```
import pandas as pd
import numpy as np


movies = pd.read_csv('tmdb_5000_movies.csv')
credits = pd.read_csv('tmdb_5000_credits.csv')


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

```
(1155, 23)
```

```
movies.head()
```

	budget	genres	homepage	id	keywords
0	237000000	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	http://www.avatarmovie.com/	19995	[{"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	245000000	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	http://www.sonypictures.com/movies/spectre/	206647	[{"id": 470, "name": "spy"}, {"id": 818, "name...
3	250000000	[{"id": 28, "name": "Action"}, {"id": 80, "nam...	http://www.thedarkknightrises.com/	49026	[{"id": 849, "name": "dc comics"}, {"id": 853,...
4	260000000	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	http://movies.disney.com/john-carter	49529	[{"id": 818, "name": "based on novel"}, {"id":...

5 rows × 23 columns

```
movies = movies[['movie_id', 'title', 'keywords', 'overview', 'genres', 'cast', 'crew']]
movies.head()
```

```
movies['genres'][0]
```

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

```
import ast

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

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

```
<ipython-input-16-082643038ae8>: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
  movies['genres'] = movies['genres'].apply(getgenres)
```

	movie_id	title	keywords	overview	genres	cast	
0	19995	Avatar	[{"id": 1463, "name": "culture clash"}, {"id":...	In the 22nd century, a paraplegic Marine is di...	[Action, Adventure, Fantasy, Science Fiction]	[{"cast_id": 242, "character": "Jake Sully", "...	"52fe
1	285	Pirates of the Caribbean: At World's End	[{"id": 270, "name": "ocean"}, {"id": 726, "na...	Captain Barbossa, long believed to be dead, ha...	[Adventure, Fantasy, Action]	[{"cast_id": 4, "character": "Captain Jack Spa...	"52fe
2	206647	Spectre	[{"id": 470, "name": "spy"}, {"id": 818, "name...	A cryptic message from Bond's past sends him o...	[Action, Adventure, Crime]	[{"cast_id": 1, "character": "James Bond", "cr...	"5480
3	49026	The Dark Knight Rises	[{"id": 849, "name": "dc comics"}, {"id": 853,...	Following the death of District Attorney Harve...	[Action, Crime, Drama, Thriller]	[{"cast_id": 2, "character": "Bruce Wayne / Ba...	"52fe
4	49529	John Carter	[{"id": 818, "name": "based on novel"}, {"id":...	John Carter is a war-weary, former military ca...	[Action, Adventure, Science Fiction]	[{"cast_id": 5, "character": "John Carter", "c...	"52fe
...	...	...	...	...	...	...	
1150	168672	American Hustle	[{"id": 10453, "name": "con artist"}, {"id": 1...	A con man, Irving Rosenfeld, along with	[Drama, Crime]	[{"cast_id": 3, "character": "Irving	"52fe

```
movies['genres'][0]
```

```
['Action', 'Adventure', 'Fantasy', 'Science Fiction']
```

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


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

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-22-2e44ebe5382d> in <cell line: 1>()
----> 1 movies['keywords'] = movies['keywords'].apply(getkeywords)
      2 movies

9 frames
/usr/lib/python3.10/ast.py in _raise_malformed_node(node)
     69         if lno := getattr(node, 'lineno', None):
     70             msg += f' on line {lno}'
---> 71         raise ValueError(msg + f': {node!r}')
     72     def _convert_num(node):
     73         if not isinstance(node, Constant) or type(node.value) not in
(int, float, complex):
```

```
movies['keywords'][0]
```

```
['culture clash',
 'future',
 'space war',
 'space colony',
 'society',
 'space travel',
 'futuristic',
 'romance',
 'space',
 'alien',
 'tribe',
 'alien planet',
 'cgi',
 'marine',
 'soldier',
 'battle',
 'love affair',
 'anti war',
 'power relations',
 'mind and soul',
 '3d']
```

```
def getcast(object):
    L = []
    counter = 0
    for i in ast.literal_eval(object):
        if counter == 3:
            break
        L.append(i['name'])
        counter += 1
    return L
```

+ Code + Text

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

```
movies['cast'][0]
```

```
['Sam Worthington', 'Zoe Saldana', 'Sigourney Weaver']
```

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

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

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

```
['James Cameron']
```

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

```
movies['overview'][0]
```

```
['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['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])
```

```
movies.head(1)
```

movie_id	title	keywords	overview	genres	cast
		[cultureclash, future,	[In, the, 22nd,	[Action, Adventure	[SamWorthington,

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

```
movies['tags'][0]
```

```
['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_movies = movies[['movie_id', 'title', 'tags']]
```

```
new_movies.head(1)
```

	movie_id	title	tags
0	19995	Avatar	[In, the, 22nd, century,, a, paraplegic, Marin...

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

```
C:\Users\Xen\AppData\Local\Temp\ipykernel_9936\2092922764.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
  new_movies['tags'] = new_movies['tags'].apply(lambda x : str(x).lower())
```

```
new_movies['tags'][0]
```

```
"['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']"
```

```
import nltk
```

```
from nltk.stem import PorterStemmer
ps = PorterStemmer()
```

```
def stem(text):
    L = []
    for i in text.split():
        L.append(ps.stem(i))
    return ' '.join(L)
```

```
new_movies['tags'][0]
```

```
"['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_movies['tags'].apply(stem)
```

```
0       ['in', 'the', '22nd', 'century,', 'a', 'parapl...
1       ['captain', 'barbossa,', 'long', 'believed', '...
2       ['a', 'cryptic', 'message', 'from', 'bond’s', ...
3       ['following', 'the', 'death', 'of', 'district'...
4       ['john', 'carter', 'is', 'a', 'war-weary,', 'f...
                              ...
4804    ['el', 'mariachi', 'just', 'wants', 'to', 'pla...
4805    ['a', 'newlywed', "couple's", 'honeymoon', 'is...
4806    ['"signed,', 'sealed,', 'delivered"', 'introdu...
4807    ['when', 'ambitious', 'new', 'york', 'attorney...
4808    ['ever', 'since', 'the', 'second', 'grade', 'w...
Name: tags, Length: 4809, dtype: object
```

```
new_movies['tags'][0]
```

```
"['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']"
```

```
from sklearn.feature_extraction.text import CountVectorizer

cv = CountVectorizer(max_features=4000, stop_words='english')
cv.fit_transform(new_movies['tags']).toarray()
```

```
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)
```

```
profiles = cv.fit_transform(new_movies['tags']).toarray()
```

```
profiles.shape
```

```
(4809, 4000)
```

```
from sklearn.metrics.pairwise import cosine_similarity

similarity = cosine_similarity(profiles)
```

```
similarity[0]
```

```
array([1.        , 0.09393364, 0.06262243, ..., 0.02615329, 0.0285831 ,
       0.        ])
```

```
def recommend(movie):
    index = np.where(new_movies['title'] == movie)[0][0]
    similar_movies = sorted(enumerate(similarity[index]), key=lambda x: x[1], reverse=True)[1:4]

    for i in similar_movies:
        print(new_movies['title'][i[0]])
```

```
recommend('Gladiator')
```

```
Hercules
Return of the Jedi
Abandoned
```

```
recommend('Aliens')
```

```
Alien
Alien³
Alien: Resurrection
```

0s completed at 9:17 PM