In [127]:
```
import pandas as pd
import numpy as np
import re
```

In [128]:
```
movie=pd.read_csv('C:\\Users\\LENOVO\\Documents\\MovieGenre.csv',encoding = 'lati
```

In [129]:
```
movie.head(6)
```

Out[129]:

	imdbId	Imdb Link	Title	IMDB Score	Genre	
0	114709	http://www.imdb.com/title/tt114709	Toy Story (1995)	8.3	Animation|Adventure|Comedy	https://imag amazon.com
1	113497	http://www.imdb.com/title/tt113497	Jumanji (1995)	6.9	Action|Adventure|Family	https://imag amazon.com
2	113228	http://www.imdb.com/title/tt113228	Grumpier Old Men (1995)	6.6	Comedy|Romance	https://imag amazon.com
3	114885	http://www.imdb.com/title/tt114885	Waiting to Exhale (1995)	5.7	Comedy|Drama|Romance	https://imag amazon.com
4	113041	http://www.imdb.com/title/tt113041	Father of the Bride Part II (1995)	5.9	Comedy|Family|Romance	https://imag amazon.com
5	113277	http://www.imdb.com/title/tt113277	Heat (1995)	8.2	Action|Crime|Drama	https://imag amazon.com

In [130]:
```
movie.columns
```

Out[130]:
```
Index(['imdbId', 'Imdb Link', 'Title', 'IMDB Score', 'Genre', 'Poster'], dtype
='object')
```

In [131]:
```
movie.drop(['imdbId', 'Imdb Link', 'IMDB Score', 'Poster'],axis=1,inplace=True)
```

In [132]:
```
movie.head(6)
```

Out[132]:

	Title	Genre
0	Toy Story (1995)	Animation|Adventure|Comedy
1	Jumanji (1995)	Action|Adventure|Family
2	Grumpier Old Men (1995)	Comedy|Romance
3	Waiting to Exhale (1995)	Comedy|Drama|Romance
4	Father of the Bride Part II (1995)	Comedy|Family|Romance
5	Heat (1995)	Action|Crime|Drama

In [133]:
```
movie.shape
```

Out[133]:
```
(40108, 2)
```

In [134]:
```
#we have to clean our title with regex inorder to remove extra character
def Clean_Title(Title):
    return re.sub("[^a-zA-Z0-9 ]","",Title)
```

In [135]:
```
movie["Clean_Title"]=movie["Title"].apply(Clean_Title)
```

In [136]:
```
movie.head(6)
```

Out[136]:

	Title	Genre	Clean_Title
0	Toy Story (1995)	Animation|Adventure|Comedy	Toy Story 1995
1	Jumanji (1995)	Action|Adventure|Family	Jumanji 1995
2	Grumpier Old Men (1995)	Comedy|Romance	Grumpier Old Men 1995
3	Waiting to Exhale (1995)	Comedy|Drama|Romance	Waiting to Exhale 1995
4	Father of the Bride Part II (1995)	Comedy|Family|Romance	Father of the Bride Part II 1995
5	Heat (1995)	Action|Crime|Drama	Heat 1995

In [137]:
```
movie.drop(['Title'],axis=1,inplace=True)
```

In [138]:
```
movie.head(5)
```

Out[138]:

	Genre	Clean_Title
0	Animation|Adventure|Comedy	Toy Story 1995
1	Action|Adventure|Family	Jumanji 1995
2	Comedy|Romance	Grumpier Old Men 1995
3	Comedy|Drama|Romance	Waiting to Exhale 1995
4	Comedy|Family|Romance	Father of the Bride Part II 1995

In [139]:
```
#build TFID Matrix
from sklearn.feature_extraction.text import TfidfVectorizer
```

In [140]:
```
vc=TfidfVectorizer(ngram_range=(1,2))
Tfid=vc.fit_transform(movie["Clean_Title"])
```

In [141]:
```
Tfid
```

Out[141]:
```
<40108x113206 sparse matrix of type '<class 'numpy.float64'>'
        with 266415 stored elements in Compressed Sparse Row format>
```

In [142]:
```
#create a search function
from sklearn.metrics.pairwise import cosine_similarity
#def search(Title):
#Title= "Toy Story"
Title=Clean_Title(Title)
query_vc=vc.transform([Title])

# similarity between the search title
similarity=cosine_similarity(query_vc,Tfid).flatten()
indices=np.argpartition(similarity,-5)[-5:]
result=movie.iloc[indices]
```

In [143]:
```
result
```

Out[143]:

	Genre	Clean_Title
19897	Documentary|War	The War
23800	Drama	8 2008
14622	Documentary	My First War 2008
14659	Action|Biography|Drama	Max Manus Man of War 2008
13668	Biography|Drama|History	W 2008

In [151]:
```
#create a search function
from sklearn.metrics.pairwise import cosine_similarity
def search(Title):
    #Title= "War 2008"
    Title=Clean_Title(Title)
    query_vc=vc.transform([Title])

    # similarity between the search title
    similarity=cosine_similarity(query_vc,Tfid).flatten()
    indices=np.argpartition(similarity,-5)[-5:]
    result=movie.iloc[indices]
    return result
```

In [152]:
```
result
```

Out[152]:

	Genre	Clean_Title
19897	Documentary|War	The War
23800	Drama	8 2008
14622	Documentary	My First War 2008
14659	Action|Biography|Drama	Max Manus Man of War 2008
13668	Biography|Drama|History	W 2008

In [146]:
```
#Build Interactive Search Box
import ipywidgets as widgets
from IPython.display import display
```

In [149]:
```
movie_input=widgets.Text(
    value="Toy Story",
    description= "Movie Title:",
    disabled=False
)
# create an output widget
movie_list=widgets.Output()
def on_type(data):
    with movie_list:
        movie_list.clear_output()
        Title =data["new"]
        if len(Title)>7:
            display(search(Title))
movie_input.observe(on_type, names='value')

display(movie_input,movie_list)
```

Movie Title: the spider

	Genre	Clean_Title
6099	Drama|Mystery|Thriller	Spider 2002
23838	Short|Action|Drama	Spider 2007
6677	Drama	Kiss of the Spider Woman 1985
23233	Horror|Sci-Fi|Thriller	Earth vs the Spider 2001
22637	Horror|Sci-Fi	The Spider 1958

In [150]:
```
movie_input=widgets.Text(
    value="Toy Story",
    description= "Movie Title:",
    disabled=False
)
# create an output widget
movie_list=widgets.Output()
def on_type(data):
    with movie_list:
        movie_list.clear_output()
        Title =data["new"]
        if len(Title)>7:
            display(search(Title))
movie_input.observe(on_type, names='value')

display(movie_input,movie_list)
```

Movie Title: the man

	Genre	Clean_Title
16918	Crime|Drama|Film-Noir	The Man I Love 1947
11314	Comedy|Drama|Romance	Trust the Man 2005
10900	Comedy|Romance|Sport	Shes the Man 2006
13992	Drama	The Man 1972
10431	Action|Comedy|Crime	The Man 2005

In [ ]: