In [86]:

import pandas as pd

import numpy as np

from sklearn.feature\_extraction.text import TfidfVectorizer

from sklearn.metrics.pairwise import cosine\_similarity

In [87]:

df1 = pd.read\_csv("movies.csv")

print(df1.head())

movieId title \

0 1 Toy Story (1995)

1 2 Jumanji (1995)

2 3 Grumpier Old Men (1995)

3 4 Waiting to Exhale (1995)

4 5 Father of the Bride Part II (1995)

genres

0 Adventure|Animation|Children|Comedy|Fantasy

1 Adventure|Children|Fantasy

2 Comedy|Romance

3 Comedy|Drama|Romance

4 Comedy

In [88]:

df1.isna().sum()

Out[88]:

movieId 0

title 0

genres 0

dtype: int64

In [89]:

import re

def clean\_title(title):

return re.sub("[^a-zA-Z0-9 ]", "", title)

In [90]:

df1['genres\_list'] = df1['genres'].str.replace('|', ' ')

df1['clean\_title'] = df1['title'].apply(clean\_title)

movies\_data = df1[['movieId', 'clean\_title', 'genres\_list']]

print(movies\_data.head())

movieId clean\_title \

0 1 Toy Story 1995

1 2 Jumanji 1995

2 3 Grumpier Old Men 1995

3 4 Waiting to Exhale 1995

4 5 Father of the Bride Part II 1995

genres\_list

0 Adventure Animation Children Comedy Fantasy

1 Adventure Children Fantasy

2 Comedy Romance

3 Comedy Drama Romance

4 Comedy

In [91]:

df2 = pd.read\_csv("ratings.csv")

print(df2.head())

userId movieId rating timestamp

0 1 1 4.0 964982703

1 1 3 4.0 964981247

2 1 6 4.0 964982224

3 1 47 5.0 964983815

4 1 50 5.0 964982931

In [92]:

df2.isna().sum()

Out[92]:

userId 0

movieId 0

rating 0

timestamp 0

dtype: int64

In [93]:

ratings\_data = df2.drop(['timestamp'], axis=1)

print(ratings\_data.head())

userId movieId rating

0 1 1 4.0

1 1 3 4.0

2 1 6 4.0

3 1 47 5.0

4 1 50 5.0

In [94]:

combined\_data = ratings\_data.merge(movies\_data, on='movieId')

print(combined\_data.head())

userId movieId rating clean\_title \

0 1 1 4.0 Toy Story 1995

1 1 3 4.0 Grumpier Old Men 1995

2 1 6 4.0 Heat 1995

3 1 47 5.0 Seven aka Se7en 1995

4 1 50 5.0 Usual Suspects The 1995

genres\_list

0 Adventure Animation Children Comedy Fantasy

1 Comedy Romance

2 Action Crime Thriller

3 Mystery Thriller

4 Crime Mystery Thriller

In [95]:

vectorizer\_title = TfidfVectorizer(ngram\_range=(1,2))

tfidf\_title = vectorizer\_title.fit\_transform(movies\_data['clean\_title'])

def search\_by\_title(title):

title = clean\_title(title)

query\_vec = vectorizer\_title.transform([title])

similarity = cosine\_similarity(query\_vec, tfidf\_title).flatten()

indices = np.argpartition(similarity, -5)[-5:]

results = movies\_data.iloc[indices][::-1]

return results

movie\_results = search\_by\_title("Toy Story")

print(movie\_results)

movieId clean\_title \

0 1 Toy Story 1995

7355 78499 Toy Story 3 2010

2355 3114 Toy Story 2 1999

3595 4929 Toy The 1982

4089 5843 Toy Soldiers 1991

genres\_list

0 Adventure Animation Children Comedy Fantasy

7355 Adventure Animation Children Comedy Fantasy IMAX

2355 Adventure Animation Children Comedy Fantasy

3595 Comedy

4089 Action Drama

In [96]:

vectorizer\_genres = TfidfVectorizer(ngram\_range=(1,2))

tfidf\_genres = vectorizer\_genres.fit\_transform(movies\_data['genres\_list'])

def search\_similar\_genres(genres):

query\_vec = vectorizer\_genres.transform([genres])

similarity = cosine\_similarity(query\_vec, tfidf\_genres).flatten()

indices = np.argpartition(similarity, -10)[-10:]

results = movies\_data.iloc[indices][::-1]

return results

gen = 'Adventure Comedy'

print(search\_similar\_genres(gen))

movieId clean\_title \

732 952 Around the World in 80 Days 1956

3438 4686 Weekend at Bernies II 1993

2807 3752 Me Myself Irene 2000

3449 4704 Hatari 1962

6091 42009 Cheaper by the Dozen 2 2005

7682 89305 Inbetweeners Movie The 2011

3470 4734 Jay and Silent Bob Strike Back 2001

7331 77841 St Trinians 2 The Legend of Frittons Gold 2009

8010 97665 Asterix Obelix God Save Britannia Astrix et O...

8051 98697 Money Money Money Laventure cest laventure 1972

genres\_list

732 Adventure Comedy

3438 Adventure Comedy

2807 Adventure Comedy

3449 Adventure Comedy

6091 Adventure Comedy

7682 Adventure Comedy

3470 Adventure Comedy

7331 Adventure Comedy

8010 Adventure Comedy

8051 Adventure Comedy

In [97]:

def scores\_calculator(movie\_id):

#find the recommendations from users who like the same movie

similar\_users = combined\_data[(combined\_data['movieId']== movie\_id) & (combined\_data['rating']>=4)]['userId'].unique()

similar\_user\_recs = combined\_data[(combined\_data['userId'].isin(similar\_users)) & (combined\_data['rating']>=4)]['movieId']

similar\_user\_recs = similar\_user\_recs.value\_counts() / len(similar\_users)

#print(similar\_user\_recs)

#find the recommendations from all users who have watch the movies above

all\_users = combined\_data[(combined\_data['movieId'].isin(similar\_user\_recs.index)) & (combined\_data['rating']>=4)]

all\_users\_recs = all\_users['movieId'].value\_counts() / all\_users['userId'].nunique()

#print(all\_users\_recs)

genres\_of\_selected\_movie = combined\_data[combined\_data['movieId']==movie\_id]['genres\_list'].unique()

genres\_of\_selected\_movie = np.array2string(genres\_of\_selected\_movie)

movies\_with\_similar\_genres = search\_similar\_genres(genres\_of\_selected\_movie)

indices = []

for index in movies\_with\_similar\_genres[(movies\_with\_similar\_genres['movieId'].isin(similar\_user\_recs.index))]['movieId']:

indices.append(index)

#times a factor 1.5 to movies with similar genres and similar users

similar\_user\_recs.loc[indices] = similar\_user\_recs.loc[indices]\*1.5

#times a factor 0.9 to movies with similar genres and all users

indices = []

for index in movies\_with\_similar\_genres[(movies\_with\_similar\_genres['movieId'].isin(all\_users\_recs.index))]['movieId']:

indices.append(index)

all\_users\_recs.loc[indices] = all\_users\_recs.loc[indices]\*0.9

rec\_percentages = pd.concat([similar\_user\_recs, all\_users\_recs], axis=1)

rec\_percentages.columns = ['similar', 'all']

rec\_percentages['score'] = rec\_percentages['similar'] / rec\_percentages['all']

rec\_percentages = rec\_percentages.sort\_values('score', ascending=False)

return rec\_percentages

scores\_calculator(3114)

Out[97]:

|  | **similar** | **all** | **score** |
| --- | --- | --- | --- |
| **movieId** |  |  |  |
| **3114** | 1.500000 | 0.082759 | 18.125000 |
| **5772** | 0.035714 | 0.003284 | 10.875000 |
| **6971** | 0.017857 | 0.001642 | 10.875000 |
| **102716** | 0.035714 | 0.003284 | 10.875000 |
| **6969** | 0.017857 | 0.001642 | 10.875000 |
| **...** | ... | ... | ... |
| **339** | 0.035714 | 0.067323 | 0.530488 |
| **25** | 0.035714 | 0.070608 | 0.505814 |
| **454** | 0.035714 | 0.078818 | 0.453125 |
| **434** | 0.017857 | 0.039409 | 0.453125 |
| **153** | 0.017857 | 0.041051 | 0.435000 |

3463 rows × 3 columns

In [98]:

def recommendation\_results(user\_input, title=0):

# user\_input = clean\_title(user\_input)

title\_candidates = search\_by\_title(user\_input)

movie\_id = title\_candidates.iloc[title]['movieId']

scores = scores\_calculator(movie\_id)

results = scores.head(10).merge(movies\_data, left\_index=True, right\_on='movieId')[['clean\_title', 'score', 'genres\_list']]

resutls = results.rename(columns={'clean\_title': 'title', 'genres\_list': 'genres'}, inplace=True)

return results

user\_input = "Toy Story"

print("Are you looking for (please choose a number): ")

for i in range(5):

print(i, ": ", search\_by\_title(user\_input)['clean\_title'].iloc[i])

title = 0

if int(title) in range(5):

print("We have following recommendations: ")

print(recommendation\_results(user\_input, int(title)))

else:

print("Sorry! please try again!")

Are you looking for (please choose a number):

0 : Toy Story 1995

1 : Toy Story 3 2010

2 : Toy Story 2 1999

3 : Toy The 1982

4 : Toy Soldiers 1991

We have following recommendations:

title score \

2809 Adventures of Rocky and Bullwinkle The 2000 6.904762

0 Toy Story 1995 6.904762

1706 Antz 1998 5.753968

2355 Toy Story 2 1999 5.671769

2639 All the Vermeers in New York 1990 4.142857

4835 Dark Passage 1947 4.142857

4746 Red River 1948 4.142857

4742 Beauty and the Beast La belle et la bte 1946 4.142857

4740 Birdman of Alcatraz 1962 4.142857

4739 Dark Victory 1939 4.142857

genres

2809 Adventure Animation Children Comedy Fantasy

0 Adventure Animation Children Comedy Fantasy

1706 Adventure Animation Children Comedy Fantasy

2355 Adventure Animation Children Comedy Fantasy

2639 Comedy Drama Romance

4835 Crime Drama Film-Noir Romance Thriller

4746 Action Adventure Western

4742 Drama Fantasy

4740 Drama

4739 Drama Romance

In [ ]: