

Movie Recommendation System

BEGIN

Objective

The Movie Recommendation System project leverages data science and machine learning techniques to provide personalized movie suggestions to users. Using datasets containing movie metadata and user ratings, the system applies content-based filtering, which recommends movies similar to those the user has liked, and collaborative filtering, which suggests movies based on the preferences of users with similar behavior.



Key Features:

- Personalized Recommendations: Recommends movies based on user behavior and ratings.
- Content-based Filtering: Utilizes movie features (e.g., genre, actors, directors) to find similar movies.
- Collaborative Filtering: Suggests movies by analyzing the preferences of users with similar tastes.



Importing libraries and tools

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



Loading The Data

```
# loading the data from the csv file to a pandas dataframe  
movies_data = pd.read_csv('/content/tmdb_5000_movies.csv')
```



Representing The Data

```
# printing the first 5 rows of the dataframe
movies_data.head()
```

	budget	genres	homepage	id	keywords	original_language	original_title	overview	popularity	production_companies
0	237000000	[{"id": 28, "name": "Action"}, {"id": 12, "name": "Adventure"}, {"id": 14, "name": "Comedy"}]	http://www.avatarmovie.com/	19995	[{"id": 1463, "name": "culture clash"}, {"id": 1464, "name": "language barrier"}, {"id": 1465, "name": "space"}, {"id": 1466, "name": "war"}, {"id": 1467, "name": "water"}]	en	Avatar	In the 22nd century, a paraplegic Marine is di...	150.437577	[{"name": "Ingenious Film Partners", "id": 289}, {"name": "The Walt Disney Company", "id": 300}]]
1	300000000	[{"id": 12, "name": "Adventure"}, {"id": 14, "name": "Comedy"}, {"id": 16, "name": "Family"}]	http://disney.go.com/disneypictures/pirates/	285	[{"id": 270, "name": "ocean"}, {"id": 271, "name": "pirate"}, {"id": 272, "name": "shipwreck"}, {"id": 273, "name": "treasure"}]	en	Pirates of the Caribbean: At World's End	Captain Barbosa, long believed to be dead, ha...	139.082615	[{"name": "Walt Disney Pictures", "id": 2}, {"name": "The Walt Disney Company", "id": 300}]]
2	245000000	[{"id": 28, "name": "Action"}, {"id": 12, "name": "Adventure"}, {"id": 14, "name": "Comedy"}, {"id": 16, "name": "Family"}]	http://www.sonypictures.com/movies/spectre/	206647	[{"id": 470, "name": "spy"}, {"id": 471, "name": "agent"}, {"id": 472, "name": "bond"}, {"id": 473, "name": "double agent"}, {"id": 474, "name": "secret service"}]	en	Spectre	A cryptic message from Bond's past sends him o...	107.376788	[{"name": "Columbia Pictures", "id": 5}, {"name": "Sony Pictures", "id": 200}]]
3	250000000	[{"id": 28, "name": "Action"}, {"id": 80, "name": "Thriller"}]	http://www.thedarkknightrises.com/	49026	[{"id": 849, "name": "dc comics"}, {"id": 850, "name": "superhero"}, {"id": 851, "name": "batman"}, {"id": 852, "name": "joker"}, {"id": 853, "name": "dark knight"}]	en	The Dark Knight Rises	Following the death of District Attorney	112.312950	[{"name": "Legendary Pictures", "id": 923}, {"name": "Warner Bros.", "id": 100}]]



selecting the relevant feature for recommendation

```
movies = movies[['id','title','overview','genres','keywords']]  
  
movies.head()  
  
      id          title                                         overview           genres           keywords  
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": ...  
1    285  Pirates of the Caribbean: At World's End Captain Barbosa, long believed to be dead, ha... [{"id": 12, "name": "Adventure"}, {"id": 14, "... [{"id": 270, "name": "ocean"}, {"id": 726, "na...  
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...  
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,...  
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": ...
```



Convert String Representations of Data into Python Objects

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



concatenating the data from the columns

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

	id	title	tags
0	19995	Avatar	[culture clash, future, space war, space colon...
1	285	Pirates of the Caribbean: At World's End	[ocean, drug abuse, exotic island, east india ...
2	206647	Spectre	[spy, based on novel, secret agent, sequel, mi...
3	49026	The Dark Knight Rises	[dc comics, crime fighter, terrorist, secret i...
4	49529	John Carter	[based on novel, mars, medallion, space travel...



Converting the text data to feature vectors

```
# converting the text data to feature vectors
v = CountVectorizer(max_features=5000, stop_words=None, token_pattern=r"(?u)\b\w+\b")
vector = cv.fit_transform(new['tags']).toarray() # Added parenthesis to call toarray()
print(vector)

[0 0 2 ... 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]
[1 0 2 ... 0 0 0]]
```



Getting the similarity scores using cosine similarity

```
similarity = cosine_similarity(vector)

print(similarity.shape)

(4800, 4800)
```



Recommanding the movies on their similarity score

```
def recommend(movie):
    index = new[new['title'] == movie].index[0]
    distances = sorted(list(enumerate(similarity[index])),reverse=True,key = lambda x: x[1])
    for i in distances[1:6]:
        print(new.iloc[i[0]].title)

recommend('Gandhi')

Inside Out
Grand Theft Parsons
License to Wed
Good Dick
Spy Game
```



Importing The Pickel File

```
import pickle

pickle.dump(new,open('movie_list.pkl','wb'))
pickle.dump(similarity,open('similarity.pkl','wb'))
```

```
pickle.load(open('movie_list.pkl','rb'))
```

	id	title	tags
0	19995	Avatar	cultureclas...
1	285	Pirates of the Caribbean: At World's End	oceandruaga...
2	206647	Spectre	spybasedon...
3	49026	The Dark Knight Rises	dccomicscr...
4	49529	John Carter	basedonnov...



Create web application using Streamlit

```
app.py > ...
    import streamlit as st
    import pickle

    # Load the movie data and similarity matrix
    movies = pickle.load(open("movie_list.pkl", 'rb')) # Ensure this is a DataFrame
    similarity = pickle.load(open("similarity.pkl", 'rb'))
    movies_list = movies['title'].values

    # Streamlit app header
    st.header("Movie Recommendation System")

    # Dropdown for movie selection
    selectvalue = st.selectbox("Select movie from dropdown:", movies_list)

    # Recommendation function
    def recommend(movie):
        # Find the index of the selected movie
        index = movies[movies['title'] == movie].index[0]
        # Get similarity scores for the selected movie
```



Final output

Movie Recommendation System

Select movie from dropdown:

Avatar

Show Recommend

Pirates of the Caribbean: At World's End

Spectre

The Dark Knight Rises

John Carter

Spider-Man 3





Thank You

by Samruddhi Gorde

