In [42]:

```
#importing the necessary libraries

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

IMPORTING THE DATASETS

In [3]:

```
dfbooks = pd.read_csv("BX-Books.csv" , encoding='Latin1')
dfbooks
```

```
C:\Users\Ankita Sharma\AppData\Local\Temp\ipykernel_26504\207741491.py:1: DtypeWarning: Columns (3) have mixed types. Speci
fy dtype option on import or set low_memory=False.
  dfbooks = pd.read_csv("BX-Books.csv" , encoding='Latin1')
```

Out[3]:

	isbn	book_title	book_author	year_of_publication	publisher
0	195153448	Classical Mythology	Mark P. O. Morford	2002	Oxford University Press
1	2005018	Clara Callan	Richard Bruce Wright	2001	HarperFlamingo Canada
2	60973129	Decision in Normandy	Carlo D'Este	1991	HarperPerennial
3	374157065	Flu: The Story of the Great Influenza Pandemic...	Gina Bari Kolata	1999	Farrar Straus Giroux
4	393045218	The Mummies of Urumchi	E. J. W. Barber	1999	W. W. Norton & Company
...	...	...	...	...	...
271374	440400988	There's a Bat in Bunk Five	Paula Danziger	1988	Random House Childrens Pub (Mm)
271375	525447644	From One to One Hundred	Teri Sloat	1991	Dutton Books
271376	006008667X	Lily Dale : The True Story of the Town that Ta...	Christine Wicker	2004	HarperSanFrancisco
271377	192126040	Republic (World's Classics)	Plato	1996	Oxford University Press
271378	767409752	A Guided Tour of Rene Descartes' Meditations o...	Christopher Biffle	2000	McGraw-Hill Humanities/Social Sciences/Languages

271379 rows × 5 columns

In [4]:

```
dfusers = pd.read_csv("BX-Users.csv" , encoding = 'Latin1')
dfusers
```

```
C:\Users\Ankita Sharma\AppData\Local\Temp\ipykernel_26504\72102068.py:1: DtypeWarning: Columns (0) have mixed types. Specif
y dtype option on import or set low_memory=False.
  dfusers = pd.read_csv("BX-Users.csv" , encoding = 'Latin1')
```

Out[4]:

	user_id	Location	Age
0	1	nyc, new york, usa	NaN
1	2	stockton, california, usa	18.0
2	3	moscow, yukon territory, russia	NaN
3	4	porto, v.n.gaia, portugal	17.0
4	5	farnborough, hants, united kingdom	NaN
...	...	...	...
278854	278854	portland, oregon, usa	NaN
278855	278855	tacoma, washington, united kingdom	50.0
278856	278856	brampton, ontario, canada	NaN
278857	278857	knoxville, tennessee, usa	NaN
278858	278858	dublin, n/a, ireland	NaN

278859 rows × 3 columns

In [5]:

```
dfratings = pd.read_csv("BX-Book-Ratings.csv" , encoding='Latin1')
dfratings
```

Out[5]:

	user_id	isbn	rating
0	276725	034545104X	0
1	276726	155061224	5
2	276727	446520802	0
3	276729	052165615X	3
4	276729	521795028	6
...	...	...	...
1048570	250764	451410777	0
1048571	250764	452264464	8
1048572	250764	048623715X	0
1048573	250764	486256588	0
1048574	250764	515069434	0

1048575 rows × 3 columns

#Checking for Null Values

In [44]:

```
dfratings.isna().sum()
```

Out[44]:

```
user_id    0
isbn       0
rating     0
dtype: int64
```

In [7]:

```
dfusers.isna().sum()
```

Out[7]:

```
user_id          0
Location         1
Age         110763
dtype: int64
```

In [8]:

```
dfbooks.isna().sum()
```

Out[8]:

```
isbn                 0
book_title           0
book_author          1
year_of_publication  0
publisher            2
dtype: int64
```

In [45]:

```
# making a new dataframe df1 merging our books and ratings dataframes on isbn.

df1 = dfratings.merge(dfbooks , on='isbn')
df1
```

Out[45]:

	user_id	isbn	rating	book_title	book_author	year_of_publication	publisher
0	276725	034545104X	0	Flesh Tones: A Novel	M. J. Rose	2002	Ballantine Books
1	2313	034545104X	5	Flesh Tones: A Novel	M. J. Rose	2002	Ballantine Books
2	6543	034545104X	0	Flesh Tones: A Novel	M. J. Rose	2002	Ballantine Books
3	8680	034545104X	5	Flesh Tones: A Novel	M. J. Rose	2002	Ballantine Books
4	10314	034545104X	9	Flesh Tones: A Novel	M. J. Rose	2002	Ballantine Books
...	...	...	...	...	...	...	...
941143	250764	440106575	0	HIDDEN FIRES	JANETTE RADCLIFFE	1978	Dell
941144	250764	451157516	0	Cheyenne (Fortunes West, No 2)	A.R. Riefe	1988	New Amer Library (Mm)
941145	250764	048623715X	0	Glamorous Movie Stars of the Thirties: Paper D...	Tom Tierney	1982	Dover Publications
941146	250764	486256588	0	Schiaparelli Fashion Review: Paper Dolls in Fu...	Tom Tierney	1988	Dover Publications
941147	250764	515069434	0	Lady Laughing Eyes (To Have and to Hold)	Lee Damon	1984	Jove Books

941148 rows × 7 columns

In [46]:

```
# instead of relying on the readerrs those who have read just a few books , we will rather trust our top readers(read>200 books).

topreaders = df1['user_id'].where(df1['user_id'].value_counts()>200).dropna()
topreaders
```

Out[46]:

```
254       135741.0
2276      250483.0
2766        5249.0
2977      138441.0
3363       12154.0
            ...
250405     75595.0
250764     20833.0
277427    200226.0
277639    167816.0
278418     47594.0
Name: user_id, Length: 737, dtype: float64
```

In [48]:

```
# creating new dataframe df2 from df1 that contains the data of our top readers only.

df2 = df1.where(df1['user_id'].isin(topreaders)).dropna()
df2
```

Out[48]:

	user_id	isbn	rating	book_title	book_author	year_of_publication	publisher
2	6543.0	034545104X	0.0	Flesh Tones: A Novel	M. J. Rose	2002	Ballantine Books
7	28523.0	034545104X	0.0	Flesh Tones: A Novel	M. J. Rose	2002	Ballantine Books
15	77940.0	034545104X	0.0	Flesh Tones: A Novel	M. J. Rose	2002	Ballantine Books
20	115435.0	034545104X	5.0	Flesh Tones: A Novel	M. J. Rose	2002	Ballantine Books
21	123981.0	034545104X	0.0	Flesh Tones: A Novel	M. J. Rose	2002	Ballantine Books
...	...	...	...	...	...	...	...
941041	250645.0	892818239	0.0	Teen Astrology: The Ultimate Guide to Making Y...	M. J. Abadie	2001	Bindu Books
941042	250645.0	921912986	8.0	Ottawa With the Kids	James Hale	1996	MacFarlane Walter & Ross
941043	250645.0	1561678155	0.0	There Is No Magic: But There Are Alternatives ...	Stephen Dubrofsky	2003	American Literary Press
941044	250645.0	1581823045	0.0	Small Change: The Secret Life of Penny Burford	J. Belinda Yandell	2002	Cumberland House Publishing
941045	250645.0	1876327219	10.0	I Still Hate Cats	Skip Morrow	2000	Ink Group

207609 rows × 7 columns

In [49]:

```
# let's have a look at our book_title column to analyse that how many times a book is read.

df2['book_title'].value_counts()
```

Out[49]:

```
Wild Animus                                        143
The Lovely Bones: A Novel                          132
Bridget Jones's Diary                              127
The Notebook                                       115
Divine Secrets of the Ya-Ya Sisterhood: A Novel    106
                                                  ...
The Landower Legacy                                  1
Goldwater                                            1
PRIZE STORIES 1988                                   1
The Miracle of Lourdes                               1
I Still Hate Cats                                    1
Name: book_title, Length: 90746, dtype: int64
```

In [50]:

```
# i think we recommending a book that is only read once is not a valuable recommendation.
# so we will filter those books which are read atleast more than 50 times.

df2['book_title'].value_counts().where(df2['book_title'].value_counts()>50).dropna()
```

Out[50]:

```
Wild Animus                                                  143.0
The Lovely Bones: A Novel                                    132.0
Bridget Jones's Diary                                        127.0
The Notebook                                                 115.0
Divine Secrets of the Ya-Ya Sisterhood: A Novel              106.0
                                                             ...
Watership Down                                                51.0
Vinegar Hill (Oprah's Book Club (Paperback))                  51.0
Here on Earth                                                 51.0
The Body Farm                                                 51.0
Hard Eight : A Stephanie Plum Novel (A Stephanie Plum Novel)  51.0
Name: book_title, Length: 107, dtype: float64
```

In [52]:

```
# top_books contains the titlles of those books that are read more than 50 times.

top_books = df2['book_title'].value_counts()>50
top_books[top_books].index
```

Out[52]:

```
Index(['Wild Animus', 'The Lovely Bones: A Novel', 'Bridget Jones's Diary',
       'The Notebook', 'Divine Secrets of the Ya-Ya Sisterhood: A Novel',
       'The Pelican Brief', 'The Nanny Diaries: A Novel',
       'Snow Falling on Cedars', 'The Chamber', 'The Secret Life of Bees',
       ...
       'The Hot Zone', 'Cradle and All', 'Black Notice', 'Outlander',
       'High Five (A Stephanie Plum Novel)', 'Watership Down',
       'Vinegar Hill (Oprah's Book Club (Paperback))', 'Here on Earth',
       'The Body Farm',
       'Hard Eight : A Stephanie Plum Novel (A Stephanie Plum Novel)'],
      dtype='object', length=107)
```

In [53]:

```
# we will create a seperate dataframe df3 from df2 , to avoid the confusion.
# df3 contains the filtered data of our top readers and top books.

df3 = df2.where(df2['book_title'].isin(top_books[top_books].index)).dropna()
df4 = df3.reset_index().drop('index' , axis=1)
df4
```

Out[53]:

	user_id	isbn	rating	book_title	book_author	year_of_publication	publisher
0	278418.0	446520802	0.0	The Notebook	Nicholas Sparks	1996	Warner Books
1	3363.0	446520802	0.0	The Notebook	Nicholas Sparks	1996	Warner Books
2	8253.0	446520802	10.0	The Notebook	Nicholas Sparks	1996	Warner Books
3	11676.0	446520802	10.0	The Notebook	Nicholas Sparks	1996	Warner Books
4	29855.0	446520802	0.0	The Notebook	Nicholas Sparks	1996	Warner Books
...	...	...	...	...	...	...	...
7528	193676.0	1590400356	0.0	The Kitchen God's Wife	Amy Tan	2002	Phoenix Audio
7529	196457.0	375404686	0.0	Snow Falling on Cedars	David Guterson	1998	Random House Audio
7530	245827.0	1931056331	0.0	The Bonesetter's Daughter	Amy Tan	2001	New Millennium Audio
7531	231857.0	074321644X	0.0	Dreamcatcher	Stephen King	2001	Scribner
7532	238120.0	60093102	0.0	Mystic River	Dennis Lehane	2002	Harpercollins

7533 rows × 7 columns

In [54]:

```
#now we will create a pivot table,that will represent book titles on rows index and columns name will be user_id of our topreaders.
# by doing this we can find the similarity relation between our top readers and ratings given by them for each of our top books.

pt = df6.pivot_table(values='rating' ,index='book_title' , columns='user_id')
pt
```

Out[54]:

user_id	254.0	383.0	651.0	741.0	1025.0	1167.0	1903.0	2276.0	2288.0	2411.0	...	250483.0	250645.0	276925.0	276964.0	277427.0	277639.0	277928.0	27818
book_title																			
1st to Die: A Novel	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
2nd Chance	NaN	NaN	NaN	NaN	NaN	NaN	NaN	10.0	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	0.0	NaN	N
A Bend in the Road	0.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
A Is for Alibi (Kinsey Millhone Mysteries (Paperback))	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
A Map of the World	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N

In [55]:

```
# as there are lots of NaN values ,  so we will fill them with 0 , so that each book can have a proper vector.

pt.fillna(0, inplace=True)
```

In [56]:

```
pt
```

Out[56]:

user_id	254.0	383.0	651.0	741.0	1025.0	1167.0	1903.0	2276.0	2288.0	2411.0	...	250483.0	250645.0	276925.0	276964.0	277427.0	277639.0	27
book_title																		
1st to Die: A Novel	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	0.0	
2nd Chance	0.0	0.0	0.0	0.0	0.0	0.0	0.0	10.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	
A Bend in the Road	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	0.0	
A Is for Alibi (Kinsey Millhone Mysteries (Paperback))	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	0.0	
A Map of the World	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	0.0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
Where the Heart Is (Oprah's Book Club (Paperback))	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	0.0	
While I Was Gone	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	0.0	
Whispers	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	0.0	
White Oleander : A Novel	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	0.0	
Wild Animus	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	

107 rows × 508 columns

In [57]:

```
# importing cosine_similarity from sklearn to find similarity between the vectors that are created above.

from sklearn.metrics.pairwise import cosine_similarity
```

In [58]:

```
similarityscore = cosine_similarity(pt)
similarityscore
```

Out[58]:

```
array([[1.        , 0.14048103, 0.11667185, ..., 0.09613903, 0.12928713,
        0.12430633],
       [0.14048103, 1.        , 0.05944963, ..., 0.07713825, 0.07762564,
        0.16063165],
       [0.11667185, 0.05944963, 1.        , ..., 0.103078  , 0.18295968,
        0.15775027],
       ...,
       [0.09613903, 0.07713825, 0.103078  , ..., 1.        , 0.08141515,
        0.05306706],
       [0.12928713, 0.07762564, 0.18295968, ..., 0.08141515, 1.        ,
        0.10934768],
       [0.12430633, 0.16063165, 0.15775027, ..., 0.05306706, 0.10934768,
        1.        ]])
```

In [59]:

```
#Now, it's the time to create our function that can recommend us the books.
#we will use our similarity score to and fetch top 5 books that have max similarity score.

def recommender(book_name):
    ind = np.where(pt.index==book_name)[0][0]
    similaritems = sorted(list(enumerate(similarityscore[ind])) ,key= lambda x:x[1] , reverse=True )[1:6]
    for i in similaritems:
        print(pt.index[i[0]])
```

In [60]:

```
# and our recommender is ready , lets have a look at our model's results.

recommender('The Notebook')
```

```
A Walk to Remember
Message in a Bottle
The Bean Trees
The Street Lawyer
Circle of Friends
```

In [ ]:

```

```