

movie_recommendation_system

September 22, 2024

Movie Recommendation System

```
[69]: # Here we are importing all the required libraries for the project

import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
import pickle
import ast
```

```
[70]: # Importing both datasets.
movies = pd.read_csv('/home/Documents/jalal/Final Project/archive/
↳tmdb_5000_movies.csv')
credits = pd.read_csv('/home/Documents/jalal/Final Project/archive/
↳tmdb_5000_credits.csv')
```

```
[71]: # checking dataset movies. just to have an oversight.
movies.head(2)
```

```
[71]:      budget      genres \
0  237000000  [{"id": 28, "name": "Action"}, {"id": 12, "nam...
1  300000000  [{"id": 12, "name": "Adventure"}, {"id": 14, "...

      homepage      id \
0  http://www.avatarmovie.com/  19995
1  http://disney.go.com/disneypictures/pirates/  285

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

      original_title \
0  Avatar
1  Pirates of the Caribbean: At World's End

      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

                                production_companies \
0 [{"name": "Ingenious Film Partners", "id": 289...
1 [{"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

    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

                                tagline \
0                                Enter the World of Pandora.
1 At the end of the world, the adventure begins.

                                title vote_average vote_count
0                                Avatar          7.2      11800
1 Pirates of the Caribbean: At World's End      6.9      4500

```

```

[72]: # checking all columns exists in movie data set
print(movies.columns)

```

```

Index(['budget', 'genres', 'homepage', 'id', 'keywords', 'original_language',
      'original_title', 'overview', 'popularity', 'production_companies',
      'production_countries', 'release_date', 'revenue', 'runtime',
      'spoken_languages', 'status', 'tagline', 'title', 'vote_average',
      'vote_count'],
      dtype='object')

```

Here onwards we are exploring our dataset using some visualizations to get know our data a little bit more.

```

[73]: # Define budget ranges
budget_bins = [0, 100000000, 500000000, 1000000000, 2000000000, 5000000000]
budget_labels = ['0-10M', '10M-50M', '50M-100M', '100M-200M', '200M+']
movies['budget_range'] = pd.cut(movies['budget'], bins=budget_bins,
                                labels=budget_labels)

# Filter top 10 movies by budget within each range
top_movies_by_budget = movies.groupby('budget_range').apply(lambda x: x.
    nlargest(10, 'budget')).reset_index(drop=True)

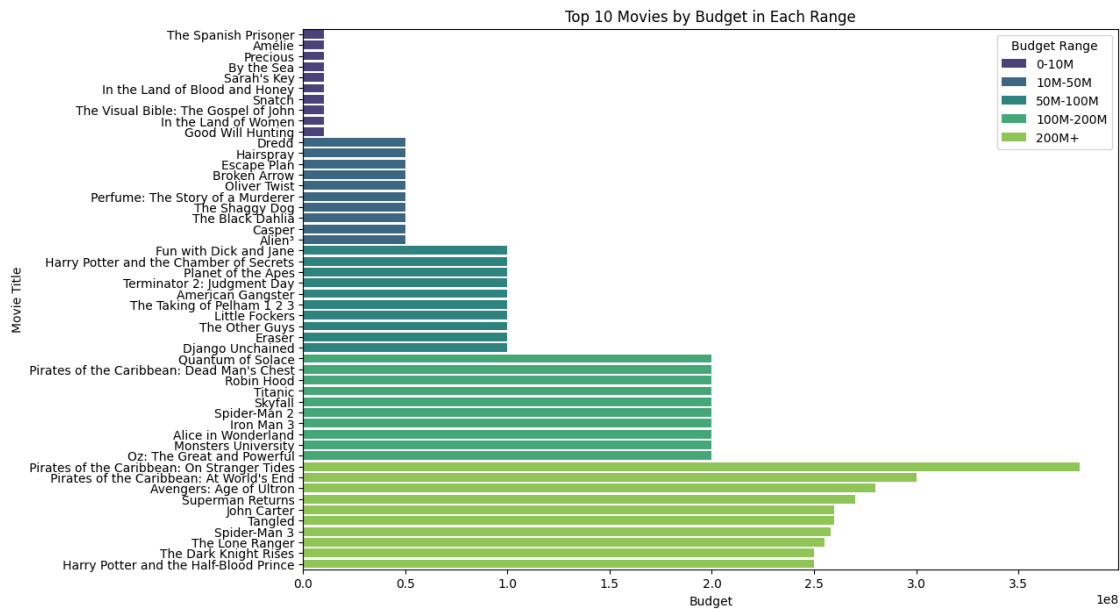
# Plot bar chart
plt.figure(figsize=(12,8))

```

```

sns.barplot(x='budget', y='title', hue='budget_range',
            data=top_movies_by_budget, palette='viridis')
plt.title('Top 10 Movies by Budget in Each Range')
plt.xlabel('Budget')
plt.ylabel('Movie Title')
plt.legend(title='Budget Range')
plt.show()

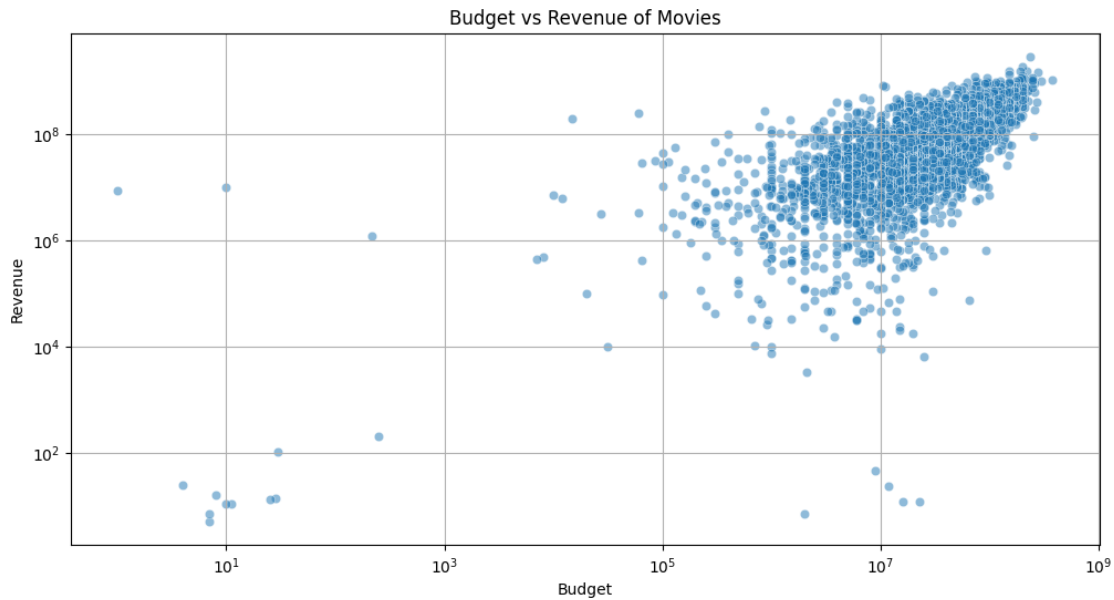
```



```

[75]: def plot_budget_vs_revenue(movies):
    plt.figure(figsize=(12, 6))
    sns.scatterplot(data=movies, x='budget', y='revenue', alpha=0.5)
    plt.title('Budget vs Revenue of Movies')
    plt.xlabel('Budget')
    plt.ylabel('Revenue')
    plt.xscale('log') # Log scale for better visibility
    plt.yscale('log') # Log scale for better visibility
    plt.grid(True)
    plt.show()
plot_budget_vs_revenue(movies)

```



```
[76]: movies.shape
```

```
[76]: (4803, 21)
```

```
[119]: # exploring our second dataset credits
credits.head()
```

```
[119]:  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...
```

```
[78]: # Here we are merging both datasets, because in Movie dataset, the crew, cast,
      ↪ and movie id columns were missing, that we need for recommendations
      movies = movies.merge(credits,on='title')
```

```
[79]: movies.head()
```

```
[79]:      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.thedarkknighttrises.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}] ...
```

	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	

	budget_range	movie_id	cast	\
0	200M+	19995	[{"cast_id": 242, "character": "Jake Sully", "...	
1	200M+	285	[{"cast_id": 4, "character": "Captain Jack Spa...	
2	200M+	206647	[{"cast_id": 1, "character": "James Bond", "cr...	
3	200M+	49026	[{"cast_id": 2, "character": "Bruce Wayne / Ba...	
4	200M+	49529	[{"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 24 columns]

```
[80]: movies = pd.read_csv('data/movies.csv')
      movies[['movie_id', 'title', 'overview', 'genres', 'keywords', 'cast', 'crew']]
```

```
[81]: movies.head()
```

	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

```

```

overview \

```

```

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...

```

```

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...

```

```

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":...

```

```

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...

```

```

[82]: def convert(text):
      L = []
      for i in ast.literal_eval(text):
          L.append(i['name'])
      return L

```

```

[83]: movies.dropna(inplace=True)

```

```

[84]: movies.head()

```

```

[84]: 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

      overview \
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...

      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...

      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":...

      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...

```

```

[85]: movies['keywords'] = movies['keywords'].apply(convert)
      movies.head()

```

```

[85]: 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

```

overview \

```

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...

```

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...

```

keywords \

```

0  [culture clash, future, space war, space colon...
1  [ocean, drug abuse, exotic island, east india ...
2  [spy, based on novel, secret agent, sequel, mi...
3  [dc comics, crime fighter, terrorist, secret i...
4  [based on novel, mars, medallion, space travel...

```

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...

```

```
[86]: ast.literal_eval(' [{"id": 28, "name": "Action"}, {"id": 12, "name": "Adventure"}, {"id": 14, "name": "Fantasy"}, {"id": 878, "name": "Science Fiction"} ]')
```

```
[86]: [{'id': 28, 'name': 'Action'},
      {'id': 12, 'name': 'Adventure'},
      {'id': 14, 'name': 'Fantasy'},
```

```
{'id': 878, 'name': 'Science Fiction']}
```

```
[88]: movies['cast'] = movies['cast'].apply(convert)
      movies.head()
```

```
[88]:  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

                                           overview \
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...

                                           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...

                                           keywords \
0  [culture clash, future, space war, space colon...
1  [ocean, drug abuse, exotic island, east india ...
2  [spy, based on novel, secret agent, sequel, mi...
3  [dc comics, crime fighter, terrorist, secret i...
4  [based on novel, mars, medallion, space travel...

                                           cast \
0  [Sam Worthington, Zoe Saldana, Sigourney Weave...
1  [Johnny Depp, Orlando Bloom, Keira Knightley, ...
2  [Daniel Craig, Christoph Waltz, Léa Seydoux, R...
3  [Christian Bale, Michael Caine, Gary Oldman, A...
4  [Taylor Kitsch, Lynn Collins, Samantha Morton,...

                                           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...
```

```
[89]: movies['cast'] = movies['cast'].apply(lambda x:x[0:3])
```

```
[90]: def fetch_director(text):
      L = []
      for i in ast.literal_eval(text):
          if i['job'] == 'Director':
              L.append(i['name'])
      return L
```

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

```
[93]: def collapse(L):
      L1 = []
      for i in L:
          L1.append(i.replace(" ", ""))
      return L1
```

```
[94]: movies['cast'] = movies['cast'].apply(collapse)
      movies['crew'] = movies['crew'].apply(collapse)
      movies['genres'] = movies['genres'].apply(collapse)
      movies['keywords'] = movies['keywords'].apply(collapse)
```

```
[95]: movies.head()
```

```
[95]:  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
```

```
          overview \
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...
```

```
          genres \
0  [[, {, ", i, d, ", :, , 2, 8, ,, , ", n, a, m,...
1  [[, {, ", i, d, ", :, , 1, 2, ,, , ", n, a, m,...
2  [[, {, ", i, d, ", :, , 2, 8, ,, , ", n, a, m,...
3  [[, {, ", i, d, ", :, , 2, 8, ,, , ", n, a, m,...
4  [[, {, ", i, d, ", :, , 2, 8, ,, , ", n, a, m,...
```

```
          keywords \
0  [cultureclash, future, spacewar, spacecolony, ...
```

```

1 [ocean, drugabuse, exoticisland, eastindiatrad...
2 [spy, basedonnovel, secretagent, sequel, mi6, ...
3 [dccomics, crimefighter, terrorist, secretiden...
4 [basedonnovel, mars, medallion, spacetravel, p...

```

	cast	crew
0	[SamWorthington, ZoeSaldana, SigourneyWeaver]	[JamesCameron]
1	[JohnnyDepp, OrlandoBloom, KeiraKnightley]	[GoreVerbinski]
2	[DanielCraig, ChristophWaltz, LéaSeydoux]	[SamMendes]
3	[ChristianBale, MichaelCaine, GaryOldman]	[ChristopherNolan]
4	[TaylorKitsch, LynnCollins, SamanthaMorton]	[AndrewStanton]

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

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

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

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

```
[99]:
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
```

```

tags
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...

```

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

```
[101]: vector = cv.fit_transform(new['tags']).toarray()
```

```
[102]: vector.shape
```

```
[102]: (4806, 5000)
```

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

```
[104]: similarity = cosine_similarity(vector)
```

```
[105]: similarity
```

```
[105]: array([[1.          , 0.          , 0.          , ..., 0.          , 0.02993422,
          0.          ],
          [0.          , 1.          , 0.          , ..., 0.02916059, 0.          ,
          0.          ],
          [0.          , 0.          , 1.          , ..., 0.02916059, 0.          ,
          0.          ],
          ...,
          [0.          , 0.02916059, 0.02916059, ..., 1.          , 0.05143445,
          0.05073463],
          [0.02993422, 0.          , 0.          , ..., 0.05143445, 1.          ,
          0.05479966],
          [0.          , 0.          , 0.          , ..., 0.05073463, 0.05479966,
          1.          ]])
```

```
[106]: new[new['title'] == 'The Lego Movie'].index[0]
```

```
[106]: 744
```

```
[107]: # This is the main function in our program, this recommends the movies
def recommend(movie):
    # Check if the movie exists in the DataFrame
    if movie in new['title'].values:
        index = new[new['title'] == movie].index[0]
        distances = sorted(list(enumerate(similarity[index])), reverse=True,
        ↪key=lambda x: x[1])

        print(f"Recommendations for {movie}:")
        for i in distances[1:6]: # Skip the first element as it is the movie
        ↪itself
            print(new.iloc[i[0]].title)
    else:
        print(f"Movie '{movie}' not found in the dataset.")
```

```
[108]: recommend('Risen')
```

```
Recommendations for Risen:
Son of God
The Robe
The Young Messiah
```

The Last Temptation of Christ
The Greatest Story Ever Told

```
[109]: recommend('Starship Troopers')
```

Recommendations for Starship Troopers:

G.I. Jane
The Four Feathers
Buffalo Soldiers
Miracle at St. Anna
The Horseman on the Roof

```
[110]: recommend('hiii') # to check if our program returns the movies that exists in  
    ↪ the database
```

Movie 'hiii' not found in the dataset.

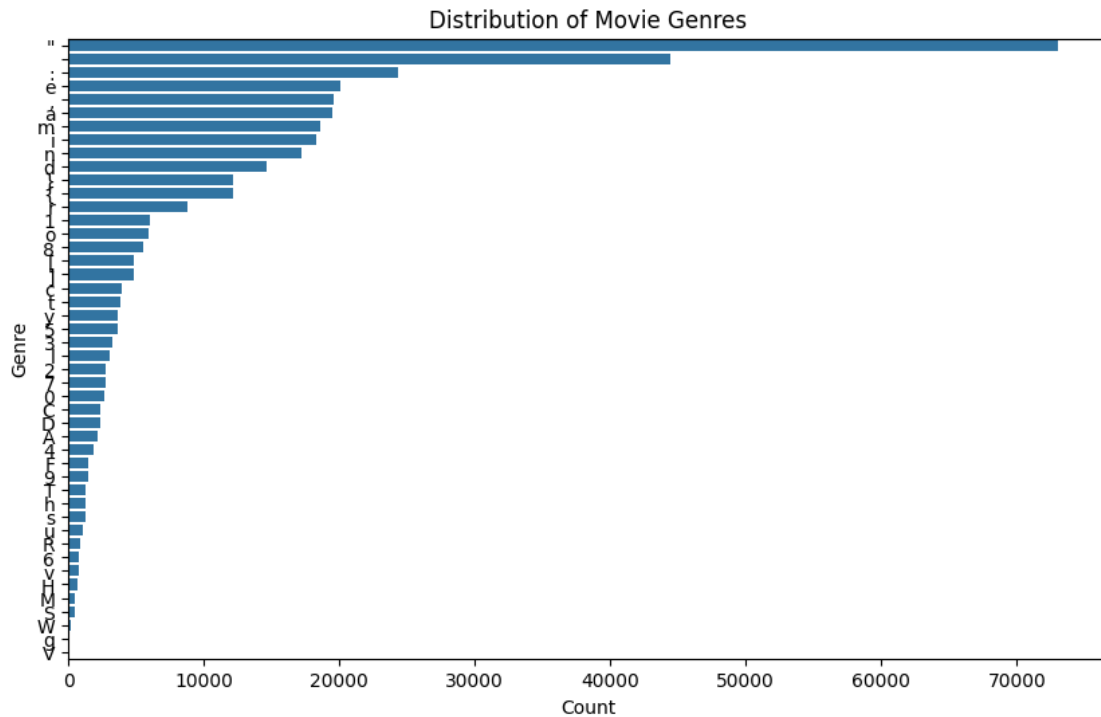
```
[111]: pickle.dump(new,open('movie_list.pkl','wb'))  
    pickle.dump(similarity,open('similarity.pkl','wb'))
```

```
[112]: new['title'].values
```

```
[112]: array(['Avatar', 'Pirates of the Caribbean: At World's End', 'Spectre',  
    ..., 'Signed, Sealed, Delivered', 'Shanghai Calling',  
    'My Date with Drew'], dtype=object)
```

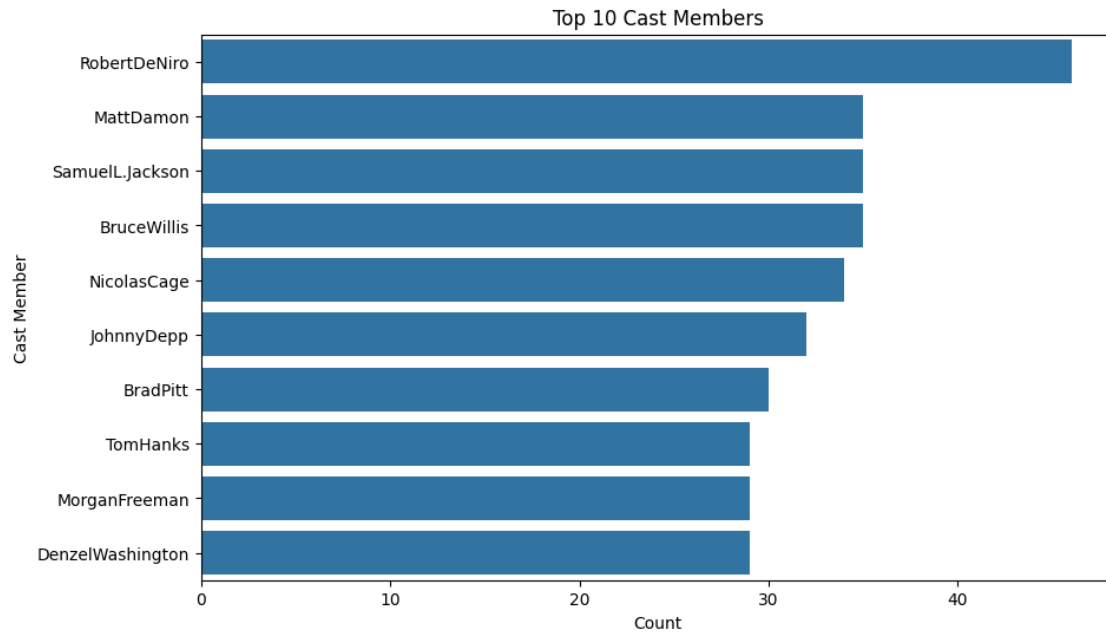
```
[113]: pickle.dump(new.to_dict(),open('movie_dict.pkl','wb'))
```

```
[114]: # Flatten the list of genres  
    genre_list = [genre for sublist in movies['genres'] for genre in sublist]  
  
    # Plot genre distribution  
    plt.figure(figsize=(10,6))  
    sns.countplot(y=genre_list, order=pd.Series(genre_list).value_counts().index)  
    plt.title('Distribution of Movie Genres')  
    plt.xlabel('Count')  
    plt.ylabel('Genre')  
    plt.show()
```



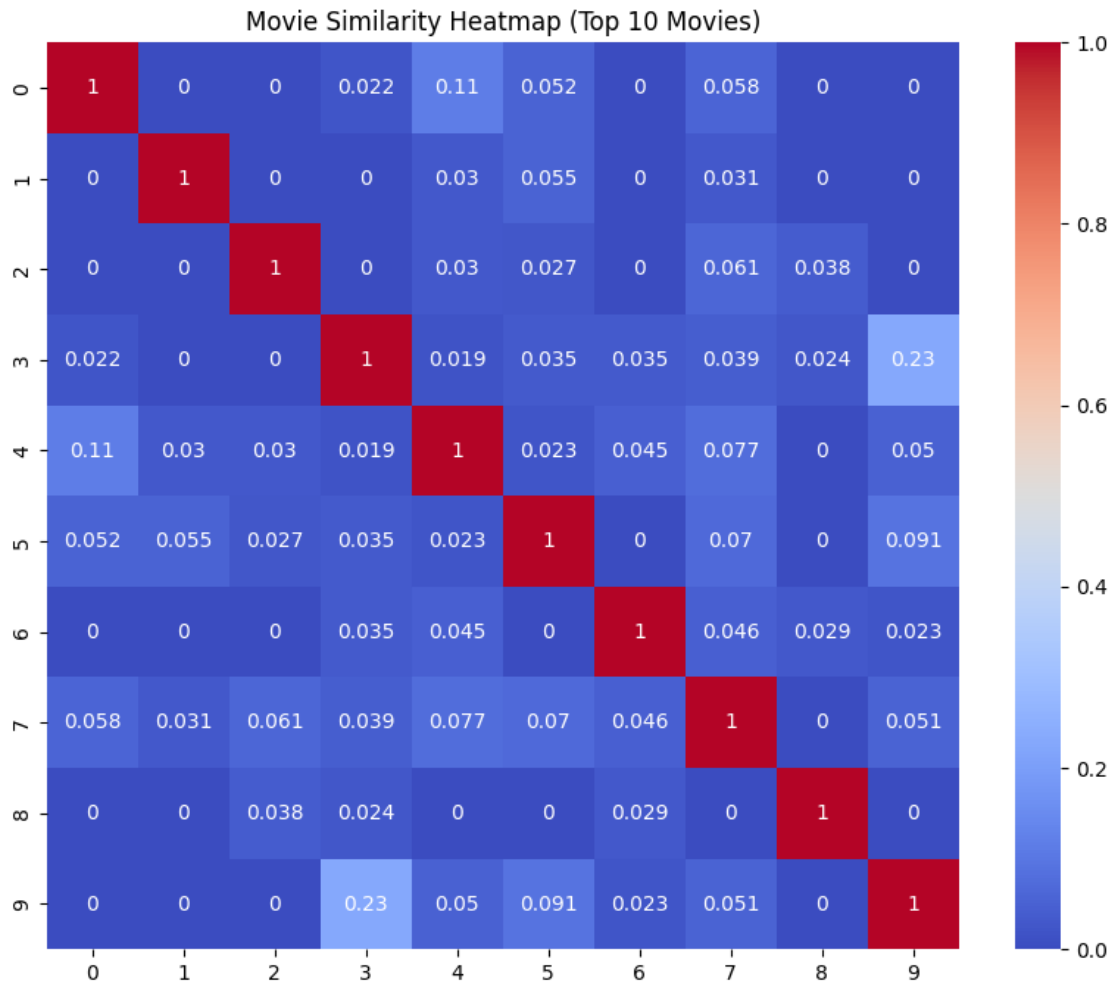
```
[115]: # Flatten the list of cast members
cast_list = [cast for sublist in movies['cast'] for cast in sublist]

# Plot top cast members
plt.figure(figsize=(10,6))
sns.countplot(y=cast_list, order=pd.Series(cast_list).value_counts().iloc[:10].
    ↪index)
plt.title('Top 10 Cast Members')
plt.xlabel('Count')
plt.ylabel('Cast Member')
plt.show()
```



```
[116]: # Create a small subset of movies for visualization purposes
small_similarity = similarity[:10, :10] # Adjust for a larger subset if needed

plt.figure(figsize=(10,8))
sns.heatmap(small_similarity, cmap="coolwarm", annot=True)
plt.title('Movie Similarity Heatmap (Top 10 Movies)')
plt.show()
```

```
[117]: def recommend_with_plot(movie):
        index = new[new['title'] == movie].index[0]
        distances = sorted(list(enumerate(similarity[index])), reverse=True,
        ↪key=lambda x: x[1])

        # Get top 5 recommendations
        recommended_movies = [new.iloc[i[0]].title for i in distances[1:6]]
        scores = [i[1] for i in distances[1:6]]

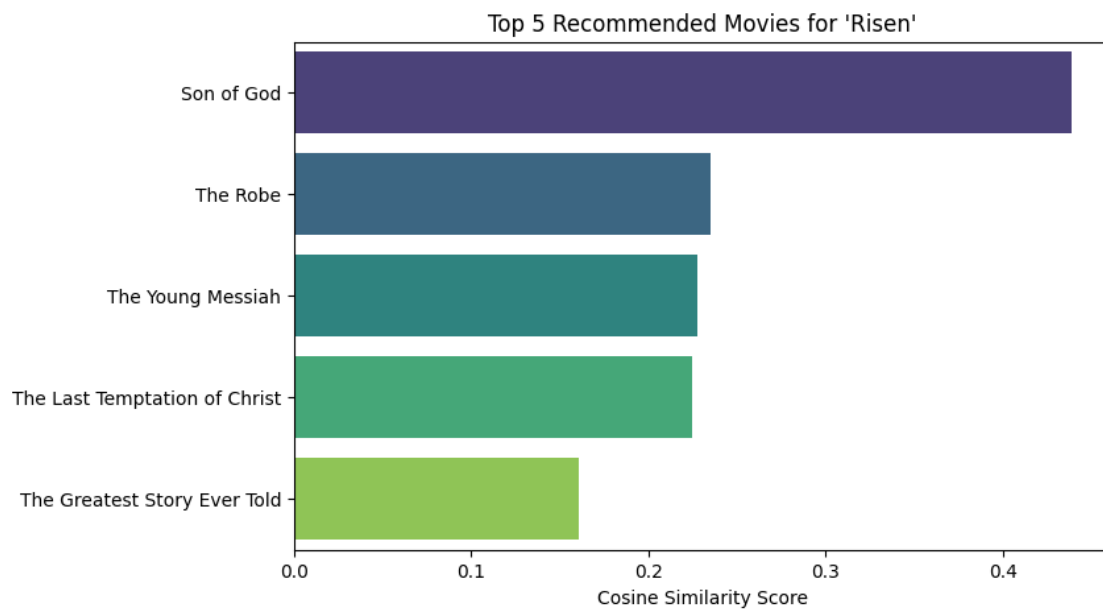
        # Plot recommendation scores
        plt.figure(figsize=(8,5))
        sns.barplot(x=scores, y=recommended_movies, palette="viridis")
        plt.title(f"Top 5 Recommended Movies for '{movie}'")
        plt.xlabel('Cosine Similarity Score')
        plt.show()
```

```
# Example
recommend_with_plot('Risen')
```

/tmp/ipykernel_35699/1125598203.py:11: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect.

```
sns.barplot(x=scores, y=recommended_movies, palette="viridis")
```



```
[118]: print(movies.columns)
```

```
Index(['movie_id', 'title', 'overview', 'genres', 'keywords', 'cast', 'crew',  
      'tags'],  
      dtype='object')
```

```
[ ]:
```