movie-recommendation

July 27, 2023

0.1 Movie Recommendation System

[1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.metrics.pairwise import linear_kernel
from sklearn.metrics.pairwise import cosine_similarity
from ast import literal_eval
```

[2]:
```
path = "./Desktop/TechVidvan/movie_recommendation"
credits_df = pd.read_csv(path + "/tmdb_credits.csv")
movies_df = pd.read_csv(path + "/tmdb_movies.csv")
```

[3]:
```
movies_df.head()
```

[3]:
```
      budget                                             genres  \
0  237000000  [{"id": 28, "name": "Action"}, {"id": 12, "nam…
1  300000000  [{"id": 12, "name": "Adventure"}, {"id": 14, "…
2  245000000  [{"id": 28, "name": "Action"}, {"id": 12, "nam…
3  250000000  [{"id": 28, "name": "Action"}, {"id": 80, "nam…
4  260000000  [{"id": 28, "name": "Action"}, {"id": 12, "nam…

                                       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.thedarkknightrises.com/   49026
4           http://movies.disney.com/john-carter   49529

                                            keywords original_language  \
0  [{"id": 1463, "name": "culture clash"}, {"id":…                en
1  [{"id": 270, "name": "ocean"}, {"id": 726, "na…                en
2  [{"id": 470, "name": "spy"}, {"id": 818, "name…                en
3  [{"id": 849, "name": "dc comics"}, {"id": 853,…                en
4  [{"id": 818, "name": "based on novel"}, {"id":…                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

                                production_companies  \
0  [{"name": "Ingenious Film Partners", "id": 289…
1  [{"name": "Walt Disney Pictures", "id": 2}, {"…
2  [{"name": "Columbia Pictures", "id": 5}, {"nam…
3  [{"name": "Legendary Pictures", "id": 923}, {"…
4        [{"name": "Walt Disney Pictures", "id": 2}]

                                production_countries release_date     revenue  \
0  [{"iso_3166_1": "US", "name": "United States o…   2009-12-10  2787965087
1  [{"iso_3166_1": "US", "name": "United States o…   2007-05-19   961000000
2  [{"iso_3166_1": "GB", "name": "United Kingdom"…   2015-10-26   880674609
3  [{"iso_3166_1": "US", "name": "United States o…   2012-07-16  1084939099
4  [{"iso_3166_1": "US", "name": "United States o…   2012-03-07   284139100

   runtime                                   spoken_languages    status  \
0    162.0  [{"iso_639_1": "en", "name": "English"}, {"iso…  Released
1    169.0           [{"iso_639_1": "en", "name": "English"}]  Released
2    148.0  [{"iso_639_1": "fr", "name": "Fran\u00e7ais"},…  Released
3    165.0           [{"iso_639_1": "en", "name": "English"}]  Released
4    132.0           [{"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
```

[4]:

```
credits_df.head()
```

[4]:

```
   movie_id                                     title  \
0     19995                                    Avatar
1       285  Pirates of the Caribbean: At World's End
2    206647                                   Spectre
3     49026                     The Dark Knight Rises
4     49529                               John Carter

                                                cast   \
0  [{"cast_id": 242, "character": "Jake Sully", "…
1  [{"cast_id": 4, "character": "Captain Jack Spa…
2  [{"cast_id": 1, "character": "James Bond", "cr…
3  [{"cast_id": 2, "character": "Bruce Wayne / Ba…
4  [{"cast_id": 5, "character": "John Carter", "c…

                                                crew
0  [{"credit_id": "52fe48009251416c750aca23", "de…
1  [{"credit_id": "52fe4232c3a36847f800b579", "de…
2  [{"credit_id": "54805967c3a36829b5002c41", "de…
3  [{"credit_id": "52fe4781c3a36847f81398c3", "de…
4  [{"credit_id": "52fe479ac3a36847f813eaa3", "de…
```

[5]:

```
credits_df.columns = ['id','tittle','cast','crew']
movies_df = movies_df.merge(credits_df, on="id")
```

[6]:

```
movies_df.head()
```

[6]:

```
      budget                                             genres  \
0  237000000  [{"id": 28, "name": "Action"}, {"id": 12, "nam…
1  300000000  [{"id": 12, "name": "Adventure"}, {"id": 14, "…
2  245000000  [{"id": 28, "name": "Action"}, {"id": 12, "nam…
3  250000000  [{"id": 28, "name": "Action"}, {"id": 80, "nam…
4  260000000  [{"id": 28, "name": "Action"}, {"id": 12, "nam…

                                       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.thedarkknightrises.com/   49026
4          http://movies.disney.com/john-carter   49529

                                           keywords original_language  \
0  [{"id": 1463, "name": "culture clash"}, {"id":…                en
1  [{"id": 270, "name": "ocean"}, {"id": 726, "na…                en
```

```
2  [{"id": 470, "name": "spy"}, {"id": 818, "name…                en
3  [{"id": 849, "name": "dc comics"}, {"id": 853,…                en
4  [{"id": 818, "name": "based on novel"}, {"id":…                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

                                production_companies  … runtime  \
0  [{"name": "Ingenious Film Partners", "id": 289…  …   162.0
1  [{"name": "Walt Disney Pictures", "id": 2}, {"…  …   169.0
2  [{"name": "Columbia Pictures", "id": 5}, {"nam…  …   148.0
3  [{"name": "Legendary Pictures", "id": 923}, {"…  …   165.0
4        [{"name": "Walt Disney Pictures", "id": 2}]  …   132.0

                                    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

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

                                                cast  \
0  [{"cast_id": 242, "character": "Jake Sully", "…
1  [{"cast_id": 4, "character": "Captain Jack Spa…
2  [{"cast_id": 1, "character": "James Bond", "cr…
3  [{"cast_id": 2, "character": "Bruce Wayne / Ba…
4  [{"cast_id": 5, "character": "John Carter", "c…

                                                crew
0  [{"credit_id": "52fe48009251416c750aca23", "de…
1  [{"credit_id": "52fe4232c3a36847f800b579", "de…
2  [{"credit_id": "54805967c3a36829b5002c41", "de…
3  [{"credit_id": "52fe4781c3a36847f81398c3", "de…
4  [{"credit_id": "52fe479ac3a36847f813eaa3", "de…

[5 rows x 23 columns]
```

[7]:

```python
# Demographic Filtering
C = movies_df["vote_average"].mean()
m = movies_df["vote_count"].quantile(0.9)

print("C: ", C)
print("m: ", m)

new_movies_df = movies_df.copy().loc[movies_df["vote_count"] >= m]
print(new_movies_df.shape)
```

```
C:  6.092171559442016
m:  1838.4000000000015
(481, 23)
```

[8]:

```python
def weighted_rating(x, C=C, m=m):
    v = x["vote_count"]
    R = x["vote_average"]

    return (v/(v + m) * R) + (m/(v + m) * C)
```

[9]:

```python
new_movies_df["score"] = new_movies_df.apply(weighted_rating, axis=1)
new_movies_df = new_movies_df.sort_values('score', ascending=False)

new_movies_df[["title", "vote_count", "vote_average", "score"]].head(10)
```

[9]:

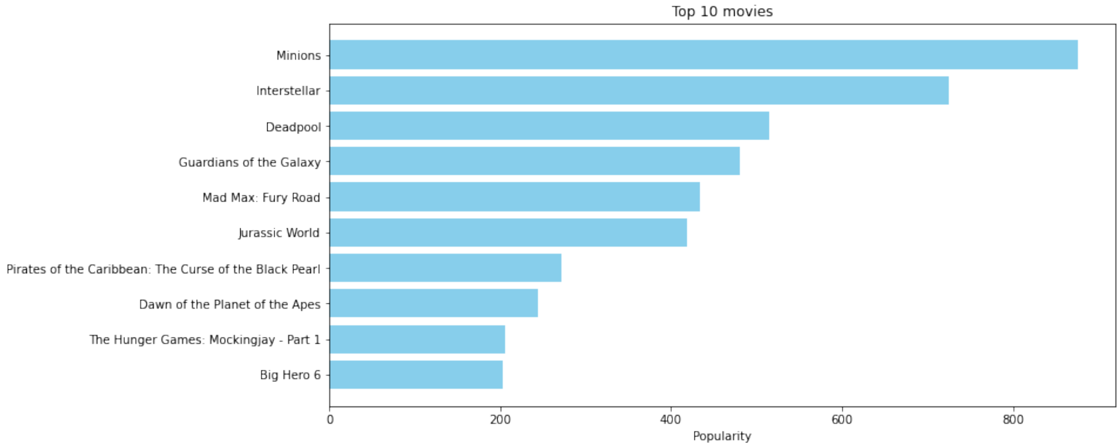
```
                                             title  vote_count  vote_average  \
1881                      The Shawshank Redemption        8205           8.5
662                                     Fight Club        9413           8.3
65                                 The Dark Knight       12002           8.2
3232                                  Pulp Fiction        8428           8.3
96                                       Inception       13752           8.1
3337                                 The Godfather        5893           8.4
95                                    Interstellar       10867           8.1
809                                   Forrest Gump        7927           8.2
329   The Lord of the Rings: The Return of the King        8064           8.1
1990                       The Empire Strikes Back        5879           8.2

         score
1881  8.059258
662   7.939256
65    7.920020
3232  7.904645
96    7.863239
3337  7.851236
95    7.809479
809   7.803188
329   7.727243
1990  7.697884
```

[10]:

```python
# Plot top 10 movies
def plot():
    popularity = movies_df.sort_values("popularity", ascending=False)
    plt.figure(figsize=(12, 6))
    plt.barh(popularity["title"].head(10), popularity["popularity"].head(10),␣
 ↪align="center", color="skyblue")
    plt.gca().invert_yaxis()
    plt.title("Top 10 movies")
    plt.xlabel("Popularity")
    plt.show()


plot()
```



[11]:

```
# Content based Filtering
print(movies_df["overview"].head(5))
```

```
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…
Name: overview, dtype: object
```

[12]:

```
tfidf = TfidfVectorizer(stop_words="english")
movies_df["overview"] = movies_df["overview"].fillna("")

tfidf_matrix = tfidf.fit_transform(movies_df["overview"])
print(tfidf_matrix.shape)
```

```
(4803, 20978)
```

[13]:

```
# Compute similarity
cosine_sim = linear_kernel(tfidf_matrix, tfidf_matrix)
print(cosine_sim.shape)

indices = pd.Series(movies_df.index, index=movies_df["title"]).drop_duplicates()
print(indices.head())
```

```
(4803, 4803)
title
Avatar                                      0
Pirates of the Caribbean: At World's End    1
Spectre                                     2
The Dark Knight Rises                       3
John Carter                                 4
```

```
dtype: int64
```

[14]:

```python
def get_recommendations(title, cosine_sim=cosine_sim):
    """
    in this function,
        we take the cosine score of given movie
        sort them based on cosine score (movie_id, cosine_score)
        take the next 10 values because the first entry is itself
        get those movie indices
        map those indices to titles
        return title list
    """
    idx = indices[title]
    sim_scores = list(enumerate(cosine_sim[idx]))
    sim_scores = sorted(sim_scores, key=lambda x: x[1], reverse=True)
    sim_scores = sim_scores[1:11]
    # (a, b) where a is id of movie, b is sim_score

    movies_indices = [ind[0] for ind in sim_scores]
    movies = movies_df["title"].iloc[movies_indices]
    return movies
```

[15]:

```python
print("################ Content Based Filtering - plot#############")
print()
print("Recommendations for The Dark Knight Rises")
print(get_recommendations("The Dark Knight Rises"))
print()
print("Recommendations for Avengers")
print(get_recommendations("The Avengers"))
```

```
################ Demographic Filtering #############

Recommendations for The Dark Knight Rises
65                              The Dark Knight
299                              Batman Forever
428                              Batman Returns
1359                                     Batman
3854    Batman: The Dark Knight Returns, Part 2
119                               Batman Begins
2507                                  Slow Burn
9            Batman v Superman: Dawn of Justice
1181                                        JFK
210                              Batman & Robin
Name: title, dtype: object

Recommendations for Avengers
7               Avengers: Age of Ultron
3144                            Plastic
```

```
1715                          Timecop
4124               This Thing of Ours
3311             Thank You for Smoking
3033                    The Corruptor
588      Wall Street: Money Never Sleeps
2136          Team America: World Police
1468                     The Fountain
1286                     Snowpiercer
Name: title, dtype: object
```

[16]:

```
features = ["cast", "crew", "keywords", "genres"]

for feature in features:
    movies_df[feature] = movies_df[feature].apply(literal_eval)

movies_df[features].head(10)
```

[16]:

```
                                                cast  \
0  [{'cast_id': 242, 'character': 'Jake Sully', '…
1  [{'cast_id': 4, 'character': 'Captain Jack Spa…
2  [{'cast_id': 1, 'character': 'James Bond', 'cr…
3  [{'cast_id': 2, 'character': 'Bruce Wayne / Ba…
4  [{'cast_id': 5, 'character': 'John Carter', 'c…
5  [{'cast_id': 30, 'character': 'Peter Parker / …
6  [{'cast_id': 34, 'character': 'Flynn Rider (vo…
7  [{'cast_id': 76, 'character': 'Tony Stark / Ir…
8  [{'cast_id': 3, 'character': 'Harry Potter', '…
9  [{'cast_id': 18, 'character': 'Bruce Wayne / B…

                                                crew  \
0  [{'credit_id': '52fe48009251416c750aca23', 'de…
1  [{'credit_id': '52fe4232c3a36847f800b579', 'de…
2  [{'credit_id': '54805967c3a36829b5002c41', 'de…
3  [{'credit_id': '52fe4781c3a36847f81398c3', 'de…
4  [{'credit_id': '52fe479ac3a36847f813eaa3', 'de…
5  [{'credit_id': '52fe4252c3a36847f80151a5', 'de…
6  [{'credit_id': '52fe46db9251416c91062101', 'de…
7  [{'credit_id': '55d5f7d4c3a3683e7e0016eb', 'de…
8  [{'credit_id': '52fe4273c3a36847f801fab1', 'de…
9  [{'credit_id': '553bf23692514135c8002886', 'de…

                                            keywords  \
0  [{'id': 1463, 'name': 'culture clash'}, {'id':…
1  [{'id': 270, 'name': 'ocean'}, {'id': 726, 'na…
2  [{'id': 470, 'name': 'spy'}, {'id': 818, 'name…
3  [{'id': 849, 'name': 'dc comics'}, {'id': 853,…
4  [{'id': 818, 'name': 'based on novel'}, {'id':…
```

```
5  [{'id': 851, 'name': 'dual identity'}, {'id': …
6  [{'id': 1562, 'name': 'hostage'}, {'id': 2343,…
7  [{'id': 8828, 'name': 'marvel comic'}, {'id': …
8  [{'id': 616, 'name': 'witch'}, {'id': 2343, 'n…
9  [{'id': 849, 'name': 'dc comics'}, {'id': 7002…

                                              genres
0  [{'id': 28, 'name': 'Action'}, {'id': 12, 'nam…
1  [{'id': 12, 'name': 'Adventure'}, {'id': 14, '…
2  [{'id': 28, 'name': 'Action'}, {'id': 12, 'nam…
3  [{'id': 28, 'name': 'Action'}, {'id': 80, 'nam…
4  [{'id': 28, 'name': 'Action'}, {'id': 12, 'nam…
5  [{'id': 14, 'name': 'Fantasy'}, {'id': 28, 'na…
6  [{'id': 16, 'name': 'Animation'}, {'id': 10751…
7  [{'id': 28, 'name': 'Action'}, {'id': 12, 'nam…
8  [{'id': 12, 'name': 'Adventure'}, {'id': 14, '…
9  [{'id': 28, 'name': 'Action'}, {'id': 12, 'nam…
```

[17]:

```
def get_director(x):
    for i in x:
        if i["job"] == "Director":
            return i["name"]
    return np.nan
```

[18]:

```
def get_list(x):
    if isinstance(x, list):
        names = [i["name"] for i in x]

        if len(names) > 3:
            names = names[:3]

        return names

    return []
```

[19]:

```
movies_df["director"] = movies_df["crew"].apply(get_director)

features = ["cast", "keywords", "genres"]
for feature in features:
    movies_df[feature] = movies_df[feature].apply(get_list)
```

[21]:

```
movies_df[['title', 'cast', 'director', 'keywords', 'genres']].head()
```

[21]:

```
                                      title  \
0                                    Avatar
1  Pirates of the Caribbean: At World's End
2                                   Spectre
```

```
3                     The Dark Knight Rises
4                               John Carter

                                               cast           director  \
0  [Sam Worthington, Zoe Saldana, Sigourney Weaver]      James Cameron
1     [Johnny Depp, Orlando Bloom, Keira Knightley]     Gore Verbinski
2      [Daniel Craig, Christoph Waltz, Léa Seydoux]         Sam Mendes
3      [Christian Bale, Michael Caine, Gary Oldman]  Christopher Nolan
4    [Taylor Kitsch, Lynn Collins, Samantha Morton]     Andrew Stanton

                                keywords                                genres
0     [culture clash, future, space war]          [Action, Adventure, Fantasy]
1     [ocean, drug abuse, exotic island]          [Adventure, Fantasy, Action]
2    [spy, based on novel, secret agent]            [Action, Adventure, Crime]
3  [dc comics, crime fighter, terrorist]                [Action, Crime, Drama]
4      [based on novel, mars, medallion]  [Action, Adventure, Science Fiction]
```

[22]:

```python
def clean_data(x):
    if isinstance(x, list):
        return [str.lower(i.replace(" ", "")) for i in x]
    else:
        if isinstance(x, str):
            return str.lower(x.replace(" ", ""))
        else:
            return ""
```

[23]:

```python
features = ['cast', 'keywords', 'director', 'genres']
for feature in features:
    movies_df[feature] = movies_df[feature].apply(clean_data)
```

[24]:

```python
def create_soup(x):
    return ' '.join(x['keywords']) + ' ' + ' '.join(x['cast']) + ' ' +␣
  ↪x['director'] + ' ' + ' '.join(x['genres'])


movies_df["soup"] = movies_df.apply(create_soup, axis=1)
print(movies_df["soup"].head())
```

```
0    cultureclash future spacewar samworthington zo…
1    ocean drugabuse exoticisland johnnydepp orland…
2    spy basedonnovel secretagent danielcraig chris…
3    dccomics crimefighter terrorist christianbale …
4    basedonnovel mars medallion taylorkitsch lynnc…
Name: soup, dtype: object
```

[25]:

```python
count_vectorizer = CountVectorizer(stop_words="english")
count_matrix = count_vectorizer.fit_transform(movies_df["soup"])
```

```
print(count_matrix.shape)

cosine_sim2 = cosine_similarity(count_matrix, count_matrix)
print(cosine_sim2.shape)

movies_df = movies_df.reset_index()
indices = pd.Series(movies_df.index, index=movies_df['title'])
```

```
(4803, 11520)
(4803, 4803)
```

[26]:

```
print("################ Content Based System - metadata #############")
print("Recommendations for The Dark Knight Rises")
print(get_recommendations("The Dark Knight Rises", cosine_sim2))
print()
print("Recommendations for Avengers")
print(get_recommendations("The Avengers", cosine_sim2))
```

```
################ Content Based System #############
Recommendations for The Dark Knight Rises
65                 The Dark Knight
119                  Batman Begins
4638    Amidst the Devil's Wings
1196                The Prestige
3073           Romeo Is Bleeding
3326              Black November
1503                      Takers
1986                      Faster
303                     Catwoman
747               Gangster Squad
Name: title, dtype: object

Recommendations for Avengers
7                   Avengers: Age of Ultron
26               Captain America: Civil War
79                               Iron Man 2
169     Captain America: The First Avenger
174                     The Incredible Hulk
85      Captain America: The Winter Soldier
31                               Iron Man 3
33                    X-Men: The Last Stand
68                                 Iron Man
94                  Guardians of the Galaxy
Name: title, dtype: object
```

[ ]: