In [4]:
```python
import pandas as pd
import numpy as np
import sklearn
from sklearn.decomposition import TruncatedSVD
import sklearn
from sklearn.neighbors import NearestNeighbors
from matplotlib import pyplot as plt
```

In [5]:
```python
# Import the data
BX_BOOKS = pd.read_csv(r"C:\Users\ahmoh\OneDrive\Desktop\Ahmed Project\ML AHMED\recommndation\Book Recomender\BX-Books.csv", sep=';', encoding = 'mbcs', error_bad_lines=False)
BX_Book_Ratings = pd.read_csv(r"C:\Users\ahmoh\OneDrive\Desktop\Ahmed Project\ML AHMED\recommndation\Book Recomender\BX-Book-Ratings.csv", encoding = 'mbcs', error_bad_lines=False)
```

```
b'Skipping line 6452: expected 8 fields, saw 9\nSkipping line 43667: expected 8 fields, saw 10\nSkipping line 51751: expected 8 fields, saw 9\n'
b'Skipping line 92038: expected 8 fields, saw 9\nSkipping line 104319: expected 8 fields, saw 9\nSkipping line 121768: expected 8 fields, saw 9\n'
b'Skipping line 144058: expected 8 fields, saw 9\nSkipping line 150789: expected 8 fields, saw 9\nSkipping line 157128: expected 8 fields, saw 9\nSkipping line 180189: expected 8 fields, saw 9\nSkipping line 185738: expected 8 fields, saw 9\n'
b'Skipping line 209388: expected 8 fields, saw 9\nSkipping line 220626: expected 8 fields, saw 9\nSkipping line 227933: expected 8 fields, saw 11\nSkipping line 228957: expected 8 fields, saw 10\nSkipping line 245933: expected 8 fields, saw 9\nSkipping line 251296: expected 8 fields, saw 9\nSkipping line 259941: expected 8 fields, saw 9\nSkipping line 261529: expected 8 fields, saw 9\n'
C:\Users\ahmoh\anaconda3\lib\site-packages\IPython\core\interactiveshell.py:3063: DtypeWarning: Columns (3) have mixed types.Specify dtype option on import or set low_memory=False.
  interactivity=interactivity, compiler=compiler, result=result)
```

In [4]:
```python
BX_BOOKS_new=BX_BOOKS
```

In [5]:
```python
# cleaning the data
BX_BOOKS['Year-Of-Publication'] = BX_BOOKS['Year-Of-Publication'].replace(['DK Publishing Inc','Gallimard'], 0)
BX_BOOKS['Year-Of-Publication']=BX_BOOKS['Year-Of-Publication'].astype(int)
BX_BOOKS['ISBN'] = BX_BOOKS['ISBN'].replace(['074322678X','080652121X'], 0)
```

In [6]:
```python
# dropping unneeded columns
BX_BOOKS=BX_BOOKS.drop(['Image-URL-S','Image-URL-M','Image-URL-L'], axis=1)
BX_BOOKS = BX_BOOKS.drop_duplicates()
BX_BOOKS=BX_BOOKS.dropna(axis=0)
```

In [7]:
```python
BX_BOOKS.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 271357 entries, 0 to 271359
Data columns (total 5 columns):
 #   Column               Non-Null Count   Dtype 
---  ------               --------------   ----- 
 0   ISBN                 271357 non-null  object
 1   Book-Title           271357 non-null  object
 2   Book-Author          271357 non-null  object
 3   Year-Of-Publication  271357 non-null  int32 
 4   Publisher            271357 non-null  object
dtypes: int32(1), object(4)
memory usage: 11.4+ MB
```

In [8]:
```python
# clean the data
BX_Book_Ratings = BX_Book_Ratings.drop_duplicates()
BX_Book_Ratings=BX_Book_Ratings.dropna(axis=0)
```

In [9]:
```python
BX_Book_Ratings.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 504320 entries, 0 to 504401
Data columns (total 3 columns):
 #   Column       Non-Null Count   Dtype 
---  ------       --------------   ----- 
 0   User-ID      504320 non-null  int64 
 1   ISBN         504320 non-null  object
 2   Book-Rating  504320 non-null  int64 
dtypes: int64(2), object(1)
memory usage: 15.4+ MB
```

In [10]:
```python
BX_BOOKS.head()
```

Out[10]:

	ISBN	Book-Title	Book-Author	Year-Of-Publication	Publisher
0	0195153448	Classical Mythology	Mark P. O. Morford	2002	Oxford University Press
1	0002005018	Clara Callan	Richard Bruce Wright	2001	HarperFlamingo Canada
2	0060973129	Decision in Normandy	Carlo D'Este	1991	HarperPerennial
3	0374157065	Flu: The Story of the Great Influenza Pandemic...	Gina Bari Kolata	1999	Farrar Straus Giroux
4	0393045218	The Mummies of Urumchi	E. J. W. Barber	1999	W. W. Norton & Company

In [11]:
```python
BX_Book_Ratings.head()
```

Out[11]:

	User-ID	ISBN	Book-Rating
0	276725	034545104X	0
1	276726	155061224	5
2	276727	446520802	0
3	276729	052165615X	3
4	276729	521795028	6

Rating based of Count of rating

In [12]:
```python
# grouuping the books based on the number of ratings
rating_count = pd.DataFrame(BX_Book_Ratings.groupby('ISBN')['Book-Rating'].count())

rating_count.sort_values('Book-Rating', ascending=False).head()
```

Out[12]:

ISBN	Book-Rating
971880107	1107
316666343	557
385504209	409
60928336	322
312195516	318

In [13]:
```python
# obtaing the names of the books and information for the top rated books
most_rated_books = pd.DataFrame([971880107, 316666343, 385504209, 60928336, 312195516], index=np.arange(5), columns=['ISBN'])
most_rated_books['ISBN'] = most_rated_books['ISBN'].astype(str)
most_rated_books['ISBN'] = most_rated_books['ISBN'].str.rjust(10,'0')

summary = pd.merge(most_rated_books, BX_BOOKS, on='ISBN')
summary
```

Out[13]:

	ISBN	Book-Title	Book-Author	Year-Of-Publication	Publisher
0	0971880107	Wild Animus	Rich Shapero	2004	Too Far
1	0316666343	The Lovely Bones: A Novel	Alice Sebold	2002	Little, Brown
2	0385504209	The Da Vinci Code	Dan Brown	2003	Doubleday
3	0060928336	Divine Secrets of the Ya-Ya Sisterhood: A Novel	Rebecca Wells	1997	Perennial
4	0312195516	The Red Tent (Bestselling Backlist)	Anita Diamant	1998	Picador USA

Grouping and Ranking Data

In [14]:
```python
# grouping the dta bases on the avergae rate of each book
rating = pd.DataFrame(BX_Book_Ratings.groupby('ISBN')['Book-Rating'].mean())
rating.head()
```

Out[14]:

ISBN	Book-Rating
904492401X	0.0
#069580216X	0.0
)452273056	8.0
)553267833	7.0
)959326839	0.0

In [15]:
```python
# counting the number of rating for each book along side the average rating
rating['Rating_count'] = pd.DataFrame(BX_Book_Ratings.groupby('ISBN')['Book-Rating'].count())
rating.head()
```

Out[15]:

ISBN	Book-Rating	Rating_count
904492401X	0.0	1
#069580216X	0.0	1
)452273056	8.0	1
)553267833	7.0	1
)959326839	0.0	1

In [16]:
```python
rating.describe()
```

Out[16]:

	Book-Rating	Rating_count
count	204217.000000	204217.000000
mean	3.142751	2.469530
std	3.514013	7.013863
min	0.000000	1.000000
25%	0.000000	1.000000
50%	2.000000	1.000000
75%	6.000000	2.000000
max	10.000000	1107.000000

In [17]:
```python
# sort based on the total rating
rating.sort_values('Rating_count', ascending=False).head()
```

Out[17]:

ISBN	Book-Rating	Rating_count
971880107	1.040650	1107
316666343	4.357271	557
385504209	4.628362	409
60928336	3.689441	322
312195516	4.418239	318

Approch 2 obtaining the books based on the correlation

In [6]:
```python
# finding the information of the books
Merged_data = pd.merge(BX_BOOKS, BX_Book_Ratings, on='ISBN')
Merged_data=Merged_data.drop(['Publisher','Year-Of-Publication'], axis=1)
Merged_data = Merged_data.drop_duplicates()

Merged_data.head()
```

Out[6]:

	ISBN	Book-Title	Book-Author	Image-URL-S	Image-URL-M	
0	074322678X	Where You'll Find Me: And Other Stories	Ann Beattie	http://images.amazon.com/images/P/074322678X.0...	http://images.amazon.com/images/P/074322678X.0...	http://images.amazon.com/imag
1	080652121X	Hitler's Secret Bankers: The Myth of Swiss Neu...	Adam Lebor	http://images.amazon.com/images/P/080652121X.0...	http://images.amazon.com/images/P/080652121X.0...	http://images.amazon.com/imag
2	1552041778	Jane Doe	R. J. Kaiser	http://images.amazon.com/images/P/1552041778.0...	http://images.amazon.com/images/P/1552041778.0...	http://images.amazon.com/imag
3	1558746218	A Second Chicken Soup for the Woman's Soul (Ch...	Jack Canfield	http://images.amazon.com/images/P/1558746218.0...	http://images.amazon.com/images/P/1558746218.0...	http://images.amazon.com/imag
4	1558746218	A Second Chicken Soup for the Woman's Soul (Ch...	Jack Canfield	http://images.amazon.com/images/P/1558746218.0...	http://images.amazon.com/images/P/1558746218.0...	http://images.amazon.com/imag

In [7]:
```python
len(Merged_data)
```

Out[7]: 88126

In [8]:
```python
#for the sake of test we use the first 50000 rows
Merged_data1=Merged_data.iloc[0:50000]
```

In [21]:
```python
# using crosstab to obtain the users rating for each book
Book_crosstab = pd.pivot_table(data=Merged_data1, values='Book-Rating', index='User-ID', columns='Book-Title')
Book_crosstab.head()
```

Out[21]:

Book-Title User-ID	Earth Prayers From around the World: 365 Prayers, Poems, and Invocations for Honoring the Earth	Final Fantasy Anthology: Official Strategy Guide (Brady Games)	Q-Zone (Star Trek The Next Generation, Book 48)	Tales of Terror and Suspense	$oft Money: The True Power in Our Nation's Capital	' Sie belieben wohl zu scherzen, Mr. Feynman.'. Abenteuer eines neugierigen Physikers.	' Small g'. Eine Sommeridylle.	'N Sync	'Salem's Lot	--Olivetti, Moulinex, Chaffoteaux et Maury (Quaderns crema. Narrativa)	...	de Parte de La Princesa Muerta	il Paradiso Degli Orchi	metamorfos La (ClÃ¡sic selecciÁ seri
8	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	N
10	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	N
12	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	N
22	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	N
32	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	N

5 rows × 13868 columns

In [22]:
```python
Book_crosstab=Book_crosstab.fillna(0)
```

In [23]:
```python
Book_crosstab.shape
```

Out[23]: (12526, 13868)

In [24]:
```python
X = Book_crosstab.T
X.shape
```

Out[24]: (13868, 12526)

In [25]:
```python
#X=X.iloc[0:20000]
```

In [26]:
```python
X.shape
```

Out[26]: (13868, 12526)

In [27]:
```python
# use SVD to transform the data
SVD = TruncatedSVD(n_components=12, random_state=17)

Books_matrix = SVD.fit_transform(X)

Books_matrix.shape
```

Out[27]: (13868, 12)

In [28]:
```python
corr_mat = np.corrcoef(Books_matrix)
corr_mat.shape
```

```
C:\Users\ahmoh\anaconda3\lib\site-packages\numpy\lib\function_base.py:2534: RuntimeWarning: invalid value encountered in true_divide
  c /= stddev[:, None]
C:\Users\ahmoh\anaconda3\lib\site-packages\numpy\lib\function_base.py:2535: RuntimeWarning: invalid value encountered in true_divide
  c /= stddev[None, :]
```

Out[28]: (13868, 13868)

In [29]:
```python
#Book_crosstab1=Book_crosstab.iloc[0:20000]
```

In [30]:
```python
# finding the index of a book
book_names = Book_crosstab.columns
book_list = list(book_names)

Book_Example = book_list.index('El Senor De Los Anillos: LA Comunidad Del Anillo (Lord of the Rings (Spanish))')
Book_Example
```

Out[30]: 3707

In [31]:
```python
corr_Book_Example= corr_mat[3707]
corr_Book_Example.shape
```

Out[31]: (13868,)

In [32]:
```python
# finding the closes book based on the correlation of .98

list(book_names[(corr_Book_Example<1.0) & (corr_Book_Example > 0.98)])
```

```
C:\Users\ahmoh\anaconda3\lib\site-packages\ipykernel_launcher.py:3: RuntimeWarning: invalid value encountered in less
  This is separate from the ipykernel package so we can avoid doing imports until
C:\Users\ahmoh\anaconda3\lib\site-packages\ipykernel_launcher.py:3: RuntimeWarning: invalid value encountered in greater
  This is separate from the ipykernel package so we can avoid doing imports until
```

Out[32]:
```
['Crossing Over',
 'Disneys Pocahontas (Classic)',
 'El Senor De Los Anillos: Las DOS Torres (Lord of the Rings (Paperback))',
 'El quinto jineto',
 'La caverna = A caverna',
 'Madame Bovary (Fabula)',
 'Sin Destino',
 'The Third Man and the Fallen Idol (Penguin Twentieth-Century Classics)',
 'Villette (Penguin Popular Classics)']
```

Content based recommendation system

In [9]:
```python
Merged_data.head()
```

Out[9]:

	ISBN	Book-Title	Book-Author	Image-URL-S	Image-URL-M	
0	074322678X	Where You'll Find Me: And Other Stories	Ann Beattie	http://images.amazon.com/images/P/074322678X.0...	http://images.amazon.com/images/P/074322678X.0...	http://images.amazon.com/imag
1	080652121X	Hitler's Secret Bankers: The Myth of Swiss Neu...	Adam Lebor	http://images.amazon.com/images/P/080652121X.0...	http://images.amazon.com/images/P/080652121X.0...	http://images.amazon.com/imag
2	1552041778	Jane Doe	R. J. Kaiser	http://images.amazon.com/images/P/1552041778.0...	http://images.amazon.com/images/P/1552041778.0...	http://images.amazon.com/imag
3	1558746218	A Second Chicken Soup for the Woman's Soul (Ch...	Jack Canfield	http://images.amazon.com/images/P/1558746218.0...	http://images.amazon.com/images/P/1558746218.0...	http://images.amazon.com/imag
4	1558746218	A Second Chicken Soup for the Woman's Soul (Ch...	Jack Canfield	http://images.amazon.com/images/P/1558746218.0...	http://images.amazon.com/images/P/1558746218.0...	http://images.amazon.com/imag

In [ ]:
```python
groups = Merged_data.groupby(['Book-Author','Book-Author']).size()
groups.plot.bar()
```

Out[ ]: <matplotlib.axes._subplots.AxesSubplot at 0x1ab034da948>

In [131]:
```python
# clean the data
Similarity_data = pd.merge(BX_BOOKS_new, BX_Book_Ratings, on='ISBN')

Similarity_data_val=Similarity_data.groupby(['ISBN','Book-Title','Book-Author','Year-Of-Publication','Publisher','Image-URL-S','Image-URL-M','Image-URL-L'])['Book-Rating'].mean().reset_index()

Similarity_data_new=Similarity_data.drop(['User-ID','Publisher','ISBN','Image-URL-S','Image-URL-M','Image-URL-L'], axis=1)

Similarity_data_new=Similarity_data_new.groupby(['Book-Title','Book-Author','Year-Of-Publication'])['Book-Rating'].mean().reset_index()

Similarity_data_new=Similarity_data_new.drop(['Book-Title','Book-Author'], axis=1)

Similarity_data_new.head()
```

Out[131]:

	Year-Of-Publication	Book-Rating
0	1991	8.666667
1	1999	5.000000
2	2002	8.000000
3	1997	0.000000
4	1994	0.000000

In [132]:
```python
# creating test set to test the closest book based on Year-Of-Publication and Book-Rating
test = [1995, 7.7]
X = Similarity_data_new.values

X[0:5]
```

Out[132]:
```
array([[1991.        ,    8.66666667],
       [1999.        ,    5.        ],
       [2002.        ,    8.        ],
       [1997.        ,    0.        ],
       [1994.        ,    0.        ]])
```

In [133]:
```python
# getting the nearstneighbor

nbrs = NearestNeighbors(n_neighbors=1).fit(X)
```

In [134]:
```python
# finding thr lovation of the closest book to the test set
print(nbrs.kneighbors([test]))
```

```
(array([[0.2]]), array([[27497]], dtype=int64))
```

In [136]:
```python
# the book infromation.
Similarity_data_val.iloc[27497]
```

Out[136]:
```
ISBN                                                  1904908004
Book-Title                                     Shakespeare My Butt
Book-Author                                          John Donoghue
Year-Of-Publication                                           2004
Publisher                                   Central Publishing Ltd
Image-URL-S            http://images.amazon.com/images/P/1904908004.0...
Image-URL-M            http://images.amazon.com/images/P/1904908004.0...
Image-URL-L            http://images.amazon.com/images/P/1904908004.0...
Book-Rating                                                      6
Name: 27497, dtype: object
```

In [ ]: