```python
import pandas as pd
import numpy as np
import ast


movies = pd.read_csv(r"/content/tmdb_5000_movies.csv.zip")
credits = pd.read_csv(r"/content/tmdb_5000_credits.csv.zip")


movies.head(2)
```

| | budget | genres | homepage | id | keywords | ori |
|---|---|---|---|---|---|---|
| **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... | |

🪄

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

| | movie_id | title | cast | crew |
|---|---|---|---|---|
| **0** | 19995 | Avatar | [{"cast_id": 242, "character": "Jake Sully", "... | [{"credit_id": "52fe48009251416c750aca23", "de... |

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

```
(4809, 23)
```

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

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

| | budget | genres | homepage | id | keywords | original_language |
|---|---|---|---|---|---|---|
| **0** | 237000000 | [{"id": 28, "name": "Action"}, {"id": 12, "nam | http://www.avatarmovie.com/ | 19995 | [{"id": 1463, "name": "culture clash"}, | en |

```
#genres
#id
#keywords
#title
#overview
#cast
#crew

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

movies.head(1)
```

| | movie_id | title | overview | genres | keywords | cast | c |
|---|---|---|---|---|---|---|---|
| **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_ "52fe48009251416c750aca "( |

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

```
    0
```

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

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

```
#action,adventure,fantasy,science fiction
```

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

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

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

| | movie_id | title | overview | genres | keywords | cast | cre |
|---|---|---|---|---|---|---|---|
| **0** | 19995 | Avatar | In the 22nd century, a paraplegic Marine is di | [Action] | [{"id": 1463, "name": "culture clash"}, {"id": | [{"cast_id": 242, "character": "Jake Sully", "... | [{"credit_id": "52fe48009251416c750aca23 "de |

```python
movies.iloc[0].keywords
```

```
'[{"id": 1463, "name": "culture clash"}, {"id": 2964, "name": "future"}, {"id": 338
6, "name": "space war"}, {"id": 3388, "name": "space colony"}, {"id": 3679, "name":
"society"}, {"id": 3801, "name": "space travel"}, {"id": 9685, "name": "futuristi
c"}, {"id": 9840, "name": "romance"}, {"id": 9882, "name": "space"}, {"id": 9951,
"name": "alien"}, {"id": 10148, "name": "tribe"}, {"id": 10158, "name": "alien plan
et"}, {"id": 10987, "name": "cgi"}, {"id": 11399, "name": "marine"}, {"id": 13065,
"name": "soldier"}, {"id": 14643, "name": "battle"}, {"id": 14720, "name": "love af
```

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

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

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

| | movie_id | title | overview | genres | keywords | cast | cre |
|---|---|---|---|---|---|---|---|
| **0** | 19995 | Avatar | In the 22nd century, a paraplegic Marine is di | [Action] | [culture clash] | [{"cast_id": 242, "character": "Jake Sully", "... | [{"credit_id": "52fe48009251416c750aca2 "de |

```python
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
```

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

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

| | movie_id | title | overview | genres | keywords | cast | cr |
|---|---|---|---|---|---|---|---|
| **0** | 19995 | Avatar | In the 22nd century, a paraplegic Marine is di | [Action] | [culture clash] | [Sam Worthington, Zoe Saldana, Sigourney Weaver] | [{"credit_ "52fe48009251416c750aca2 "d |

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

'[{"credit_id": "52fe48009251416c750aca23", "department": "Editing", "gender": 0, "id": 1721, "job": "Editor", "name": "Stephen E. Rivkin"}, {"credit_id": "539c47ecc 3a36810e3001f87", "department": "Art", "gender": 2, "id": 496, "job": "Production D esign", "name": "Rick Carter"}, {"credit_id": "54491c89c3a3680fb4001cf7", "departme nt": "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"}, {"cr edit_id": "539c4a4cc3a36810c9002101", "department": "Production", "gender": 1, "i

```
def fetch_director(obj):
    L = []
    for i in ast.literal_eval(obj):
        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] | [culture clash] | [Sam Worthington, Zoe Saldana, Sigourney Weaver] | [James Cameron] |
| | | | Captain | | | [Johnny Depp | |

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

```
'In the 22nd century, a paraplegic Marine is dispatched to the moon Pandora on a un
ique mission, but becomes torn between following orders and protecting an alien civ
ilization.'
```

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

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

| | movie_id | title | overview | genres | keywords | cast | crew |
|---|---|---|---|---|---|---|---|
| **0** | 19995 | Avatar | [In, the, 22nd, century,, a, paraplegic, Marin... | [Action] | [culture clash] | [Sam Worthington, Zoe Saldana, Sigourney Weaver] | [James Cameron] |

```
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()
```

| | movie_id | title | overview | genres | keywords | cast | |
|---|---|---|---|---|---|---|---|
| **0** | 19995 | Avatar | [In, the, 22nd, century,, a, paraplegic, Marin... | [Action] | [culture clash] | [SamWorthington, ZoeSaldana, SigourneyWeaver] | [JamesCam |
| **1** | 285 | Pirates of the Caribbean: At World's End | [Captain, Barbossa,, long, believed, to, be, d... | [Adventure] | [ocean] | [JohnnyDepp, OrlandoBloom, KeiraKnightley] | [GoreVerb |
| **2** | 206647 | Spectre | [A, cryptic, message, from, Bond's, past, | [Action] | [spy] | [DanielCraig, ChristophWaltz, LéaSeydoux] | [SamMe |

```
movies['tag'] = movies['overview'] + movies['genres'] + movies['keywords'] + movies['cast'
```

```
movies.head()
```

| | movie_id | title | overview | genres | keywords | cast | |
|---|---|---|---|---|---|---|---|
| **0** | 19995 | Avatar | [In, the, 22nd, century,, a, paraplegic, Marin... | [Action] | [culture clash] | [SamWorthington, ZoeSaldana, SigourneyWeaver] | [JamesC |
| **1** | 285 | Pirates of the Caribbean: At World's End | [Captain, Barbossa,, long, believed, to, be, d... | [Adventure] | [ocean] | [JohnnyDepp, OrlandoBloom, KeiraKnightley] | [GoreV |

```
new = movies.drop(columns=['overview','genres','keywords','cast','crew'])
new.head()
```

| | movie_id | title | tag |
|---|---|---|---|
| **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... |

```
new['tag'] = new['tag'].apply(lambda x: " ".join(x))
new.head()
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-88-413578ef6284> in <module>()
----> 1 new['tag'] = new['tag'].apply(lambda x: " ".join(x))
      2 new.head()
```

```
━━━━━━━━━━━━━━━━━━━━━━━ ⌃⌄ 1 frames ━━━━━━━━━━━━━━━━━━━━━━━
```

```
pandas/_libs/lib.pyx in pandas._libs.lib.map_infer()

<ipython-input-88-413578ef6284> in <lambda>(x)
----> 1 new['tag'] = new['tag'].apply(lambda x: " ".join(x))
      2 new.head()
```

```
TypeError: can only join an iterable
```

SEARCH STACK OVERFLOW

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-84-594643205845> in <module>()
----> 1 new_df['tag'].apply(lambda x: " ".join(x))

                           ↕ 1 frames

pandas/_libs/lib.pyx in pandas._libs.lib.map_infer()

<ipython-input-84-594643205845> in <lambda>(x)
```

```
new_df.head()
```

| | movie_id | title | tag |
|---|---|---|---|
| **0** | 19995 | Avatar | \<built-in method join of str object at 0x7fb52... |
| **1** | 285 | Pirates of the Caribbean: At World's End | \<built-in method join of str object at 0x7fb52... |
| **2** | 206647 | Spectre | \<built-in method join of str object at 0x7fb52... |
| | | The Dark Knight Ri | \<built-in method join of str object at |