In [32]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import sys
import pickle
```

In [34]:
```
movies = pd.read_csv('movie.csv')
tags = pd.read_csv('tag.csv')
ratings = pd.read_csv('rating1.csv')
```

In [35]:
```
movies.head()
```

Out[35]:

	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 [36]:
```
tags.head()
```

Out[36]:

	userId	movieId	tag	timestamp
0	18	4141	Mark Waters	2009-04-24 18:19:40
1	65	208	dark hero	2013-05-10 01:41:18
2	65	353	dark hero	2013-05-10 01:41:19
3	65	521	noir thriller	2013-05-10 01:39:43
4	65	592	dark hero	2013-05-10 01:41:18

In [37]:
```
ratings.head()
```

Out[37]:

	userId	movieId	rating	timestamp
0	1.0	2.0	3.5	02-04-2005 23:53
1	1.0	29.0	3.5	02-04-2005 23:31
2	1.0	32.0	3.5	02-04-2005 23:33
3	1.0	47.0	3.5	02-04-2005 23:32
4	1.0	50.0	3.5	02-04-2005 23:29

In [38]:
```
movies['genres'] = movies['genres'].str.replace('|',' ')
```

In [39]:
```
len(movies.movieId.unique())
```

Out[39]:
```
27278
```

In [40]:
```
len(ratings.movieId.unique())
```

Out[40]:
```
13609
```

In [41]:
```
ratings_f = ratings.groupby('userId').filter(lambda x : len(x) >= 55)
movie_list_rating = ratings_f.movieId.unique().tolist()
```

In [42]:
```
len(ratings_f.movieId.unique())/len(movies.movieId.unique()) *100
```

Out[42]: 49.53442334482

In [43]:
```
len(ratings_f.userId.unique())/len(ratings.userId.unique()) *100
```

Out[43]: 58.80814496543994

In [44]:
```
movies = movies[movies.movieId.isin(movie_list_rating)]
```

In [45]:
```
movies.head()
```

Out[45]:

	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 [46]:
```
mapping_file = dict(zip(movies.title.tolist(), movies.movieId.tolist()))
```

In [47]:
```
tags.drop(['timestamp'],1,inplace= True)
ratings_f.drop(['timestamp'],1,inplace= True)
```

In [48]:
```
mixed = pd.merge(movies , tags , on='movieId',how ='left')
mixed.head(3)
```

Out[48]:

	movieId	title	genres	userId	tag
0	1	Toy Story (1995)	Adventure Animation Children Comedy Fantasy	1644.0	Watched
1	1	Toy Story (1995)	Adventure Animation Children Comedy Fantasy	1741.0	computer animation
2	1	Toy Story (1995)	Adventure Animation Children Comedy Fantasy	1741.0	Disney animated feature

In [49]:
```
mixed.fillna("", inplace = True)
mixed = pd.DataFrame(mixed.groupby('movieId')['tag'].apply( lambda x: "%s" % ' '.joi
final = pd.merge(movies , mixed , on ='movieId', how ='left')
final['metadata'] = final[['tag','genres']].apply(lambda x : ' '.join(x), axis =1 )
final[['movieId','title','metadata']].head(3)
```

Out[49]:

	movieId	title	metadata
0	1	Toy Story (1995)	Watched computer animation Disney animated fea...
1	2	Jumanji (1995)	time travel adapted from:book board game child...
2	3	Grumpier Old Men (1995)	old people that is actually funny sequel fever...

In [50]:
```
final.shape
```

Out[50]:
```
(13512, 5)
```

In [51]:
```
final.loc[0,"metadata"]
```

Out[51]:
```
"Watched computer animation Disney animated feature Pixar animation TÃ©a Leoni doe
not star in this movie Pixar animation family Tom Hanks Pixar witty Pixar adventure
animated animation clever comedy computer animation family fantasy Tom Hanks bright
DARING RESCUES fanciful HEROIC MISSION humorous light rousing TOYS COME TO LIFE UNLI
KELY FRIENDSHIPS warm witty animation humorous Pixar time travel Pixar Pixar animati
on animation kids movie Pixar Pixar Pixar witty Disney Tim Allen time travel action
figure action figures Buzz Lightyear CG animation toy toys Woody animation Pixar ani
mation Disney villian hurts toys pixar animation disney fantasy Pixar animation pixa
r children é˜®ä¸€é¸£ animation computer animation funny humorous Pixar Tom Hanks wit
ty 3D Disney funny Pixar time travel Pixar time travel animation Pixar Cartoon Disne
y toy toys Pixar Pixar animation pixar animated animation comedy Disney Pixar ya boy
clever computer animation Disney fantasy Pixar toys witty animation cgi rated-G Pixa
r children computer animation family funny Pixar Tom Hanks toys lots of heart Animat
ion Pixar want to see again children Disney computer animation funny Pixar animation
fantasy Pixar animation Pixar Disney Pixar Tim Allen Tom Hanks Pixar animation comed
y Disney Pixar imdb top 250 animation pixar Tim Allen Tom Hanks 3D animated children
comedy computer animation Disney family humorous Pixar time travel Tom Hanks childre
n Pixar Tom Hanks animation Pixar animated animation buddy movie computer animation
funny Pixar Tom Hanks Tom Hanks Cartoon animation comedy funny imdb top 250 Pixar Pi
xar Tom Hanks pixar animation cgi Disney family Pixar toys computer animation Pixar
children family Pixar Tom Hanks toys witty Pixar the boys Pixar animated cgi comedy
animated animation children comedy fantasy funny humorous Pixar time travel very goo
d Best of Rotten Tomatoes: All Time John Lasseter Pixar animation computer animation
pixar toys adventure animation comedy family fantasy John Lasseter USA adventure chi
ldren classic computer animation Disney funny Pixar Tim Allen Tom Hanks animation Pi
xar adventure children family funny animation Tom Hanks avi buy animated fun pixar c
omputer animation 3D children Want classic pixar children computer animation family
humorous time travel Tom Hanks witty Pixar animation pixar Pixar CGI classic disney
pixar pixar animation Disney Pixar soothing Tom Hanks almost favorite toys computer
animation Disney humorous Pixar funny Pixar adventure animated animation buddy movie
children classic clever comedy computer animation Disney family fantasy funny humoro
us imdb top 250 Pixar time travel Tom Hanks toys witty adventure animation children
comedy Disney animation fun animation children clever Disney family funny humorous i
mdb top 250 Pixar Pixar animation Tom Hanks Pixar Disney Pixar adventure animated an
imation classic Disney fantasy Pixar Tom Hanks toys animation children computer anim
ation Disney family Pixar animation Pixar animation friendship toys computer animati
on Pixar adventure computer animation Pixar pixar animation Pixar Tim Allen Tom Hank
s family film friendship toys cute funny story voice acting witty classic Disney Pix
ar animation Pixar animation classic comedy computer animation Disney funny humorous
Pixar time travel Tom Hanks witty first cgi film animation children Disney animation
children computer animation Disney imdb top 250 John Lasseter Pixar Tom Hanks Engagi
ng animation comedy funny Pixar 2009 reissue in Stereoscopic 3-D 55 movies every kid
should see--Entertainment Weekly BD-Video CLV DVD-Video animation children Disney Pi
xar animation animated animation buddy movie children clever time travel witty kids
and family Pixar witty animation erlend's DVDs funny Pixar witty innovative buddy mo
vie Tom Hanks witty time travel dolls National Film Registry adventure animation com
edy funny humorous Pixar animation Disney Pixar toys adventure funny Tumey's To See
Again Tumey's VHS Adventure Animation Children Comedy Fantasy"
```

In [52]:
```
from sklearn.feature_extraction.text import TfidfVectorizer
tfidf = TfidfVectorizer(stop_words='english')
tfidf_matrix = tfidf.fit_transform(final['metadata'])
tfidf_df= pd.DataFrame(tfidf_matrix.toarray(),index=final.index.tolist())
```

In [53]:
```
print(tfidf_df.shape)
```

```
(13512, 21295)
```

In [54]:
```
tfidf_df
```

Out[54]:

	0	1	2	3	4	5	6	7	8	9	...	21285	21286	21287	21288	21289	212

	0	1	2	3	4	5	6	7	8	9	...	21285	21286	21287	21288	21289	212
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	0.0	0.0	
1	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	0.0	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	0.0	0.0	
3	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	0.0	0.0	
4	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	0.0	0.0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
13507	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	0.0	0.0	
13508	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	0.0	0.0	
13509	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	0.0	0.0	
13510	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	0.0	0.0	
13511	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	0.0	0.0	

13512 rows × 21295 columns

In [55]:
```
tfidf_df.loc(0)
```

Out[55]:
```
<pandas.core.indexing._LocIndexer at 0x11c942b7090>
```

In [ ]:

In [ ]:

In [56]:
```
from sklearn.decomposition import TruncatedSVD
svd = TruncatedSVD(n_components = 200)
latent_matrix = svd.fit_transform(tfidf_df)
# explained = svd.explained_varience_ratio.cunsum()
```

In [57]:
```
n=200
latent_matrix_1_df = pd.DataFrame(latent_matrix[:,0:n],index=final.title.tolist())
latent_matrix.shape
```

Out[57]:
```
(13512, 200)
```

In [58]:
```
latent_matrix
```

Out[58]:
```
array([[ 4.86551571e-02,  5.45150074e-02,  2.73885722e-02, ...,
        -1.76173777e-02, -2.12158809e-02, -2.44521746e-02],
       [ 2.42914060e-02,  1.09861969e-02,  4.47227687e-02, ...,
        -5.79036959e-03, -4.04206271e-03, -1.51421231e-02],
       [ 6.10138801e-02,  6.57547493e-02, -6.02443225e-04, ...,
        -5.27462253e-03, -6.58785915e-03,  4.21624741e-03],
       ...,
       [ 4.29819080e-01, -2.87452339e-01,  4.00509987e-01, ...,
        -2.31573308e-03, -1.23724881e-03, -4.17755471e-04],
       [ 7.66240107e-02,  1.00308724e-02,  1.22924198e-01, ...,
         1.57081952e-02,  8.47801446e-03,  2.53304712e-03],
       [ 4.04459204e-01, -2.81481140e-01,  6.24525033e-01, ...,
        -3.08445747e-03, -6.60260934e-04, -2.78113722e-03]])
```

In [ ]:

In [ ]:
```
```

In [59]:
```
ratings_f.head()
```

Out[59]:

	userId	movieId	rating
0	1.0	2.0	3.5
1	1.0	29.0	3.5
2	1.0	32.0	3.5
3	1.0	47.0	3.5
4	1.0	50.0	3.5

In [60]:
```
ratings_f1=pd.merge(movies[['movieId']],ratings_f,on="movieId",how="right")
ratings_f2 = ratings_f1.pivot(index = 'movieId', columns = 'userId', values ='rating
ratings_f2.head(3)
```

Out[60]:

userId	1.0	2.0	3.0	5.0	7.0	8.0	11.0	13.0	14.0	16.0	...	5337.0	5338.0	5340.0	5341.0
movieId															
1	0.0	0.0	4.0	0.0	0.0	4.0	4.5	4.0	4.5	3.0	...	3.5	5.0	0.0	0.0
2	3.5	0.0	0.0	3.0	0.0	0.0	0.0	3.0	0.0	0.0	...	2.5	0.0	0.0	0.0
3	0.0	4.0	0.0	0.0	3.0	5.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0

3 rows × 3148 columns

In [61]:
```
ratings_f2.shape
```

Out[61]:
```
(13512, 3148)
```

In [62]:
```
len(ratings_f.movieId.unique())
```

Out[62]:
```
13512
```

In [63]:
```
from sklearn.decomposition import TruncatedSVD
svd = TruncatedSVD(n_components = 200)
latent_matrix_2 = svd.fit_transform(ratings_f2)
latent_matrix_2_df=pd.DataFrame(latent_matrix_2,index=final.title.tolist())
```

In [64]:
```
latent_matrix_2_df.shape
```

Out[64]:
```
(13512, 200)
```

In [66]:
```
latent_matrix_1_df.head()
```

Out[66]:

	0	1	2	3	4	5	6	7	
Toy Story (1995)	0.048655	0.054515	0.027389	-0.006518	0.067833	0.083618	-0.021187	0.130454	-0.009
Jumanji (1995)	0.024291	0.010986	0.044723	0.002007	0.037822	0.072494	-0.003981	0.089086	-0.011

	0	1	2	3	4	5	6	7	
Grumpier Old Men (1995)	0.061014	0.065755	-0.000602	0.009389	0.005679	0.022065	0.024684	-0.005944	-0.001
Waiting to Exhale (1995)	0.169644	0.044606	-0.025022	0.035987	-0.006459	0.068136	0.088714	-0.035772	-0.005
Father of the Bride Part II (1995)	0.070181	0.082221	0.009964	-0.000787	0.016626	0.011374	-0.015768	0.025338	-0.001

5 rows × 200 columns

In [67]:
```
latent_matrix_2_df.head()
```

Out[67]:

	0	1	2	3	4	5	6	7
Toy Story (1995)	114.007390	-1.573990	26.136520	15.953648	5.843151	35.990086	-3.399555	7.665011
Jumanji (1995)	53.023975	-0.054846	31.700414	-6.739358	-10.451463	5.997618	-10.925832	10.561201
Grumpier Old Men (1995)	23.355985	-8.717053	16.705118	-8.190293	-9.342945	0.178811	-0.305697	-0.404921
Waiting to Exhale (1995)	5.476817	-5.402270	4.627437	-1.751648	-4.668420	-1.349797	1.742444	1.029241
Father of the Bride Part II (1995)	20.414781	-8.791025	21.007581	-9.035887	-12.250778	1.195477	-0.447868	0.468119

5 rows × 200 columns

In [ ]:

In [ ]:

In [ ]:

In [75]:
```
from sklearn.metrics.pairwise import cosine_similarity
a_1 = np.array(latent_matrix_1_df.loc['Toy Story (1995)']).reshape(1,-1)
a_2 = np.array(latent_matrix_2_df.loc['Toy Story (1995)']).reshape(1,-1)
```

In [76]:
```
score_1=cosine_similarity(latent_matrix_1_df,a_1).reshape(-1)
score_2=cosine_similarity(latent_matrix_2_df,a_2).reshape(-1)
```

In [77]:
```
hybrid = ((score_1+score_2)/2.0)
```

In [78]:
```
dictDf = {'content':score_1,'colloborative':score_2,'hybrid':hybrid}
```

```
similar = pd.DataFrame(dictDf, index = latent_matrix_1_df.index)
```

In [80]:
```
similar.sort_values('hybrid',ascending =False , inplace = True)
```

In [81]:
```
similar[1:].head(11)
```

Out[81]:

	content	colloborative	hybrid
Toy Story 2 (1999)	0.966038	0.751993	0.859015
Bug's Life, A (1998)	0.909306	0.669680	0.789493
Monsters, Inc. (2001)	0.890918	0.626687	0.758803
Finding Nemo (2003)	0.882317	0.605464	0.743890
Ice Age (2002)	0.881194	0.474145	0.677669
Incredibles, The (2004)	0.796920	0.557879	0.677400
Ratatouille (2007)	0.898907	0.401276	0.650091
Antz (1998)	0.751994	0.541300	0.646647
Toy Story 3 (2010)	0.864853	0.379689	0.622271
Shrek (2001)	0.574865	0.641949	0.608407
Up (2009)	0.760080	0.398540	0.579310

In [83]:
```
similar.sort_values('content',ascending =False , inplace = True)
similar[1:].head(11)
```

Out[83]:

	content	colloborative	hybrid
Toy Story 2 (1999)	0.966038	0.751993	0.859015
Bug's Life, A (1998)	0.909306	0.669680	0.789493
Ratatouille (2007)	0.898907	0.401276	0.650091
Monsters, Inc. (2001)	0.890918	0.626687	0.758803
Finding Nemo (2003)	0.882317	0.605464	0.743890
Ice Age (2002)	0.881194	0.474145	0.677669
Toy Story 3 (2010)	0.864853	0.379689	0.622271
Monsters University (2013)	0.820516	0.169262	0.494889
Tin Toy (1988)	0.799957	0.053521	0.426739
Red's Dream (1987)	0.797502	0.053521	0.425512
Incredibles, The (2004)	0.796920	0.557879	0.677400

In [84]:
```
similar.sort_values('colloborative',ascending =False , inplace = True)
similar[1:].head(11)
```

Out[84]:

	content	colloborative	hybrid
Toy Story 2 (1999)	0.966038	0.751993	0.859015
Forrest Gump (1994)	0.275978	0.699971	0.487975
Aladdin (1992)	0.389957	0.697274	0.543616

	content	colloborative	hybrid
Jurassic Park (1993)	0.059816	0.696833	0.378324
Back to the Future (1985)	0.150203	0.695755	0.422979
Lion King, The (1994)	0.430952	0.690957	0.560954
Independence Day (a.k.a. ID4) (1996)	0.005165	0.689500	0.347332
Star Wars: Episode IV - A New Hope (1977)	0.028591	0.687976	0.358283
Star Wars: Episode VI - Return of the Jedi (1983)	0.016794	0.673369	0.345082
Mission: Impossible (1996)	0.239535	0.670785	0.455160
Bug's Life, A (1998)	0.909306	0.669680	0.789493

In [ ]: