

```
In [1]: import numpy as np
import pandas as pd
```

```
In [2]: movies = pd.read_csv(r"C:\Users\Nikhil\OneDrive\Desktop\MajorPro\tmdb_5000_
credits = pd.read_csv(r"C:\Users\Nikhil\OneDrive\Desktop\MajorPro\tmdb_5000
```

```
In [3]: movies.head(1)
```

Out[3]:

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

```
In [4]: credits.head(1)
```

Out[4]:

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

```
In [5]: credits.head(1)['cast'].values
```

```

Out[5]: array([{"cast_id": 242, "character": "Jake Sully", "credit_id": "5602a8a7c3a3685532001c9a", "gender": 2, "id": 65731, "name": "Sam Worthington", "order": 0}, {"cast_id": 3, "character": "Neytiri", "credit_id": "52fe48009251416c750ac9cb", "gender": 1, "id": 8691, "name": "Zoe Saldana", "order": 1}, {"cast_id": 25, "character": "Dr. Grace Augustine", "credit_id": "52fe48009251416c750aca39", "gender": 1, "id": 10205, "name": "Sigourney Weaver", "order": 2}, {"cast_id": 4, "character": "Col. Quaritch", "credit_id": "52fe48009251416c750ac9cf", "gender": 2, "id": 32747, "name": "Stephen Lang", "order": 3}, {"cast_id": 5, "character": "Trudy Chacon", "credit_id": "52fe48009251416c750ac9d3", "gender": 1, "id": 17647, "name": "Michelle Rodriguez", "order": 4}, {"cast_id": 8, "character": "Selfridge", "credit_id": "52fe48009251416c750ac9e1", "gender": 2, "id": 1771, "name": "Giovanni Ribisi", "order": 5}, {"cast_id": 7, "character": "Norm Spellman", "credit_id": "52fe48009251416c750ac9dd", "gender": 2, "id": 59231, "name": "Joel David Moore", "order": 6}, {"cast_id": 9, "character": "Moat", "credit_id": "52fe48009251416c750ac9e5", "gender": 1, "id": 30485, "name": "CCH Pounder", "order": 7}, {"cast_id": 11, "character": "Eytukan", "credit_id": "52fe48009251416c750ac9ed", "gender": 2, "id": 15853, "name": "Wes Studi", "order": 8}, {"cast_id": 10, "character": "Tsu'Tey", "credit_id": "52fe48009251416c750ac9e9", "gender": 2, "id": 10964, "name": "Laz Alonso", "order": 9}, {"cast_id": 12, "character": "Dr. Max Patel", "credit_id": "52fe48009251416c750ac9f1", "gender": 2, "id": 95697, "name": "Dileep Rao", "order": 10}, {"cast_id": 13, "character": "Lyle Wainfleet", "credit_id": "52fe48009251416c750ac9f5", "gender": 2, "id": 98215, "name": "Matt Gerald", "order": 11}, {"cast_id": 32, "character": "Private Fike", "credit_id": "52fe48009251416c750aca5b", "gender": 2, "id": 154153, "name": "Sean Anthony Moran", "order": 12}, {"cast_id": 33, "character": "Cryo Vault Med Tech", "credit_id": "52fe48009251416c750aca5f", "gender": 2, "id": 397312, "name": "Jason Whyte", "order": 13}, {"cast_id": 34, "character": "Venture Star Crew Chief", "credit_id": "52fe48009251416c750aca63", "gender": 2, "id": 42317, "name": "Scott Lawrence", "order": 14}, {"cast_id": 35, "character": "Lock Up Trooper", "credit_id": "52fe48009251416c750aca67", "gender": 2, "id": 986734, "name": "Kelly Kilgour", "order": 15}, {"cast_id": 36, "character": "Shuttle Pilot", "credit_id": "52fe48009251416c750aca6b", "gender": 0, "id": 1207227, "name": "James Patrick Pitt", "order": 16}, {"cast_id": 37, "character": "Shuttle Co-Pilot", "credit_id": "52fe48009251416c750aca6f", "gender": 0, "id": 1180936, "name": "Sean Patrick Murphy", "order": 17}, {"cast_id": 38, "character": "Shuttle Crew Chief", "credit_id": "52fe48009251416c750aca73", "gender": 2, "id": 1019578, "name": "Peter Dillon", "order": 18}, {"cast_id": 39, "character": "Tractor Operator / Troupe", "credit_id": "52fe48009251416c750aca77", "gender": 0, "id": 91443, "name": "Kevin Dorman", "order": 19}, {"cast_id": 40, "character": "Dragon Gunship Pilot", "credit_id": "52fe48009251416c750aca7b", "gender": 2, "id": 173391, "name": "Kelson Henderson", "order": 20}, {"cast_id": 41, "character": "Dragon Gunship Gunner", "credit_id": "52fe48009251416c750aca7f", "gender": 0, "id": 1207236, "name": "David Van Horn", "order": 21}, {"cast_id": 42, "character": "Dragon Gunship Navigator", "credit_id": "52fe48009251416c750aca83", "gender": 0, "id": 215913, "name": "Jacob Tomuri", "order": 22}, {"cast_id": 43, "character": "Suit #1", "credit_id": "52fe48009251416c750aca87", "gender": 0, "id": 143206, "name": "Michael Blain-Rozgay", "order": 23}, {"cast_id": 44, "character": "Suit #2", "credit_id": "52fe48009251416c750aca8b", "gender": 2, "id": 169676, "name": "Jon Curry", "order": 24}, {"cast_id": 46, "character": "Ambient Room Tech", "credit_id": "52fe48009251416c750aca8f", "gender": 0, "id": 1048610, "name": "Luke Hawker", "order": 25}, {"cast_id": 47, "character": "Ambient Room Tech / Troupe", "credit_id": "52fe48009251416c750aca93", "gender": 0, "id": 42288, "name": "Woody Schultz", "order": 26}, {"cast_id": 48, "character": "Horse Clan Leader", "credit_id": "52fe48009251416c750aca97", "gender": 2, "id": 68278, "name": "Peter Mensah", "order": 27}, {"cast_id": 49, "character": "Link Room Tech", "credit_id": "52fe48009251416c750aca9b", "gender": 0, "id": 1207247, "name": "Sonia Yee", "order": 28}, {"cast_id": 50, "character": "Link Room Tech", "credit_id": "52fe48009251416c750aca9f", "gender": 0, "id": 1207248, "name": "Sonia Yee", "order": 29}])

```

```

r": "Basketball Avatar / Troupe", "credit_id": "52fe48009251416c750aca9f",
"gender": 1, "id": 1207248, "name": "Jahnel Curfman", "order": 29}, {"cast_id": 51, "character": "Basketball Avatar", "credit_id": "52fe48009251416c750acaa3", "gender": 0, "id": 89714, "name": "Ilram Choi", "order": 30}, {"cast_id": 52, "character": "Na\\'vi Child", "credit_id": "52fe48009251416c750acaa7", "gender": 0, "id": 1207249, "name": "Kyla Warren", "order": 31}, {"cast_id": 53, "character": "Troupe", "credit_id": "52fe48009251416c750acaab", "gender": 0, "id": 1207250, "name": "Lisa Roumain", "order": 32}, {"cast_id": 54, "character": "Troupe", "credit_id": "52fe48009251416c750acaaf", "gender": 1, "id": 83105, "name": "Debra Wilson", "order": 33}, {"cast_id": 57, "character": "Troupe", "credit_id": "52fe48009251416c750acabb", "gender": 0, "id": 1207253, "name": "Chris Mala", "order": 34}, {"cast_id": 55, "character": "Troupe", "credit_id": "52fe48009251416c750acab3", "gender": 0, "id": 1207251, "name": "Taylor Kibby", "order": 35}, {"cast_id": 56, "character": "Troupe", "credit_id": "52fe48009251416c750acab7", "gender": 0, "id": 1207252, "name": "Jodie Landau", "order": 36}, {"cast_id": 58, "character": "Troupe", "credit_id": "52fe48009251416c750acabf", "gender": 0, "id": 1207254, "name": "Julie Lamm", "order": 37}, {"cast_id": 59, "character": "Troupe", "credit_id": "52fe48009251416c750acac3", "gender": 0, "id": 1207257, "name": "Cullen B. Madden", "order": 38}, {"cast_id": 60, "character": "Troupe", "credit_id": "52fe48009251416c750acac7", "gender": 0, "id": 1207259, "name": "Joseph Brady Madden", "order": 39}, {"cast_id": 61, "character": "Troupe", "credit_id": "52fe48009251416c750acacb", "gender": 0, "id": 1207262, "name": "Frankie Torres", "order": 40}, {"cast_id": 62, "character": "Troupe", "credit_id": "52fe48009251416c750acacf", "gender": 1, "id": 1158600, "name": "Austin Wilson", "order": 41}, {"cast_id": 63, "character": "Troupe", "credit_id": "52fe48019251416c750acad3", "gender": 1, "id": 983705, "name": "Sara Wilson", "order": 42}, {"cast_id": 64, "character": "Troupe", "credit_id": "52fe48019251416c750acad7", "gender": 0, "id": 1207263, "name": "Tamica Washington-Miller", "order": 43}, {"cast_id": 65, "character": "Op Center Staff", "credit_id": "52fe48019251416c750acadb", "gender": 1, "id": 1145098, "name": "Lucy Briant", "order": 44}, {"cast_id": 66, "character": "Op Center Staff", "credit_id": "52fe48019251416c750acadf", "gender": 2, "id": 33305, "name": "Nathan Meister", "order": 45}, {"cast_id": 67, "character": "Op Center Staff", "credit_id": "52fe48019251416c750acae3", "gender": 0, "id": 1207264, "name": "Gerry Blair", "order": 46}, {"cast_id": 68, "character": "Op Center Staff", "credit_id": "52fe48019251416c750acae7", "gender": 2, "id": 33311, "name": "Matthew Chamberlain", "order": 47}, {"cast_id": 69, "character": "Op Center Staff", "credit_id": "52fe48019251416c750acaeb", "gender": 0, "id": 1207265, "name": "Paul Yates", "order": 48}, {"cast_id": 70, "character": "Op Center Duty Officer", "credit_id": "52fe48019251416c750acaef", "gender": 0, "id": 1207266, "name": "Wray Wilson", "order": 49}, {"cast_id": 71, "character": "Op Center Staff", "credit_id": "52fe48019251416c750acaf3", "gender": 2, "id": 54492, "name": "James Gaylyn", "order": 50}, {"cast_id": 72, "character": "Dancer", "credit_id": "52fe48019251416c750acaf7", "gender": 0, "id": 1207267, "name": "Melvin Leno Clark III", "order": 51}, {"cast_id": 73, "character": "Dancer", "credit_id": "52fe48019251416c750acafb", "gender": 0, "id": 1207268, "name": "Carvon Futrell", "order": 52}, {"cast_id": 74, "character": "Dancer", "credit_id": "52fe48019251416c750acaff", "gender": 0, "id": 1207269, "name": "Brandon Jelkes", "order": 53}, {"cast_id": 75, "character": "Dancer", "credit_id": "52fe48019251416c750acb03", "gender": 0, "id": 1207270, "name": "Micah Moch", "order": 54}, {"cast_id": 76, "character": "Dancer", "credit_id": "52fe48019251416c750acb07", "gender": 0, "id": 1207271, "name": "Hanniyah Muhammad", "order": 55}, {"cast_id": 77, "character": "Dancer", "credit_id": "52fe48019251416c750acb0b", "gender": 0, "id": 1207272, "name": "Christopher Nolen", "order": 56}, {"cast_id": 78, "character": "Dancer", "credit_id": "52fe48019251416c750acb0f", "gender": 0, "id": 1207273, "name": "Christa Oliver", "order": 57}, {"cast_id": 79, "character": "Dancer", "credit_id": "52fe48019251416c750acb13", "gender": 0, "id": 1207274, "name": "April Marie Thomas", "order": 58}

```

```

r": 58}, {"cast_id": 80, "character": "Dancer", "credit_id": "52fe48019251416c750acb17", "gender": 0, "id": 1207275, "name": "Bravita A. Threatt", "order": 59}, {"cast_id": 81, "character": "Mining Chief (uncredited)", "credit_id": "52fe48019251416c750acb1b", "gender": 0, "id": 1207276, "name": "Colin Bleasdale", "order": 60}, {"cast_id": 82, "character": "Veteran Miner (uncredited)", "credit_id": "52fe48019251416c750acb1f", "gender": 0, "id": 107969, "name": "Mike Bodnar", "order": 61}, {"cast_id": 83, "character": "Richard (uncredited)", "credit_id": "52fe48019251416c750acb23", "gender": 0, "id": 1207278, "name": "Matt Clayton", "order": 62}, {"cast_id": 84, "character": "Nav\ 'i (uncredited)", "credit_id": "52fe48019251416c750acb27", "gender": 1, "id": 147898, "name": "Nicole Dionne", "order": 63}, {"cast_id": 85, "character": "Trooper (uncredited)", "credit_id": "52fe48019251416c750acb2b", "gender": 0, "id": 1207280, "name": "Jamie Harrison", "order": 64}, {"cast_id": 86, "character": "Trooper (uncredited)", "credit_id": "52fe48019251416c750acb2f", "gender": 0, "id": 1207281, "name": "Allan Henry", "order": 65}, {"cast_id": 87, "character": "Ground Technician (uncredited)", "credit_id": "52fe48019251416c750acb33", "gender": 2, "id": 1207282, "name": "Anthony Ingruber", "order": 66}, {"cast_id": 88, "character": "Flight Crew Mechanic (uncredited)", "credit_id": "52fe48019251416c750acb37", "gender": 0, "id": 1207283, "name": "Ashley Jeffery", "order": 67}, {"cast_id": 14, "character": "Samson Pilot", "credit_id": "52fe48009251416c750ac9f9", "gender": 0, "id": 98216, "name": "Dean Knowsley", "order": 68}, {"cast_id": 89, "character": "Trooper (uncredited)", "credit_id": "52fe48019251416c750acb3b", "gender": 0, "id": 1201399, "name": "Joseph Mika-Hunt", "order": 69}, {"cast_id": 90, "character": "Banshee (uncredited)", "credit_id": "52fe48019251416c750acb3f", "gender": 0, "id": 236696, "name": "Terry Notary", "order": 70}, {"cast_id": 91, "character": "Soldier (uncredited)", "credit_id": "52fe48019251416c750acb43", "gender": 0, "id": 1207287, "name": "Kai Pantano", "order": 71}, {"cast_id": 92, "character": "Blast Technician (uncredited)", "credit_id": "52fe48019251416c750acb47", "gender": 0, "id": 1207288, "name": "Logan Pithyou", "order": 72}, {"cast_id": 93, "character": "Vindum Raah (uncredited)", "credit_id": "52fe48019251416c750acb4b", "gender": 0, "id": 1207289, "name": "Stuart Pollock", "order": 73}, {"cast_id": 94, "character": "Hero (uncredited)", "credit_id": "52fe48019251416c750acb4f", "gender": 0, "id": 584868, "name": "Rajaa", "order": 74}, {"cast_id": 95, "character": "Ops Centreworke (uncredited)", "credit_id": "52fe48019251416c750acb53", "gender": 0, "id": 1207290, "name": "Gareth Ruck", "order": 75}, {"cast_id": 96, "character": "Engineer (uncredited)", "credit_id": "52fe48019251416c750acb57", "gender": 0, "id": 1062463, "name": "Rhian Sheehan", "order": 76}, {"cast_id": 97, "character": "Col. Quaritch\ 's Mech Suit (uncredited)", "credit_id": "52fe48019251416c750acb5b", "gender": 0, "id": 60656, "name": "T. J. Storm", "order": 77}, {"cast_id": 98, "character": "Female Marine (uncredited)", "credit_id": "52fe48019251416c750acb5f", "gender": 0, "id": 1207291, "name": "Jodie Taylor", "order": 78}, {"cast_id": 99, "character": "Ikran Clan Leader (uncredited)", "credit_id": "52fe48019251416c750acb63", "gender": 1, "id": 1186027, "name": "Alicia Vela-Bailey", "order": 79}, {"cast_id": 100, "character": "Geologist (uncredited)", "credit_id": "52fe48019251416c750acb67", "gender": 0, "id": 1207292, "name": "Richard Whiteside", "order": 80}, {"cast_id": 101, "character": "Na\ 'vi (uncredited)", "credit_id": "52fe48019251416c750acb6b", "gender": 0, "id": 103259, "name": "Nikie Zambo", "order": 81}, {"cast_id": 102, "character": "Ambient Room Tech / Troupe", "credit_id": "52fe48019251416c750acb6f", "gender": 1, "id": 42286, "name": "Julene Renee", "order": 82}]]],
dtype=object)

```

In [6]: `credits.head(1)['crew'].values`

Out[6]: array([{"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": "Casting", "name": "Mali Finn"}, {"credit_id": "5544ee3b925141499f0008fc", "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 Cameron"}, {"credit_id": "52fe48009251416c750aca17", "department": "Editing", "gender": 2, "id": 2710, "job": "Editor", "name": "James Cameron"}, {"credit_id": "52fe48009251416c750aca29", "department": "Production", "gender": 2, "id": 2710, "job": "Producer", "name": "James Cameron"}])

In [7]: `movies = movies.merge(credits,on='title')`

In [8]: `movies.head(1)`

Out[8]:

	budget	genres	homepage	id	keywords	original_language	original_title
0	237000000	[{"id": 28, "name": "Action"}, {"id": 12, "name": "Adventure"}]	http://www.avatarmovie.com/	19995	[{"id": 1463, "name": "culture clash"}, {"id": 1464, "name": "culture clash"}]	en	Avatar

1 rows × 23 columns

In [9]: `credits.shape`

Out[9]: (4803, 4)

In [10]: `# genres
id
keywords
title
overview
cast
crew

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

In [11]: `movies.info()`*# no of mmovies for each Language*

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 4809 entries, 0 to 4808
Data columns (total 7 columns):
 #   Column      Non-Null Count  Dtype
---  -
 0   movie_id    4809 non-null   int64
 1   title       4809 non-null   object
 2   overview    4806 non-null   object
 3   genres      4809 non-null   object
 4   keywords    4809 non-null   object
 5   cast        4809 non-null   object
 6   crew        4809 non-null   object
dtypes: int64(1), object(6)
memory usage: 300.6+ KB
```

In [12]: `movies.head()`

Out[12]:

	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", "...
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...
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...
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...
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...

```
In [13]: movies.isnull().sum()
```

```
Out[13]: movie_id      0  
         title        0  
         overview     3  
         genres       0  
         keywords     0  
         cast         0  
         crew         0  
         dtype: int64
```

```
In [14]: movies.dropna(inplace=True)
```

```
In [15]: movies.duplicated().sum()
```

```
Out[15]: 0
```

```
In [16]: movies.iloc[0].genres
```

```
Out[16]: '[{"id": 28, "name": "Action"}, {"id": 12, "name": "Adventure"}, {"id": 14, "name": "Fantasy"}, {"id": 878, "name": "Science Fiction"}]'
```

```
In [17]: # '[{"id": 28, "name": "Action"}, {"id": 12, "name": "Adventure"}, {"id": 14, "name": "Fantasy"}, {"id": 878, "name": "Science Fiction"}]'  
# ['Action', 'Adventure', 'Fantasy', 'SciFi']
```

```
In [18]: def convert(obj):  
         L = []  
         for i in ast.literal_eval(obj):  
             L.append(i['name'])  
         return L
```



```
In [57]: movies['genres'] = movies['genres'].apply(convert) # this helps make the vi
```

```

-----
-
ValueError                                Traceback (most recent call last)
Cell In[57], line 1
----> 1 movies['genres'] = movies['genres'].apply(convert)

File ~\anaconda3\lib\site-packages\pandas\core\series.py:4771, in Series.apply(self, func, convert_dtype, args, **kwargs)
    4661 def apply(
    4662     self,
    4663     func: AggFuncType,
    4664     (...)
    4665     **kwargs,
    4666 ) -> DataFrame | Series:
    4667     """
    4668     Invoke function on values of Series.
    4669     (...)
    4670     dtype: float64
    4671     """
-> 4771     return SeriesApply(self, func, convert_dtype, args, kwargs).apply()

File ~\anaconda3\lib\site-packages\pandas\core\apply.py:1123, in SeriesApply.apply(self)
    1120     return self.apply_str()
    1122 # self.f is Callable
-> 1123 return self.apply_standard()

File ~\anaconda3\lib\site-packages\pandas\core\apply.py:1174, in SeriesApply.apply_standard(self)
    1172     else:
    1173         values = obj.astype(object)._values
-> 1174         mapped = lib.map_infer(
    1175             values,
    1176             f,
    1177             convert=self.convert_dtype,
    1178         )
    1180 if len(mapped) and isinstance(mapped[0], ABCSeries):
    1181     # GH#43986 Need to do list(mapped) in order to get treated as
nested
    1182     # See also GH#25959 regarding EA support
    1183     return obj._constructor_expanddim(list(mapped), index=obj.index)

File ~\anaconda3\lib\site-packages\pandas\_libs\lib.pyx:2924, in pandas._libs.lib.map_infer()

Cell In[18], line 3, in convert(obj)
    1 def convert(obj):
    2     L = []
----> 3     for i in ast.literal_eval(obj):
    4         L.append(i['name'])
    5     return L

File ~\anaconda3\lib\ast.py:110, in literal_eval(node_or_string)
    108     return left - right
    109     return _convert_signed_num(node)
--> 110 return _convert(node_or_string)

```

```

File ~\anaconda3\lib\ast.py:109, in literal_eval.<locals>._convert(node)
    107         else:
    108             return left - right
--> 109 return _convert_signed_num(node)

File ~\anaconda3\lib\ast.py:83, in literal_eval.<locals>._convert_signed_num(node)
    81     else:
    82         return - operand
---> 83 return _convert_num(node)

File ~\anaconda3\lib\ast.py:74, in literal_eval.<locals>._convert_num(node)
    72 def _convert_num(node):
    73     if not isinstance(node, Constant) or type(node.value) not in
(int, float, complex):
---> 74         _raise_malformed_node(node)
    75     return node.value

File ~\anaconda3\lib\ast.py:71, in literal_eval.<locals>._raise_malformed_node(node)
    69 if lno := getattr(node, 'lineno', None):
    70     msg += f' on line {lno}'
---> 71 raise ValueError(msg + f': {node!r}')

ValueError: malformed node or string: ['[', '{', '"', 'i', 'd', '"', ':',
',', '2', '8', ',', ',', '"', 'n', 'a', 'm', 'e', '"', ':', '"', 'A',
',', 'c', 't', 'i', 'o', 'n', '"', '}', ',', '{', '"', 'i', 'd', '"', ':',
',', '1', '2', ',', ',', '"', 'n', 'a', 'm', 'e', '"', ':', '"', 'A',
',', 'd', 'v', 'e', 'n', 't', 'u', 'r', 'e', '"', '}', ',', '{', '"', 'i',
',', 'd', '"', ':', '1', '4', ',', ',', '"', 'n', 'a', 'm', 'e', '"', ':',
',', '"', 'F', 'a', 'n', 't', 'a', 's', 'y', '"', '}', ',', '{', '"',
',', 'd', '"', ':', '8', '7', '8', ',', ',', '"', 'n', 'a', 'm', 'e',
',', '"', ':', '"', 'S', 'c', 'i', 'e', 'n', 'c', 'e', '"', 'F', 'i', 'c',
',', 't', 'i', 'o', 'n', '"', '}', '']']

```

```

In [21]: import ast
ast.literal_eval(['{"id": 28, "name": "Action"}, {"id": 12, "name": "Advent
# we are converting string list to normal list as string list cant be data

```

```

Out[21]: [{'id': 28, 'name': 'Action'},
{'id': 12, 'name': 'Adventure'},
{'id': 14, 'name': 'Fantasy'},
{'id': 878, 'name': 'Science Fiction'}]

```

```
In [22]: movies.head()
```

Out[22]:

	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", "...
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...
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...
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...
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...

```
In [23]: movies['keywords'] = movies['keywords'].apply(convert)
```

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

Out[24]:

	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...	[culture clash, future, space war, space colon...	[{"cast_id": 242, "character": "Jake Sully", "...
1	285	Pirates of the Caribbean: At World's End	Captain Barbossa, long believed to be dead, ha...	[{"id": 12, "name": "Adventure"}, {"id": 14, "...	[ocean, drug abuse, exotic island, east india ...	[{"cast_id": 4, "character": "Captain Jack Spa...
2	206647	Spectre	A cryptic message from Bond's past sends him o...	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	[spy, based on novel, secret agent, sequel, mi...	[{"cast_id": 1, "character": "James Bond", "cr...
3	49026	The Dark Knight Rises	Following the death of District Attorney Harve...	[{"id": 28, "name": "Action"}, {"id": 80, "nam...	[dc comics, crime fighter, terrorist, secret i...	[{"cast_id": 2, "character": "Bruce Wayne / Ba...
4	49529	John Carter	John Carter is a war-weary, former military ca...	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	[based on novel, mars, medallion, space travel...	[{"cast_id": 5, "character": "John Carter", "c...

In [25]: `# movies['cast'][0]`

```
In [26]: 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 [27]: `movies['cast'] = movies['cast'].apply(convert)`

```
In [28]: movies.head()
```

Out[28]:

	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...	[culture clash, future, space war, space colon...	[Sam Worthington, Zoe Saldana, Sigourney Weave...
1	285	Pirates of the Caribbean: At World's End	Captain Barbossa, long believed to be dead, ha...	[{"id": 12, "name": "Adventure"}, {"id": 14, "...	[ocean, drug abuse, exotic island, east india ...	[Johnny Depp, Orlando Bloom, Keira Knightley, ...
2	206647	Spectre	A cryptic message from Bond's past sends him o...	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	[spy, based on novel, secret agent, sequel, mi...	[Daniel Craig, Christoph Waltz, Léa Seydoux, R...
3	49026	The Dark Knight Rises	Following the death of District Attorney Harve...	[{"id": 28, "name": "Action"}, {"id": 80, "nam...	[dc comics, crime fighter, terrorist, secret i...	[Christian Bale, Michael Caine, Gary Oldman, A...
4	49529	John Carter	John Carter is a war-weary, former military ca...	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	[based on novel, mars, medallion, space travel...	[Taylor Kitsch, Lynn Collins, Samantha Morton,...

```
In [29]: movies['crew'][0]
```

Out[29]:

```
'[{"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": "Casting", "name": "Mali Finn"}, {"credit_id": "5544ee3b925141499f0008fc", "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 Cameron"}, {"credit_id": "52fe48009251416c750aca17", "department": "Editing", "gender": 2, "id": 2710, "job": "Editor", "name": "James Cameron"}, {"credit_id": "52fe48009251416c750aca29", "department": "Production", "gender": 2, "id": 2710, "job": "Producer", "name": "James Cameron"}]
```

```
In [30]: def fetch_director(obj):
        L = []
        for i in ast.literal_eval(obj):
            if i['job'] == 'Director':
                L.append(i['name'])
            break
        return L
```

```
In [31]: movies['crew'] = movies['crew'].apply(fetch_director)
```

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

Out[32]:

	movie_id	title	overview	genres	keywords	cast	crew
0	19995	Avatar	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.	[{"id": 28, "name": "Action"}, {"id": 12, "name": "Adventure"}]	[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, has returned to his crew.	[{"id": 12, "name": "Adventure"}, {"id": 14, "name": "Fantasy"}]	[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 on a quest to find a mysterious villain.	[{"id": 28, "name": "Action"}, {"id": 12, "name": "Adventure"}]	[spy, based on novel, secret agent, sequel, mi...]	[Daniel Craig, Christoph Waltz, Léa Seydoux, R...]	[Sam Mendes]
3	49026	The Dark Knight Rises	Following the death of District Attorney Harvey Dent, Batman deduces a link between his enemy and a woman on the run.	[{"id": 28, "name": "Action"}, {"id": 80, "name": "Crime"}]	[dc comics, crime fighter, terrorist, secret i...]	[Christian Bale, Michael Caine, Gary Oldman, A...]	[Christopher Nolan]
4	49529	John Carter	John Carter is a war-weary, former military captain who has returned home to a peaceful life.	[{"id": 28, "name": "Action"}, {"id": 12, "name": "Adventure"}]	[based on novel, mars, medallion, space travel...]	[Taylor Kitsch, Lynn Collins, Samantha Morton, ...]	[Andrew Stanton]

```
In [33]: movies['overview'][0]
```

Out[33]: '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.'

```
In [34]: movies['overview'] = movies['overview'].apply(lambda x:x.split())
```

In [35]: `movies.head()`

Out[35]:

	movie_id	title	overview	genres	keywords	cast	crew
0	19995	Avatar	[In, the, 22nd, century,, a, paraplegic, Marin...	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	[culture clash, future, space war, space colon...	[Sam Worthington, Zoe Saldana, Sigourney Weave...	[James Cameron]
1	285	Pirates of the Caribbean: At World's End	[Captain, Barbossa,, long, believed, to, be, d...	[{"id": 12, "name": "Adventure"}, {"id": 14, "...	[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...	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	[spy, based on novel, secret agent, sequel, mi...	[Daniel Craig, Christoph Waltz, Léa Seydoux, R...	[Sam Mendes]
3	49026	The Dark Knight Rises	[Following, the, death, of, District, Attorney...	[{"id": 28, "name": "Action"}, {"id": 80, "nam...	[dc comics, crime fighter, terrorist, secret i...	[Christian Bale, Michael Caine, Gary Oldman, A...	[Christopher Nolan]
4	49529	John Carter	[John, Carter, is, a, war-weary,, former, mili...	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	[based on novel, mars, medallion, space travel...	[Taylor Kitsch, Lynn Collins, Samantha Morton,...	[Andrew Stanton]

In [36]: `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 [37]: movies.head()
```

Out[37]:

	movie_id	title	overview	genres	keywords	cast	cre
0	19995	Avatar	[In, the, 22nd, century,, a, paraplegic, Marin...	[[{"id": 1, "name": "Action"}, {"id": 2, "name": "Adventure"}, {"id": 8, "name": "Science Fiction"}, {"id": 14, "name": "Fantasy"}], [{"text": "culture clash", "score": 1}, {"text": "future", "score": 1}, {"text": "space war", "score": 1}, {"text": "space colony", "score": 1}]]	[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...	[[{"id": 1, "name": "Action"}, {"id": 2, "name": "Adventure"}, {"id": 8, "name": "Fantasy"}], [{"text": "ocean", "score": 1}, {"text": "drug abuse", "score": 1}, {"text": "exotic island", "score": 1}, {"text": "east indian trad...", "score": 1}]]	[Johnny Depp, Orlando Bloom, Keira Knightley, Ste...	[Gore Verbinski]	
2	206647	Spectre	[A, cryptic, message, from, Bond's, past, send...	[[{"id": 1, "name": "Action"}, {"id": 2, "name": "Adventure"}, {"id": 8, "name": "Thriller"}, {"id": 14, "name": "Mystery"}], [{"text": "spy", "score": 1}, {"text": "based on novel", "score": 1}, {"text": "secret agent", "score": 1}, {"text": "sequel, mi6, ...", "score": 1}]]	[Daniel Craig, Christoph Waltz, Léa Seydoux, Ralph...	[Sam Mendes]	
3	49026	The Dark Knight Rises	[Following, the, death, of, District, Attorney...	[[{"id": 1, "name": "Action"}, {"id": 2, "name": "Adventure"}, {"id": 8, "name": "Thriller"}, {"id": 14, "name": "Mystery"}], [{"text": "dc comics", "score": 1}, {"text": "crime fighter", "score": 1}, {"text": "terrorist", "score": 1}, {"text": "secret iden...", "score": 1}]]	[Christian Bale, Michael Caine, Gary Oldman, Anne...	[Christopher Nolan]	
4	49529	John Carter	[John, Carter, is, a, war-weary,, former, mili...	[[{"id": 1, "name": "Action"}, {"id": 2, "name": "Adventure"}, {"id": 8, "name": "Fantasy"}, {"id": 14, "name": "Mystery"}], [{"text": "based on novel", "score": 1}, {"text": "mars", "score": 1}, {"text": "medallion", "score": 1}, {"text": "space travel", "score": 1}, {"text": "p...", "score": 1}]]	[Taylor Kitsch, Lynn Collins, Samantha Morton, Wi...	[Andrew Stanton]	

```
In [38]: movies['tags'] = movies['overview'] + movies['genres'] + movies['keywords']
```

```
In [39]: movies.head()
```

```
Out[39]:
```

	movie_id	title	overview	genres	keywords	cast	crea
0	19995	Avatar	[In, the, 22nd, century,, a, paraplegic, Marin...	[[, {, ", i, d, ", :, , 2, 8, ,, , ", n, a, m,...	[cultureclash, future, spacewar, spacecolony, ...	[SamWorthington, ZoeSaldana, SigourneyWeaver, ...	[JamesCameror
1	285	Pirates of the Caribbean: At World's End	[Captain, Barbossa,, long, believed, to, be, d...	[[, {, ", i, d, ", :, , 1, 2, ,, , ", n, a, m,...	[ocean, drugabuse, exoticisland, eastindiatrad...	[JohnnyDepp, OrlandoBloom, KeiraKnightley, Ste...	[GoreVerbinsk
2	206647	Spectre	[A, cryptic, message, from, Bond's, past, send...	[[, {, ", i, d, ", :, , 2, 8, ,, , ", n, a, m,...	[spy, basedonnovel, secretagent, sequel, mi6, ...	[DanielCraig, ChristophWaltz, LéaSeydoux, Ralp...	[SamMende:
3	49026	The Dark Knight Rises	[Following, the, death, of, District, Attorney...	[[, {, ", i, d, ", :, , 2, 8, ,, , ", n, a, m,...	[dccomics, crimefighter, terrorist, secretiden...	[ChristianBale, MichaelCaine, GaryOldman, Anne...	[ChristopherNolar
4	49529	John Carter	[John, Carter, is, a, war-weary,, former, mili...	[[, {, ", i, d, ", :, , 2, 8, ,, , ", n, a, m,...	[basedonnovel, mars, medallion, spacetravel, p...	[TaylorKitsch, LynnCollins, SamanthaMorton, Wi...	[AndrewStantor



```
In [40]: new_df = movies[['movie_id', 'title', 'tags']]
```

```
In [41]: new_df['tags'] = new_df['tags'].apply(lambda x: " ".join(x))
```

C:\Users\Nikhil\AppData\Local\Temp\ipykernel_12896\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-versus-a-copy (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))
```

```
In [42]: new_df.head()
```

```
Out[42]:
```

	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 [43]: new_df['tags'][0]
```

```
Out[43]: '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. [ { "id": 28, "name": "Action" }, { "id": 12, "name": "Adventure" }, { "id": 14, "name": "Fantasy" }, { "id": 878, "name": "Science Fiction" } ] cultureclash future spacewar spacecolony society spacetravel futuristic romance space alien tribe alienplanet cgi marine soldier battle loveaffair antiwar powerrelations mindandsoul 3d SamWorthington ZoeSaldana SigourneyWeaver StephenLang MichelleRodriguez GiovanniRibisi JoelDavidMoore CCHPounder WesStudi LazAlonso DileepRao MattGerald SeanAnthonyMoran JasonWhyte ScottLawrence KellyKilgour JamesPatrickPitt SeanPatrickMurphy PeterDillon KevinDorman KelsonHenderson DavidVanHorn JacobTomuri MichaelBlain-Rozgay JonCurry LukeHawker WoodySchultz PeterMensah SoniaYee JahnelCurfman IlramChoi KylaWarren LisaRoumain DebraWilson ChrisMala TaylorKibby JodieLandau JulieLamm CullenB.Madden JosephBradyMadden FrankieTorres AustinWilson SaraWilson TamicaWashington-Miller LucyBriant NathanMeister GerryBlair MatthewChamberlain PaulYates WrayWilson JamesGaylyn MelvinLenoClarkIII CarvonFutrell BrandonJelkes MicahMoch HanniyahMuhammad ChristopherNolen ChristaOliver AprilMarieThomas BravitaA.Threatt ColinBleasdale MikeBodnar MattClayton NicoleDionne JamieHarrison AllanHenry AnthonyIngruber AshleyJeffery DeanKnowsley JosephMika-Hunt TerryNotary KaiPantano LoganPithyou StuartPollock Raja GarethRuck RhianSheehan T.J.Storm JodieTaylor AliciaVela-Bailey RichardWhiteside NikieZambo JulieRenee JamesCameron'
```

```
In [44]: new_df['tags'] = new_df['tags'].apply(lambda x:x.lower())
```

C:\Users\Nikhil\AppData\Local\Temp\ipykernel_12896\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 (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 [45]: `new_df.head()`

Out[45]:

	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 [46]: `import nltk`

In [47]: `!pip install nltk`

Requirement already satisfied: nltk in c:\users\nikhil\anaconda3\lib\site-packages (3.7)
 Requirement already satisfied: regex>=2021.8.3 in c:\users\nikhil\anaconda3\lib\site-packages (from nltk) (2022.7.9)
 Requirement already satisfied: click in c:\users\nikhil\anaconda3\lib\site-packages (from nltk) (8.0.4)
 Requirement already satisfied: joblib in c:\users\nikhil\anaconda3\lib\site-packages (from nltk) (1.1.1)
 Requirement already satisfied: tqdm in c:\users\nikhil\anaconda3\lib\site-packages (from nltk) (4.64.1)
 Requirement already satisfied: colorama in c:\users\nikhil\anaconda3\lib\site-packages (from click->nltk) (0.4.6)

In [48]: `from nltk.stem.porter import PorterStemmer`
`ps = PorterStemmer()`

In [49]: `def stem(text):`
 `y = []`
 `for i in text.split():`
 `y.append(ps.stem(i))`
 `return " ".join(y)`

In [50]: `new_df['tags'] = new_df['tags'].apply(stem)`

C:\Users\Nikhil\AppData\Local\Temp\ipykernel_12896\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 (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 [51]: #text vectorization
```

```
In [52]: new_df['tags'][0]
```

```
Out[52]: '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. [ { " i d " : 2 8 , " n a m e " : " a c t i o n " } , { " i
d " : 1 2 , " n a m e " : " a d v e n t u r e " } , { " i d " : 1 4 , " n
a m e " : " f a n t a s y " } , { " i d " : 8 7 8 , " n a m e " : " s c i
e n c e f i c t i o n " } ] cultureclash futur spacewar spacecoloni societ
i spacetravel futurist romanc space alien tribe alienplanet cgi marin sold
ier battl loveaffair antiwar powerrel mindandsoul 3d samworthington zoesal
dana sigourneyweav stephenlang michellerodriguez giovanniribisi joeldavidm
oor cchpounder wesstudi lazalonso dileeprao mattgerald seananthonymoran ja
sonwhyt scottlawr kellykilgour jamespatrickpitt seanpatrickmurphi peterdil
lon kevindorman kelsonhenderson davidvanhorn jacobtomuri michaelblain-rozg
ay joncurri lukehawk woodyschultz petermensah soniaye jahnelcurfman ilramc
hoi kylawarren lisaroumain debrawilson chrismala taylorkibbi jodielandau j
ulielamm cullenb.madden josephbradymadden frankietorr austinwilson sarawil
son tamicawashington-mil lucybri nathanmeist gerryblair matthewchamberlain
paulyat wraywilson jamesgaylyn melvinlenoclarkiii carvonfutrel brandonjclk
micahmoch hanniyahmuhammad christophernolen christaoliv aprilmarbethoma br
avita.threatt colinbleasdal mikebodnar mattclayton nicoledionn jamieharri
son allanhenri anthonyingrub ashleyjefferi deanknowsley josephmika-hunt te
rryotari kaipantano loganpithy stuartpollock raja garethruck rhianssheehan
t.j.storm jodietaylor aliciavela-bailey richardwhitesid nikiezambo julener
ene jamescameron'
```

```
In [53]: new_df['tags'][1]
```

```
Out[53]: 'captain barbossa, long believ to be dead, ha come back to life and is hea
d to the edg of the earth with will turner and elizabeth swann. but noth i
s quit as it seems. [ { " i d " : 1 2 , " n a m e " : " a d v e n t u r e
" } , { " i d " : 1 4 , " n a m e " : " f a n t a s y " } , { " i d " : 2
8 , " n a m e " : " a c t i o n " } ] ocean drugabus exoticisland eastindi
atradingcompani loveofone\slif traitor shipwreck strongwoman ship allianc
calypso afterlif fighter pirat swashbuckl aftercreditssting johnnydepp orl
andobloom keiraknightley stellanskarsgård chowyun-fat billnighi geoffreyru
sh jackdavenport kevinmcn tomholland naomieharri jonathanpryc keithrichard
leeareenberg mackenziecrook gregelli davidbaili martinklebba davidschofield
laurenmah vanessabbranch angusbarnett gilesnew reggiele dominicscottkay tak
ayofisch davidmeuni ho-kwants andybeckwith peterdonaldbadalamentiii christ
ophers.capp keithrichard hakeemkae-kazim ghassanmassoud goreverbinski'
```

```
In [54]: from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer(max_features = 5000,stop_words = 'english')
```

```
In [55]: vectors = cv.fit_transform(new_df['tags']).toarray()
```

```
In [56]: vectors
```

```
Out[56]: 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 [ ]: vectors[0]
```

```
In [ ]: len(cv.get_feature_names_out())
```

```
In [ ]: cv.get_feature_names_out()
```

```
In [ ]: ps.stem('loving') # the stem function chnages loving , loved etc to love
```

```
In [ ]: stem('in the 22nd century, a paraplegic marine is dispatched to the moon pa
```

```
In [ ]: from sklearn.metrics.pairwise import cosine_similarity
```

```
In [ ]: cosine_similarity(vectors)
```

```
In [ ]: similarity = cosine_similarity(vectors)
```

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

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

```
In [ ]: new_df[new_df['title'] == 'Batman Begins'].index[0] #shows index position o
```

```
In [ ]: recommend('Batman Begins')
```

```
In [ ]: new_df.iloc[1215].title # shows the movie
```

