

movie_recommendation

May 12, 2022

```
[10]: import pandas as pd
import numpy as np
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.metrics.pairwise import cosine_similarity

df = pd.read_csv("movie_dataset.csv")
```

```
[12]: # we choose keywords ,cast,genres and director as our fields to recommend
      ↪ movies.
features = ['keywords','cast','genres','director']
def combine_features(row):
    return row['keywords']+" "+row['cast']+" "+row['genres']+" "+row['director']
```

```
[13]: for feature in features:
        df[feature] = df[feature].fillna('') #filling all NaNs with blank string

df["combined_features"] = df.apply(combine_features,axis=1)
#applying combined_features() method over each rows of dataframe and storing
      ↪ the combined
#string in "combined_features" column
```

```
[26]: df.describe()
```

```
[26]:
```

	index	budget	id	popularity	revenue \
count	4803.000000	4.803000e+03	4803.000000	4803.000000	4.803000e+03
mean	2401.000000	2.904504e+07	57165.484281	21.492301	8.226064e+07
std	1386.651002	4.072239e+07	88694.614033	31.816650	1.628571e+08
min	0.000000	0.000000e+00	5.000000	0.000000	0.000000e+00
25%	1200.500000	7.900000e+05	9014.500000	4.668070	0.000000e+00
50%	2401.000000	1.500000e+07	14629.000000	12.921594	1.917000e+07
75%	3601.500000	4.000000e+07	58610.500000	28.313505	9.291719e+07
max	4802.000000	3.800000e+08	459488.000000	875.581305	2.787965e+09

	runtime	vote_average	vote_count
count	4801.000000	4803.000000	4803.000000
mean	106.875859	6.092172	690.217989
std	22.611935	1.194612	1234.585891

min	0.000000	0.000000	0.000000
25%	94.000000	5.600000	54.000000
50%	103.000000	6.200000	235.000000
75%	118.000000	6.800000	737.000000
max	338.000000	10.000000	13752.000000

```
[27]: df.head()
```

```
[27]:
```

	index	budget	genres	\
0	0	237000000	Action Adventure Fantasy Science Fiction	
1	1	300000000	Adventure Fantasy Action	
2	2	245000000	Action Adventure Crime	
3	3	250000000	Action Crime Drama Thriller	
4	4	260000000	Action Adventure Science Fiction	

	homepage	id	\
0	http://www.avatarmovie.com/	19995	
1	http://disney.go.com/disneypictures/pirates/	285	
2	http://www.sonypictures.com/movies/spectre/	206647	
3	http://www.thedarkknightrisers.com/	49026	
4	http://movies.disney.com/john-carter	49529	

	keywords	original_language	\
0	culture clash future space war space colony so...	en	
1	ocean drug abuse exotic island east india trad...	en	
2	spy based on novel secret agent sequel mi6	en	
3	dc comics crime fighter terrorist secret ident...	en	
4	based on novel mars medallion space travel pri...	en	

	original_title	\
0	Avatar	
1	Pirates of the Caribbean: At World's End	
2	Spectre	
3	The Dark Knight Rises	
4	John Carter	

	overview	popularity	...	\
0	In the 22nd century, a paraplegic Marine is di...	150.437577	...	
1	Captain Barbossa, long believed to be dead, ha...	139.082615	...	
2	A cryptic message from Bond's past sends him o...	107.376788	...	
3	Following the death of District Attorney Harve...	112.312950	...	
4	John Carter is a war-weary, former military ca...	43.926995	...	

	spoken_languages	status	\
0	[{"iso_639_1": "en", "name": "English"}, {"iso...	Released	
1	[{"iso_639_1": "en", "name": "English"}]	Released	
2	[{"iso_639_1": "fr", "name": "Fran\u00e7ais"},...	Released	

```

3      [{"iso_639_1": "en", "name": "English"}] Released
4      [{"iso_639_1": "en", "name": "English"}] Released

```

```

                                tagline \
0      Enter the World of Pandora.
1  At the end of the world, the adventure begins.
2      A Plan No One Escapes
3      The Legend Ends
4      Lost in our world, found in another.

```

```

                                title  vote_average  vote_count \
0      Avatar                                7.2          11800
1  Pirates of the Caribbean: At World's End          6.9          4500
2      Spectre                                6.3          4466
3      The Dark Knight Rises                    7.6          9106
4      John Carter                                6.1          2124

```

```

                                cast \
0  Sam Worthington Zoe Saldana Sigourney Weaver S...
1  Johnny Depp Orlando Bloom Keira Knightley Stel...
2  Daniel Craig Christoph Waltz L\u00e9a Seydoux ...
3  Christian Bale Michael Caine Gary Oldman Anne ...
4  Taylor Kitsch Lynn Collins Samantha Morton Wil...

```

```

                                crew          director \
0  [{'name': 'Stephen E. Rivkin', 'gender': 0, 'd...   James Cameron
1  [{'name': 'Dariusz Wolski', 'gender': 2, 'depa...   Gore Verbinski
2  [{'name': 'Thomas Newman', 'gender': 2, 'depar...   Sam Mendes
3  [{'name': 'Hans Zimmer', 'gender': 2, 'departm...   Christopher Nolan
4  [{'name': 'Andrew Stanton', 'gender': 2, 'depa...   Andrew Stanton

```

```

                                combined_features
0  culture clash future space war space colony so...
1  ocean drug abuse exotic island east india trad...
2  spy based on novel secret agent sequel mi6 Dan...
3  dc comics crime fighter terrorist secret ident...
4  based on novel mars medallion space travel pri...

```

[5 rows x 25 columns]

```
[14]: df.iloc[0].combined_features
```

```
[14]: 'culture clash future space war space colony society Sam Worthington Zoe Saldana
Sigourney Weaver Stephen Lang Michelle Rodriguez Action Adventure Fantasy
Science Fiction James Cameron'
```

```

[15]: cv = CountVectorizer() #creating new CountVectorizer() object
      count_matrix = cv.fit_transform(df["combined_features"])
      #feeding combined strings(movie contents) to CountVectorizer() object

[16]: cosine_sim = cosine_similarity(count_matrix)

[17]: def get_title_from_index(index):
      return df[df.index == index]["title"].values[0]
      def get_index_from_title(title):
      return df[df.title == title]["index"].values[0]

[23]: movie_user_likes = "John Carter"
      movie_index = get_index_from_title(movie_user_likes)
      similar_movies = list(enumerate(cosine_sim[movie_index]))
      #accessing the row corresponding to given movie to find all the similarity_
      ↪scores for that movie
      #and then enumerating over it

[24]: sorted_similar_movies = sorted(similar_movies, key=lambda x:x[1], reverse=True)[1:
      ↪]

[25]: i=0
      print("Top 5 similar movies to "+movie_user_likes+" are:\n")
      for element in sorted_similar_movies:
          print(get_title_from_index(element[0]))
          i=i+1
          if i>5:
              break

```

Top 5 similar movies to John Carter are:

The Host
 Heaven is for Real
 Beastmaster 2: Through the Portal of Time
 The Helix... Loaded
 Transformers
 The Martian