit_id": "54959734c3a3686ae10045e0", "department": "Sound", "gende
r": 0, "id": 1400906, "job": "Sound Effects Editor", "name": "Christopher Scarabosio"}, {"credit_id": "549595dd92514130fc002d
79", "department": "Production", "gender": 0, "id": 1401784, "job": "Production Supervisor", "name": "Jennifer Teves"}, {"cre
dit_id": "549596009251413af70028cc", "department": "Production", "gender": 0, "id": 1401785, "job": "Production Manager", "na
me": "Brigitte Yorke"}, {"credit_id": "549596e892514130fc002d99", "department": "Sound", "gender": 0, "id": 1401786, "job":
"Sound Effects Editor", "name": "Ken Fischer"}, {"credit_id": "549598229251412c4e002a1c", "department": "Crew", "gender": 0,

"id": 1401787, "job": "Special Effects Coordinator", "name": "Iain Hutton"}, {"credit_id": "549598349251416e2b00432b", "depar
tment": "Crew", "gender": 0, "id": 1401788, "job": "Special Effects Coordinator", "name": "Steve Ingram"}, {"credit_id": "549
59905c3a3686ae3004324", "department": "Visual Effects", "gender": 0, "id": 1401789, "job": "Visual Effects Producer", "name":
"Joyce Cox"}, {"credit_id": "5495994b92514132ed002951", "department": "Visual Effects", "gender": 0, "id": 1401790, "job": "V
isual Effects Producer", "name": "Jenny Foster"}, {"credit_id": "549599cbc3a3686ae1004613", "department": "Crew", "gender":
0, "id": 1401791, "job": "Visual Effects Editor", "name": "Christopher Marino"}, {"credit_id": "549599f2c3a3686ae100461e", "d
epartment": "Crew", "gender": 0, "id": 1401792, "job": "Visual Effects Editor", "name": "Jim Milton"}, {"credit_id": "54959a5
1c3a3686af3003eb5", "department": "Visual Effects", "gender": 0, "id": 1401793, "job": "Visual Effects Producer", "name": "Cy
ndi Ochs"}, {"credit_id": "54959a7cc3a36811530026f4", "department": "Crew", "gender": 0, "id": 1401794, "job": "Visual Effect
s Editor", "name": "Lucas Putnam"}, {"credit_id": "54959b91c3a3680ff5002cb4", "department": "Visual Effects", "gender": 0, "i
d": 1401795, "job": "Visual Effects Supervisor", "name": "Anthony \'Max\' Ivins"}, {"credit_id": "54959bb69251412c4e002a5f",
"department": "Visual Effects", "gender": 0, "id": 1401796, "job": "Visual Effects Supervisor", "name": "John Knoll"}, {"cred
it_id": "54959cbbc3a3686ae3004391", "department": "Visual Effects", "gender": 2, "id": 1401799, "job": "Visual Effects Superv
isor", "name": "Eric Saindon"}, {"credit_id": "54959d06c3a3686ae90043f6", "department": "Visual Effects", "gender": 0, "id":
1401800, "job": "Visual Effects Supervisor", "name": "Wayne Stables"}, {"credit_id": "54959d259251416e1e003f11", "departmen
t": "Visual Effects", "gender": 0, "id": 1401801, "job": "Visual Effects Supervisor", "name": "David Stinnett"}, {"credit_i
d": "54959db49251413af7002975", "department": "Visual Effects", "gender": 0, "id": 1401803, "job": "Visual Effects Superviso
r", "name": "Guy Williams"}, {"credit_id": "54959de4c3a3681153002750", "department": "Crew", "gender": 0, "id": 1401804, "jo
b": "Stunt Coordinator", "name": "Stuart Thorp"}, {"credit_id": "54959ef2c3a3680fc60027f2", "department": "Lighting", "gende
r": 0, "id": 1401805, "job": "Best Boy Electric", "name": "Giles Coburn"}, {"credit_id": "54959f07c3a3680fc60027f9", "departm
ent": "Camera", "gender": 2, "id": 1401806, "job": "Still Photographer", "name": "Mark Fellman"}, {"credit_id": "54959f47c3a3
681153002774", "department": "Lighting", "gender": 0, "id": 1401807, "job": "Lighting Technician", "name": "Scott Sprague"},
{"credit_id": "54959f8cc3a36831b8001df2", "department": "Visual Effects", "gender": 0, "id": 1401808, "job": "Animation Direc
tor", "name": "Jeremy Hollobon"}, {"credit_id": "54959fa0c3a36831b8001dfb", "department": "Visual Effects", "gender": 0, "i
d": 1401809, "job": "Animation Director", "name": "Orlando Meunier"}, {"credit_id": "54959fb6c3a3686af3003f54", "department":
"Visual Effects", "gender": 0, "id": 1401810, "job": "Animation Director", "name": "Taisuke Tanimura"}, {"credit_id": "54959f
d2c3a36831b8001e02", "department": "Costume & Make-Up", "gender": 0, "id": 1401812, "job": "Set Costumer", "name": "Lilia Mis
hel Acevedo"}, {"credit_id": "54959ff9c3a3686ae300440c", "department": "Costume & Make-Up", "gender": 0, "id": 1401814, "jo
b": "Set Costumer", "name": "Alejandro M. Hernandez"}, {"credit_id": "5495a0ddc3a3686ae10046fe", "department": "Editing", "ge
nder": 0, "id": 1401815, "job": "Digital Intermediate", "name": "Marvin Hall"}, {"credit_id": "5495a1f7c3a3686ae3004443", "de
partment": "Production", "gender": 0, "id": 1401816, "job": "Publicist", "name": "Judy Alley"}, {"credit_id": "5592b29fc3a368
69d100002f", "department": "Crew", "gender": 0, "id": 1418381, "job": "CG Supervisor", "name": "Mike Perry"}, {"credit_id":
"5592b23a9251415df8001081", "department": "Crew", "gender": 0, "id": 1426854, "job": "CG Supervisor", "name": "Andrew Morle
y"}, {"credit_id": "55491e1192514104c40002d8", "department": "Art", "gender": 0, "id": 1438901, "job": "Conceptual Design",
"name": "Seth Engstrom"}, {"credit_id": "5525d5809251417276002b06", "department": "Crew", "gender": 0, "id": 1447362, "job":
"Visual Effects Art Director", "name": "Eric Oliver"}, {"credit_id": "554427ca925141586500312a", "department": "Visual Effect
s", "gender": 0, "id": 1447503, "job": "Modeling", "name": "Matsune Suzuki"}, {"credit_id": "551906889251415aab001c88", "depa
rtment": "Art", "gender": 0, "id": 1447524, "job": "Art Department Manager", "name": "Paul Tobin"}, {"credit_id": "5592af8492
514152cc0010de", "department": "Costume & Make-Up", "gender": 0, "id": 1452643, "job": "Hairstylist", "name": "Roxane Griffi
n"}, {"credit_id": "553d3c109251415852001318", "department": "Lighting", "gender": 0, "id": 1453938, "job": "Lighting Artis
t", "name": "Arun Ram-Mohan"}, {"credit_id": "5592af4692514152d5001355", "department": "Costume & Make-Up", "gender": 0, "i

d": 1457305, "job": "Makeup Artist", "name": "Georgia Lockhart-Adams"}, {"credit_id": "5592b2eac3a36877470012a5", "departmen
t": "Crew", "gender": 0, "id": 1466035, "job": "CG Supervisor", "name": "Thrain Shadbolt"}, {"credit_id": "5592b032c3a3687745
0015f1", "department": "Crew", "gender": 0, "id": 1483220, "job": "CG Supervisor", "name": "Brad Alexander"}, {"credit_id":
"5592b05592514152d80012f6", "department": "Crew", "gender": 0, "id": 1483221, "job": "CG Supervisor", "name": "Shadi Almassiz
adeh"}, {"credit_id": "5592b090c3a36877570010b5", "department": "Crew", "gender": 0, "id": 1483222, "job": "CG Supervisor",
"name": "Simon Clutterbuck"}, {"credit_id": "5592b0dbc3a368774b00112c", "department": "Crew", "gender": 0, "id": 1483223, "jo
b": "CG Supervisor", "name": "Graeme Demmocks"}, {"credit_id": "5592b0fe92514152db0010c1", "department": "Crew", "gender": 0,
"id": 1483224, "job": "CG Supervisor", "name": "Adrian Fernandes"}, {"credit_id": "5592b11f9251415df8001059", "department":
"Crew", "gender": 0, "id": 1483225, "job": "CG Supervisor", "name": "Mitch Gates"}, {"credit_id": "5592b15dc3a3687745001645",
"department": "Crew", "gender": 0, "id": 1483226, "job": "CG Supervisor", "name": "Jerry Kung"}, {"credit_id": "5592b18e92514
1645a0004ae", "department": "Crew", "gender": 0, "id": 1483227, "job": "CG Supervisor", "name": "Andy Lomas"}, {"credit_id":
"5592b1bfc3a368775d0010e7", "department": "Crew", "gender": 0, "id": 1483228, "job": "CG Supervisor", "name": "Sebastian Mari
no"}, {"credit_id": "5592b2049251415df8001078", "department": "Crew", "gender": 0, "id": 1483229, "job": "CG Supervisor", "na
me": "Matthias Menz"}, {"credit_id": "5592b27b92514152d800136a", "department": "Crew", "gender": 0, "id": 1483230, "job": "CG
Supervisor", "name": "Sergei Nevshupov"}, {"credit_id": "5592b2c3c3a36869e800003c", "department": "Crew", "gender": 0, "id":
1483231, "job": "CG Supervisor", "name": "Philippe Rebours"}, {"credit_id": "5592b317c3a36877470012af", "department": "Crew",
"gender": 0, "id": 1483232, "job": "CG Supervisor", "name": "Michael Takarangi"}, {"credit_id": "5592b345c3a36877470012bb",
"department": "Crew", "gender": 0, "id": 1483233, "job": "CG Supervisor", "name": "David Weitzberg"}, {"credit_id": "5592b37c
c3a368775100113b", "department": "Crew", "gender": 0, "id": 1483234, "job": "CG Supervisor", "name": "Ben White"}, {"credit_i
d": "573c8e2f9251413f5d000094", "department": "Crew", "gender": 1, "id": 1621932, "job": "Stunts", "name": "Min Windle"}]'

In [22]:
```
# The provided code defines a function fetch_director that takes a string representation of a list of dictionaries (obj) as in
# It uses ast.literal_eval to safely parse this string into an actual list of dictionaries.
# The function initializes an empty list L.
# It then iterates through each dictionary in the parsed list, checking if the value associated with the key 'job' is 'Directo
# When it finds a dictionary where 'job' is 'Director', it appends the corresponding 'name' value to L and breaks the loop.
# Finally, the function returns the list L, which contains the name of the director.

import ast

def fetch_director(obj):
    L = []
    for i in ast.literal_eval(obj):
        if i['job'] == 'Director':
            L.append(i['name'])
            break
    return L
```

In [23]:
```
# The code applies the fetch_director function to the crew column of the movies DataFrame, transforming each entry to a list c
```

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

In [24]:
```
movies.head()
```

Out[24]:

	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]
1	285	Pirates of the Caribbean: At World's End	Captain Barbossa, long believed to be dead, ha...	[Adventure, Fantasy, Action]	[ocean, drug abuse, exotic island, east india ...	[Johnny Depp, Orlando Bloom, Keira Knightley]	[Gore Verbinski]
2	206647	Spectre	A cryptic message from Bond’s past sends him o...	[Action, Adventure, Crime]	[spy, based on novel, secret agent, sequel, mi...	[Daniel Craig, Christoph Waltz, Léa Seydoux]	[Sam Mendes]
3	49026	The Dark Knight Rises	Following the death of District Attorney Harve...	[Action, Crime, Drama, Thriller]	[dc comics, crime fighter, terrorist, secret i...	[Christian Bale, Michael Caine, Gary Oldman]	[Christopher Nolan]
4	49529	John Carter	John Carter is a war-weary, former military ca...	[Action, Adventure, Science Fiction]	[based on novel, mars, medallion, space travel...	[Taylor Kitsch, Lynn Collins, Samantha Morton]	[Andrew Stanton]

In [25]:
```
movies['overview'][0]
```

Out[25]:
```
'In the 22nd century, a paraplegic Marine is dispatched to the moon Pandora on a unique mission, but becomes torn between fol
lowing orders and protecting an alien civilization.'
```

Creating Tags for Each Movie

A 'tags' column is created by combining the transformed data from the 'overview', 'genres', 'keywords', 'cast', and 'crew' columns.

In [26]:
```
# The code applies a lambda function to the overview column of the movies DataFrame, transforming each entry by splitting the

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

In [27]:
```
movies.head()
```

Out[27]:

	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]
2	206647	Spectre	[A, cryptic, message, from, Bond’s, past, send...	[Action, Adventure, Crime]	[spy, based on novel, secret agent, sequel, mi...	[Daniel Craig, Christoph Waltz, Léa Seydoux]	[Sam Mendes]
3	49026	The Dark Knight Rises	[Following, the, death, of, District, Attorney...	[Action, Crime, Drama, Thriller]	[dc comics, crime fighter, terrorist, secret i...	[Christian Bale, Michael Caine, Gary Oldman]	[Christopher Nolan]
4	49529	John Carter	[John, Carter, is, a, war-weary,, former, mili...	[Action, Adventure, Science Fiction]	[based on novel, mars, medallion, space travel...	[Taylor Kitsch, Lynn Collins, Samantha Morton]	[Andrew Stanton]

In [28]:
```
#Sam Worthington
#SamWorthington
```

In [29]:
```
# The code applies a lambda function to the genres, keywords, cast, and crew columns of the movies DataFrame.
# For each of these columns, the lambda function iterates through the list of strings in each entry, removing any spaces withi
# This effectively removes all spaces from each name or term in these lists.
# The transformed lists are then reassigned to their respective columns in the DataFrame.
# This operation ensures that the names and terms in these columns are formatted without spaces, likely for consistency or fur


movies['genres'] = movies['genres'].apply(lambda x:[i.replace(" ","") for i in x])
```

```
movies['keywords'] = movies['keywords'].apply(lambda x:[i.replace(" ","") for i in x])
movies['cast'] = movies['cast'].apply(lambda x:[i.replace(" ","") for i in x])
movies['crew'] = movies['crew'].apply(lambda x:[i.replace(" ","") for i in x])
```

In [30]:
```
movies.head()
```

Out[30]:

	movie_id	title	overview	genres	keywords	cast	crew
0	19995	Avatar	[In, the, 22nd, century,, a, paraplegic, Marin...	[Action, Adventure, Fantasy, ScienceFiction]	[cultureclash, future, spacewar, spacecolony, ...	[SamWorthington, ZoeSaldana, SigourneyWeaver]	[JamesCameron]
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]	[GoreVerbinski]
2	206647	Spectre	[A, cryptic, message, from, Bond’s, past, send...	[Action, Adventure, Crime]	[spy, basedonnovel, secretagent, sequel, mi6, ...	[DanielCraig, ChristophWaltz, LéaSeydoux]	[SamMendes]
3	49026	The Dark Knight Rises	[Following, the, death, of, District, Attorney...	[Action, Crime, Drama, Thriller]	[dccomics, crimefighter, terrorist, secretiden...	[ChristianBale, MichaelCaine, GaryOldman]	[ChristopherNolan]
4	49529	John Carter	[John, Carter, is, a, war-weary,, former, mili...	[Action, Adventure, ScienceFiction]	[basedonnovel, mars, medallion, spacetravel, p...	[TaylorKitsch, LynnCollins, SamanthaMorton]	[AndrewStanton]

In [31]:
```
# Combine multiple columns into tags in movies DataFrame.

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

In [32]:
```
movies.head()
```

Out[32]:

	movie_id	title	overview	genres	keywords	cast	crew	tags
0	19995	Avatar	[In, the, 22nd, century,, a, paraplegic, Marin...	[Action, Adventure, Fantasy, ScienceFiction]	[cultureclash, future, spacewar, spacecolony, ...	[SamWorthington, ZoeSaldana, SigourneyWeaver]	[JamesCameron]	[In, the, 22nd, century,, a, paraplegic, Marin...
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]	[GoreVerbinski]	[Captain, Barbossa,, long, believed, to, be, d...
2	206647	Spectre	[A, cryptic, message, from, Bond’s, past, send...	[Action, Adventure, Crime]	[spy, basedonnovel, secretagent, sequel, mi6, ...	[DanielCraig, ChristophWaltz, LéaSeydoux]	[SamMendes]	[A, cryptic, message, from, Bond’s, past, send...
3	49026	The Dark Knight Rises	[Following, the, death, of, District, Attorney...	[Action, Crime, Drama, Thriller]	[dccomics, crimefighter, terrorist, secretiden...	[ChristianBale, MichaelCaine, GaryOldman]	[ChristopherNolan]	[Following, the, death, of, District, Attorney...
4	49529	John Carter	[John, Carter, is, a, war-weary,, former, mili...	[Action, Adventure, ScienceFiction]	[basedonnovel, mars, medallion, spacetravel, p...	[TaylorKitsch, LynnCollins, SamanthaMorton]	[AndrewStanton]	[John, Carter, is, a, war-weary,, former, mili...

In [33]:
```
new_df = movies[['movie_id','title','tags']]
```

In [34]:
```
new_df
```

Out[34]:

	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	49026	The Dark Knight Rises	[Following, the, death, of, District, Attorney...
4	49529	John Carter	[John, Carter, is, a, war-weary,, former, mili...
...	...	...	...
4804	9367	El Mariachi	[El, Mariachi, just, wants, to, play, his, gui...
4805	72766	Newlyweds	[A, newlywed, couple's, honeymoon, is, upended...
4806	231617	Signed, Sealed, Delivered	["Signed,, Sealed,, Delivered", introduces, a,...
4807	126186	Shanghai Calling	[When, ambitious, New, York, attorney, Sam, is...
4808	25975	My Date with Drew	[Ever, since, the, second, grade, when, he, fi...

4806 rows × 3 columns

In [35]:
```
# Apply a lambda function to join list elements in tags column into a single string in new_df.

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

Out[35]:
```
0       In the 22nd century, a paraplegic Marine is di...
1       Captain Barbossa, long believed to be dead, ha...
2       A cryptic message from Bond’s past sends him o...
3       Following the death of District Attorney Harve...
4       John Carter is a war-weary, former military ca...
                              ...
4804    El Mariachi just wants to play his guitar and ...
4805    A newlywed couple's honeymoon is upended by th...
4806    "Signed, Sealed, Delivered" introduces a dedic...
4807    When ambitious New York attorney Sam is sent t...
4808    Ever since the second grade when he first saw ...
Name: tags, Length: 4806, dtype: object
```

In [36]:
```
# The code updates the tags column in new_df by applying a lambda function that joins the list elements in each entry into a s

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

```
C:\Users\ROHIT RAI\AppData\Local\Temp\ipykernel_1400\3089450492.py: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-ve
rsus-a-copy
  new_df['tags'] = new_df['tags'].apply(lambda x:" ".join(x))
```

In [37]:
```
new_df.head()
```

Out[37]:

	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	49026	The Dark Knight Rises	Following the death of District Attorney Harve...
4	49529	John Carter	John Carter is a war-weary, former military ca...

In [38]:
```
# The code applies a lambda function to convert all text in the tags column to lowercase in new_df.

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

Out[38]:
```
0       in the 22nd century, a paraplegic marine is di...
1       captain barbossa, long believed to be dead, ha...
2       a cryptic message from bond’s past sends him o...
3       following the death of district attorney harve...
4       john carter is a war-weary, former military ca...
                              ...
4804    el mariachi just wants to play his guitar and ...
4805    a newlywed couple's honeymoon is upended by th...
4806    "signed, sealed, delivered" introduces a dedic...
4807    when ambitious new york attorney sam is sent t...
4808    ever since the second grade when he first saw ...
Name: tags, Length: 4806, dtype: object
```

In [39]:
```
# The code updates the tags column in new_df by applying a lambda function that converts all text to lowercase.

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

```
C:\Users\ROHIT RAI\AppData\Local\Temp\ipykernel_1400\3214958533.py: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:x.lower())
```

In [40]:
```
new_df.head()
```

Out[40]:

	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	49026	The Dark Knight Rises	following the death of district attorney harve...
4	49529	John Carter	john carter is a war-weary, former military ca...

In [41]:
```
!# pip install nltk
```

```
'#' is not recognized as an internal or external command,
operable program or batch file.
```

In [42]:
```
import nltk
```

In [43]:
```
# This code imports the PorterStemmer class from the NLTK library, which is used for stemming in natural language processing.
# It also downloads the required tokenization model using NLTK's download function. Finally, it initializes an instance of the

from nltk.stem import PorterStemmer
nltk.download('punkt')
ps = PorterStemmer()
```

```
[nltk_data] Downloading package punkt to C:\Users\ROHIT
[nltk_data]     RAI\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

In [44]:
```
# The provided code defines a function stem that takes a piece of text as input.
# It initializes an empty list y and then iterates through each word in the input text after splitting it.
# For each word, it applies stemming using the PorterStemmer instance ps, and appends the stemmed word to the list y.
# Finally, it joins the stemmed words into a single string separated by spaces and returns the result.
# This function essentially stems each word in a text input.

def stem(text):
    y = []

    for i in text.split():
```

```
        y.append(ps.stem(i))

    return " ".join(y)
```

In [45]:
```
# Apply stemming function to tags column in new_df.

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

```
C:\Users\ROHIT RAI\AppData\Local\Temp\ipykernel_1400\3213734980.py: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(stem)
```

In [46]:
```
new_df['tags'][0]
```

Out[46]:
```
'in the 22nd century, a parapleg marin is dispatch to the moon pandora on a uniqu mission, but becom torn between follow order and protect an alien civilization. action adventur fantasi sciencefict samworthington zoesaldana sigourneyweav jamescameron'
```

In [47]:
```
new_df['tags'][1]
```

Out[47]:
```
'captain barbossa, long believ to be dead, ha come back to life and is head to the edg of the earth with will turner and elizabeth swann. but noth is quit as it seems. adventur fantasi action johnnydepp orlandobloom keiraknightley goreverbinski'
```

Text Vectorization and Similarity Calculation

The text data in the 'tags' column is vectorized using CountVectorizer, and cosine similarity is calculated between the vectors.

In [49]:
```
# This code imports the CountVectorizer class from scikit-learn for text preprocessing.
# It initializes an instance of CountVectorizer.
# It sets max_features to limit the number of features to 5000 and stop_words to remove common English words during vectorizat

from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer()
```

```
cv = CountVectorizer(max_features=5000, stop_words='english')
```

In [50]:
```
# Convert text data into a numerical vector representation.

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

In [51]:
```
vectors
```

Out[51]:
```
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]], dtype=int64)
```

In [52]:
```
vectors[0]
```

Out[52]:
```
array([0, 0, 0, ..., 0, 0, 0], dtype=int64)
```

In [53]:
```
# The result of stemming the word "loving" using the PorterStemmer is "love".

ps.stem('loving')
```

Out[53]:
```
'love'
```

In [54]:
```
stem('in the 22nd century, a paraplegic marine is dispatched to the moon pandora on a unique mission, but becomes torn between
```

Out[54]:
```
'in the 22nd century, a parapleg marin is dispatch to the moon pandora on a uniqu mission, but becom torn between follow orde
r and protect an alien civilization. action adventur fantasi sciencefict samworthington zoesaldana sigourneyweav jamescamero
n'
```

In [55]:
```
# This line imports the cosine_similarity function from scikit-learn, which computes the cosine similarity between pairs of ve

from sklearn.metrics.pairwise import cosine_similarity
```

In [56]:
```
# This line computes the cosine similarity between all pairs of vectors in the vectors array, storing the result in the simila
```

```
similarity = cosine_similarity(vectors)
```

In [57]:
```
similarity.shape
```

Out[57]:
```
(4806, 4806)
```

In [58]:
```
#sorted(similarity[0],reverse = True)
#sorted(list(enumerate(similarity[0])),reverse = True,key=lambda x:x[1])[1:6]
```

Recommendation Function

A function to recommend movies based on the cosine similarity is implemented.

In [59]:
```
# The function recommend takes a movie title as input.
# First, it finds the index of the movie in the DataFrame new_df using the title.
# Then, it retrieves the cosine similarity scores of the input movie with all other movies from the similarity matrix.
# Next, it creates a list of tuples containing the index of each movie and its corresponding similarity score, sorted in desce
# It excludes the input movie itself by starting from the second element.
# Then, it iterates through the top 5 similar movies, printing their titles.
# Finally, it returns without explicitly returning any value.
# This function effectively recommends 5 movies similar to the input movie based on their cosine similarity scores, allowing u



def recommend(movie):
    movie_index = new_df[new_df['title'] == movie].index[0]
    distances = similarity[movie_index]
    movies_list = sorted(list(enumerate(distances)),reverse = True,key=lambda x:x[1])[1:6]

    for i in movies_list:
        print(new_df.iloc[i[0]].title)

    return
```

In [60]:
```
#new_df[new_df['title'] == 'Avatar'].index[0]
#new_df.iloc[2999].title
```

Conclusion

This Python-based movie recommender system processes movie data to extract relevant features, converts text data into vectors, and uses cosine similarity to recommend movies. The system is designed to provide recommendations based on the content similarity of movies.

Movie Recommendation Output

A function to recommend movies based on the cosine similarity is implemented. The recommendation function effectively provides personalized movie recommendations based on the content similarity of movies. This feature can be extremely useful for users looking for new movies to watch that are similar to their favorites. The combination of data preprocessing, text vectorization, and cosine similarity calculation ensures that the recommendations are relevant and based on comprehensive content analysis.

Functionality

Recommendation Function: recommend(movie)

Input: A movie title (e.g., 'Titanic').

Process: The function finds the index of the movie in the DataFrame. It calculates the cosine similarity between the movie's tags and the tags of all other movies in the dataset. It sorts the movies based on their similarity scores in descending order. It selects the top 5 movies with the highest similarity scores, excluding the input movie itself.

Output: Prints the titles of the top 5 recommended movies.

In [63]:
```
# Write The Movie Name/Title
```

```
recommend('Titanic')
```

```
Under the Same Moon
The Switch
Nowhere Boy
Baby Boy
The Notebook
```