Group C (Mini Project 2)

Aim:

Develop a movie recommendation model using the scikit-learn library in python.

Requirement:

- Anaconda Installer
- Windows 11 OS
- Jupyter Notebook

Theory:

What is scikit-learn?

Scikit-Learn is a free machine learning library for Python. It supports both supervised and unsupervised machine learning, providing diverse algorithms for classification, regression, clustering, and dimensionality reduction. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use.

What is a Recommendation System?

Simply put a Recommendation System is a filtration program whose prime goal is to predict the “rating” or “preference” of a user towards a domain-specific item or item. In our case, this domain-specific item is a movie, therefore the main focus of our recommendation system is to filter and predict only those movies which a user would prefer given some data about the user him or herself.

Libraries Used:

NumPy: Base n-dimensional array package

SciPy: Fundamental library for scientific computing. SciPy care conventionally named SciKits.

Pandas: Data structures and analysis

Sklearn: It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python.

Conslusion:

Hence, we successfully implemented sentiment analysis using python.

In [1]:
```
import pandas as pd
import numpy as np
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.metrics.pairwise import cosine_similarity
```

In [2]:
```
df = pd.read_csv("movie_dataset.csv")
```

In [4]:
```
df.head().T
```

Out[4]:

	0	1	
index	0	1	
budget	237000000	300000000	
genres	Action Adventure Fantasy Science Fiction	Adventure Fantasy Action	
homepage	http://www.avatarmovie.com/	http://disney.go.com/disneypictures/pirates/	http://www
id	19995	285	
keywords	culture clash future space war space colony so...	ocean drug abuse exotic island east india trad...	spy base
original_language	en	en	
original_title	Avatar	Pirates of the Caribbean: At World's End	
overview	In the 22nd century, a paraplegic Marine is di...	Captain Barbossa, long believed to be dead, ha...	A cryp
popularity	150.437577	139.082615	
production_companies	[{"name": "Ingenious Film Partners", "id": 289...	[{"name": "Walt Disney Pictures", "id": 2}, {"...	[{"name": "
production_countries	[{"iso_3166_1": "US", "name": "United States o...	[{"iso_3166_1": "US", "name": "United States o...	[{
release_date	2009-12-10	2007-05-19	
revenue	2787965087	961000000	
runtime	162.0	169.0	
spoken_languages	[{"iso_639_1": "en", "name": "English"}, {"iso...	[{"iso_639_1": "en", "name": "English"}]	[{"iso_639
status	Released	Released	
tagline	Enter the World of Pandora.	At the end of the world, the adventure begins.	
title	Avatar	Pirates of the Caribbean: At World's End	
vote_average	7.2	6.9	
vote_count	11800	4500	
cast	Sam Worthington Zoe Saldana Sigourney Weaver S...	Johnny Depp Orlando Bloom Keira Knightley Stel...	Da
crew	[{'name': 'Stephen E. Rivkin', 'gender': 0, 'd...	[{'name': 'Dariusz Wolski', 'gender': 2, 'depa...	[{'na
director	James Cameron	Gore Verbinski	

In [5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4803 entries, 0 to 4802
Data columns (total 24 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   index                 4803 non-null   int64  
 1   budget                4803 non-null   int64  
 2   genres                4775 non-null   object 
 3   homepage              1712 non-null   object 
 4   id                    4803 non-null   int64  
 5   keywords              4391 non-null   object 
 6   original_language     4803 non-null   object 
 7   original_title        4803 non-null   object 
 8   overview              4800 non-null   object 
 9   popularity            4803 non-null   float64
 10  production_companies  4803 non-null   object 
 11  production_countries  4803 non-null   object 
 12  release_date          4802 non-null   object 
 13  revenue               4803 non-null   int64  
 14  runtime               4801 non-null   float64
 15  spoken_languages      4803 non-null   object 
 16  status                4803 non-null   object 
 17  tagline               3959 non-null   object 
 18  title                 4803 non-null   object 
 19  vote_average          4803 non-null   float64
 20  vote_count            4803 non-null   int64  
 21  cast                  4760 non-null   object 
 22  crew                  4803 non-null   object 
 23  director              4773 non-null   object 
dtypes: float64(3), int64(5), object(16)
memory usage: 900.7+ KB
```

In [6]:
```
df.isnull().sum()
```

Out[6]:
```
index                     0
budget                    0
genres                   28
homepage               3091
id                        0
keywords                412
original_language         0
original_title            0
overview                  3
popularity                0
production_companies      0
production_countries      0
release_date              1
revenue                   0
runtime                   2
spoken_languages          0
status                    0
tagline                 844
title                     0
vote_average              0
vote_count                0
cast                     43
crew                      0
director                 30
dtype: int64
```

In [9]:
```
df[['genres','homepage','keywords','overview','release_date','runtime','tagline','cast','dire
```

In [10]:
```
df.isnull().sum()
```

Out[10]:
```
index                 0
budget                0
genres                0
homepage              0
id                    0
keywords              0
original_language     0
original_title        0
overview              0
popularity            0
production_companies  0
production_countries  0
release_date          0
revenue               0
runtime               0
spoken_languages      0
status                0
tagline               0
title                 0
vote_average          0
vote_count            0
cast                  0
crew                  0
director              0
dtype: int64
```

In [11]:
```
# Select the features to combine
features = ['keywords', 'cast', 'genres', 'director']

# Fill missing values with an empty string
for feature in features:
    df[feature] = df[feature].fillna('')
```

In [12]:
```
# Function to combine selected features
def combine_features(row):
    return row['keywords'] + " " + row['cast'] + " " + row["genres"] + " " + row["director"]
```

In [13]:
```
# Create a new column with combined features
df["combined_features"] = df.apply(combine_features, axis=1)
```

In [14]:
```
# Convert text data to count matrix
cv = CountVectorizer()
count_matrix = cv.fit_transform(df["combined_features"])
```

In [15]:
```
# Compute cosine similarity
cosine_sim = cosine_similarity(count_matrix)
```

In [16]:
```
# Helper functions
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]
```

In [19]:
```
# Enter a movie title
movie_user_likes = "The Dark Knight Rises"

# Find the movie index
movie_index = get_index_from_title(movie_user_likes)

# Get similarity scores
similar_movies = list(enumerate(cosine_sim[movie_index]))

# Sort movies based on similarity score
sorted_similar_movies = sorted(similar_movies, key=lambda x: x[1], reverse=True)[1:]
```

In [20]:
```
# Print top 5 similar movies
print("Top 5 similar movies to " + movie_user_likes + " are:\n")
i = 0
for element in sorted_similar_movies:
    print(get_title_from_index(element[0]))
    i += 1
    if i >= 5:
        break
```

```
Top 5 similar movies to The Dark Knight Rises are:

Batman Begins
The Dark Knight
The Killer Inside Me
The Prestige
Batman Returns
```