In [5]:

```
import pandas as pd
import matplotlib.pyplot as plt

```

In [6]:

```
df = pd.read_csv("S:\ml resources\ml-25m\movies.csv")
df.head()
```

Out[6]:

	movieId	title	genres
0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
1	2	Jumanji (1995)	Adventure|Children|Fantasy
2	3	Grumpier Old Men (1995)	Comedy|Romance
3	4	Waiting to Exhale (1995)	Comedy|Drama|Romance
4	5	Father of the Bride Part II (1995)	Comedy

In [7]:

```
# Cleaning the data using Regex
import re

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

In [8]:

```
df["cleaned_title"] = df['title'].apply(clean_text)
df
```

Out[8]:

	movieId	title	genres	cleaned_title
0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy	Toy Story 1995
1	2	Jumanji (1995)	Adventure|Children|Fantasy	Jumanji 1995
2	3	Grumpier Old Men (1995)	Comedy|Romance	Grumpier Old Men 1995
3	4	Waiting to Exhale (1995)	Comedy|Drama|Romance	Waiting to Exhale 1995
4	5	Father of the Bride Part II (1995)	Comedy	Father of the Bride Part II 1995
...	...	...	...	...
62418	209157	We (2018)	Drama	We 2018
62419	209159	Window of the Soul (2001)	Documentary	Window of the Soul 2001
62420	209163	Bad Poems (2018)	Comedy|Drama	Bad Poems 2018
62421	209169	A Girl Thing (2001)	(no genres listed)	A Girl Thing 2001
62422	209171	Women of Devil's Island (1962)	Action|Adventure|Drama	Women of Devils Island 1962

62423 rows × 4 columns

In [9]:

```
from sklearn.feature_extraction.text import TfidfVectorizer

vectorizer = TfidfVectorizer(ngram_range = (1,2))
freq = vectorizer.fit_transform(df["cleaned_title"])
```

In [10]:

```
from sklearn.metrics.pairwise import cosine_similarity
import numpy as np

def search(title):
    title = clean_text(title)
    que = vectorizer.transform([title])
    similarity = cosine_similarity(que,freq).flatten()
    index = np.argpartition(similarity,-5)[-5:]
    results = df.iloc[index][::-1]
    return results
```

In [11]:

```python
import ipywidgets as widgets
from IPython.display import display

movie_input = widgets.Text(
    value='Toy Story',
    description='Movie Title:',
    disabled=False
)
movie_list = widgets.Output()

def on_type(data):
    with movie_list:
        movie_list.clear_output()
        title = data["new"]
        if len(title) > 5:
            display(search(title))

movie_input.observe(on_type, names='value')


display(movie_input, movie_list)
```

Movie Title: Toy Story

In [12]:

```python
df2 = pd.read_csv("S:/ml resources/ml-25m/ratings.csv")
df2.head()
```

Out[12]:

	userId	movieId	rating	timestamp
0	1	296	5.0	1147880044
1	1	306	3.5	1147868817
2	1	307	5.0	1147868828
3	1	665	5.0	1147878820
4	1	899	3.5	1147868510

In [13]:

```python
# finding users who liked same movies and rated them
movieid = 1
similar = df2[(df2['movieId']== movieid) & (df2['rating']>4) ]['userId'].unique()
```

In [14]:

```python
similar
```

Out[14]:

```
array([    36,     75,     86, ..., 162527, 162530, 162533], dtype=int64)
```

In [15]:

```
similar_recs = df2[(df2['userId'].isin(similar)) & (df2['rating']>4)]['movieId']
```

In [18]:

```
similar_recs = similar_recs.value_counts() / len(similar)

similar_recs = similar_recs[similar_recs > .1]
```

In [19]:

```
print(similar_recs)
```

```
1        1.000000
318      0.445607
260      0.403770
356      0.370215
296      0.367295
           ...
953      0.103053
551      0.101195
1222     0.100876
745      0.100345
48780    0.100186
Name: movieId, Length: 113, dtype: float64
```

In [31]:

```
# finding how many people like the movies
all_users = df2[(df2["movieId"].isin(similar_recs.index)) & (df2["rating"] > 4)]
```

In [33]:

```
all_users
```

Out[33]:

	userId	movieId	rating	timestamp
0	1	296	5.0	1147880044
29	1	4973	4.5	1147869080
48	1	7361	5.0	1147880055
72	2	110	5.0	1141416589
76	2	260	5.0	1141417172
...	...	...	...	...
25000062	162541	5618	4.5	1240953299
25000065	162541	5952	5.0	1240952617
25000078	162541	7153	5.0	1240952613
25000081	162541	7361	4.5	1240953484
25000090	162541	50872	4.5	1240953372

1727573 rows × 4 columns

In [34]:

```
all_recs = all_users["movieId"].value_counts() / len(all_users['userId'].unique())
```

In [35]:

```
all_recs
```

Out[35]:

```
318      0.342220
296      0.284674
2571     0.244033
356      0.235266
593      0.225909
           ...
551      0.040918
50872    0.039111
745      0.037031
78499    0.035131
2355     0.025091
Name: movieId, Length: 113, dtype: float64
```

In [37]:

```
# creating a recommendation score
rec_per = pd.concat([similar_recs,all_recs], axis=1)
rec_per.columns=['similar','all']
```

In [39]:

```
rec_per['score'] = rec_per['similar'] / rec_per['all']
rec_per = rec_per.sort_values('score' , ascending=False)
```

In [40]:

```
rec_per
```

Out[40]:

```
        similar       all     score
    1  1.000000  0.124728  8.017414
 3114  0.280648  0.053706  5.225654
 2355  0.110539  0.025091  4.405452
78499  0.152960  0.035131  4.354038
 4886  0.235147  0.070811  3.320783
  ...       ...       ...       ...
 2858  0.216724  0.167634  1.292845
  296  0.367295  0.284674  1.290232
79132  0.166817  0.131384  1.269693
 4973  0.142501  0.112405  1.267747
 2959  0.262649  0.216717  1.211946

113 rows × 3 columns
```

In [44]:

```python
# merging recommendations with
rec_per.head(10).merge(df,left_index =True ,right_on="movieId")
```

Out[44]:

	similar	all	score	movieId	title	
0	1.000000	0.124728	8.017414	1	Toy Story (1995)	Adventure|Animation|Children|C
3021	0.280648	0.053706	5.225654	3114	Toy Story 2 (1999)	Adventure|Animation|Children|C
2264	0.110539	0.025091	4.405452	2355	Bug's Life, A (1998)	Adventure|Animation|Ch
14813	0.152960	0.035131	4.354038	78499	Toy Story 3 (2010)	Adventure|Animation|Children|Comedy
4780	0.235147	0.070811	3.320783	4886	Monsters, Inc. (2001)	Adventure|Animation|Children|C
580	0.216618	0.067513	3.208539	588	Aladdin (1992)	Adventure|Animation|Children|C
6258	0.228139	0.072268	3.156862	6377	Finding Nemo (2003)	Adventure|Animation|Ch
587	0.179400	0.059977	2.991150	595	Beauty and the Beast (1991)	Animation|Children|Fantasy|Musical|I
8246	0.203504	0.068453	2.972889	8961	Incredibles, The (2004)	Action|Adventure|Animation|Ch
359	0.253411	0.085764	2.954762	364	Lion King, The (1994)	Adventure|Animation|Children|Drama

In [60]:

```python
# Building recommendation function for above values
def find_similar_movies(movie_id):
    similar_users = df2[(df2["movieId"] == movie_id) & (df2["rating"] > 4)]["userId
    similar_user_recs = df2[(df2["userId"].isin(similar_users)) & (df2["rating"] > 
    similar_user_recs = similar_user_recs.value_counts() / len(similar_users)

    similar_user_recs = similar_user_recs[similar_user_recs > .10]
    all_users = df2[(df2["movieId"].isin(similar_user_recs.index)) & (df2["rating"]
    all_user_recs = all_users["movieId"].value_counts() / len(all_users["userId"].u
    rec_percentages = pd.concat([similar_user_recs, all_user_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.head(10).merge(df, left_index=True, right_on="movieId")[
```

In [61]:

```
movie_input_list = widgets.Text(
    value = 'Toy Story',
    description = 'Movie title:',
    disabled = False
)

recommendation_list = widgets.Output()

def on_type(data):
    with recommendation_list:
        recommendation_list.clear_output()
        title = data['new']
        if len(title)>5:
           results = search(title)
           movie_id=results.iloc[0]['movieId']
           display(find_similar_movies(movie_id))

movie_input_list.observe(on_type, names='value')

display(movie_input_list, recommendation_list)
```

Movie title: Avenger

	score	title	genres
19759	150765.0	Company of Heroes (2013)	Action|War
33750	150765.0	Operator (2015)	Action|Drama|Thriller
30808	150765.0	The Killing Game (2011)	Mystery|Thriller
31074	150765.0	Escape Clause (1996)	Thriller
31186	150765.0	Casualties (1997)	Drama|Thriller
42514	150765.0	Broken Vows (2016)	Thriller
37195	150765.0	Home Invasion (2016)	Thriller
38009	150765.0	Body Language (1995)	Romance|Thriller
39408	150765.0	Despite the Falling Snow (2016)	Drama|Romance|Thriller
28626	150765.0	Say Nothing (2001)	Action|Drama|Mystery|Romance|Sci-Fi|Thriller

In [ ]:

```

```

In [ ]:

```

```