In [1]:

```
import numpy as np
import pandas as pd
```

Importing data files

In [2]:

```
books = pd.read_csv('books.csv')
users = pd.read_csv('users.csv')
ratings = pd.read_csv('ratings.csv')
```

```
C:\Users\Varun kumar\anaconda3\lib\site-packages\IPython\core\interactiveshell.py:31
65: DtypeWarning: Columns (3) have mixed types.Specify dtype option on import or set
low_memory=False.
  has_raised = await self.run_ast_nodes(code_ast.body, cell_name,
```

In [3]:

```
users.head()
```

Out[3]:

	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

In [4]:

```
print(books.shape)
print(ratings.shape)
print(users.shape)
```

```
(271360, 8)
(1149780, 3)
(278858, 3)
```

In [ ]:

Total number of books: 271360 Total number of rating sgiven by users: 11.4 Lakhs

In [5]:

```
##checking null values in data sets
```

In [6]:

```
books.isnull().sum()
```

Out[6]:

```
ISBN                   0
Book-Title             0
Book-Author            1
Year-Of-Publication    0
Publisher              2
Image-URL-S            0
Image-URL-M            0
Image-URL-L            3
dtype: int64
```

In [7]:

```
users.isnull().sum()
```

Out[7]:
```
User-ID          0
Location         0
Age         110762
dtype: int64
```

In [8]:
```
ratings.isnull().sum()
```

Out[8]:
```
User-ID        0
ISBN           0
Book-Rating    0
dtype: int64
```

In [9]:
```
books.duplicated().sum()
```

Out[9]:
```
0
```

In [10]:
```
ratings.duplicated().sum()
```

Out[10]:
```
0
```

In [11]:
```
users.duplicated().sum()
```

Out[11]:
```
0
```

Conclusion : Ages values of users are missing so we are taking this faeture into consideration

Popularity Based Recommender System

In [12]:
```
ratings_with_name = ratings.merge(books,on='ISBN') #merging two dataframes on basis
ratings_with_name
```

Out[12]:

	User-ID	ISBN	Book-Rating	Book-Title	Book-Author	Year-Of-Publication	Publisher	
0	276725	034545104X	0	Flesh Tones: A Novel	M. J. Rose	2002	Ballantine Books	http://ima
1	2313	034545104X	5	Flesh Tones: A Novel	M. J. Rose	2002	Ballantine Books	http://ima
2	6543	034545104X	0	Flesh Tones: A Novel	M. J. Rose	2002	Ballantine Books	http://ima
3	8680	034545104X	5	Flesh Tones: A Novel	M. J. Rose	2002	Ballantine Books	http://ima
4	10314	034545104X	9	Flesh Tones: A Novel	M. J. Rose	2002	Ballantine Books	http://ima
...	...	...	...	...	...	...	...	
1031131	276688	0517145553	0	Mostly Harmless	Douglas Adams	1995	Random House Value Pub	http://ima
1031132	276688	1575660792	7	Gray Matter	Shirley Kennett	1996	Kensington Publishing Corporation	http://ima

	User-ID	ISBN	Book-Rating	Book-Title	Book-Author	Year-Of-Publication	Publisher	
1031133	276690	0590907301	0	Triplet Trouble and the Class Trip (Triplet Tr...	Debbie Dadey	1997	Apple	http://ima
1031134	276704	0679752714	0	A Desert of Pure Feeling (Vintage Contemporaries)	Judith Freeman	1997	Vintage Books USA	http://ima
1031135	276704	0806917695	5	Perplexing Lateral Thinking Puzzles: Scholasti...	Paul Sloane	1997	Sterling Publishing	http://ima

1031136 rows × 10 columns

In [13]:

```
num_rating_df = ratings_with_name.groupby('Book-Title').count()['Book-Rating'].reset
num_rating_df.rename(columns={'Book-Rating':'num_ratings'},inplace=True)
num_rating_df
```

Out[13]:

	Book-Title	num_ratings
0	A Light in the Storm: The Civil War Diary of ...	4
1	Always Have Popsicles	1
2	Apple Magic (The Collector's series)	1
3	Ask Lily (Young Women of Faith: Lily Series, ...	1
4	Beyond IBM: Leadership Marketing and Finance ...	1
...	...	...
241066	Ã?Â?lpiraten.	2
241067	Ã?Â?rger mit Produkt X. Roman.	4
241068	Ã?Â?sterlich leben.	1
241069	Ã?Â?stlich der Berge.	3
241070	Ã?Â?thique en toc	2

241071 rows × 2 columns

In [14]:

```
avg_rating_df = ratings_with_name.groupby('Book-Title').mean()['Book-Rating'].reset_
avg_rating_df.rename(columns={'Book-Rating':'avg_rating'},inplace=True)
avg_rating_df
```

Out[14]:

	Book-Title	avg_rating
0	A Light in the Storm: The Civil War Diary of ...	2.250000
1	Always Have Popsicles	0.000000
2	Apple Magic (The Collector's series)	0.000000
3	Ask Lily (Young Women of Faith: Lily Series, ...	8.000000

	Book-Title	avg_rating
4	Beyond IBM: Leadership Marketing and Finance ...	0.000000
...	...	...
241066	Ã?Â?lpiraten.	0.000000
241067	Ã?Â?rger mit Produkt X. Roman.	5.250000
241068	Ã?Â?sterlich leben.	7.000000
241069	Ã?Â?stlich der Berge.	2.666667
241070	Ã?Â?thique en toc	4.000000

241071 rows × 2 columns

In [15]:

```
popular_df = num_rating_df.merge(avg_rating_df,on='Book-Title')
popular_df
```

Out[15]:

	Book-Title	num_ratings	avg_rating
0	A Light in the Storm: The Civil War Diary of ...	4	2.250000
1	Always Have Popsicles	1	0.000000
2	Apple Magic (The Collector's series)	1	0.000000
3	Ask Lily (Young Women of Faith: Lily Series, ...	1	8.000000
4	Beyond IBM: Leadership Marketing and Finance ...	1	0.000000
...	...	...	...
241066	Ã?Â?lpiraten.	2	0.000000
241067	Ã?Â?rger mit Produkt X. Roman.	4	5.250000
241068	Ã?Â?sterlich leben.	1	7.000000
241069	Ã?Â?stlich der Berge.	3	2.666667
241070	Ã?Â?thique en toc	2	4.000000

241071 rows × 3 columns

In [16]:

```
popular_df = popular_df[popular_df['num_ratings']>=200].sort_values('avg_rating',asc
popular_df.head()
```

Out[16]:

	Book-Title	num_ratings	avg_rating
80434	Harry Potter and the Prisoner of Azkaban (Book 3)	428	5.852804
80422	Harry Potter and the Goblet of Fire (Book 4)	387	5.824289
80441	Harry Potter and the Sorcerer's Stone (Book 1)	278	5.737410
80426	Harry Potter and the Order of the Phoenix (Boo...	347	5.501441
60582	Ender's Game (Ender Wiggins Saga (Paperback))	249	5.409639

In [17]:

```
#popular_df = popular_df.merge(books,on='Book-Title').drop_duplicates('Book-Title')[
```

In [ ]:

```

```

Collaborative Filtering Based Recommender System

In [18]:

```
x = ratings_with_name.groupby('User-ID').count()['Book-Rating'] > 200 #users who use
users_rated = x[x].index  #index of users who rated
```

In [19]:

```
filtered_rating = ratings_with_name[ratings_with_name['User-ID'].isin(users_rated )]
```

In [20]:

```
y = filtered_rating.groupby('Book-Title').count()['Book-Rating']>=50
famous_books = y[y].index
```

In [21]:

```
#final ratings
final_ratings = filtered_rating[filtered_rating['Book-Title'].isin(famous_books)]
final_ratings=final_ratings.drop(["Image-URL-S","Image-URL-M","Publisher","Image-URL
final_ratings
```

Out[21]:

	User-ID	ISBN	Book-Rating	Book-Title	Book-Author
63	278418	0446520802	0	The Notebook	Nicholas Sparks
65	3363	0446520802	0	The Notebook	Nicholas Sparks
66	7158	0446520802	10	The Notebook	Nicholas Sparks
69	11676	0446520802	10	The Notebook	Nicholas Sparks
74	23768	0446520802	6	The Notebook	Nicholas Sparks
...	...	...	...	...	...
1026724	266865	0531001725	10	The Catcher in the Rye	Jerome David Salinger
1027923	269566	0670809381	0	Echoes	Maeve Binchy
1028777	271284	0440910927	0	The Rainmaker	John Grisham
1029070	271705	B0001PIOX4	0	Fahrenheit 451	Ray Bradbury
1030868	275970	1586210661	9	Me Talk Pretty One Day	David Sedaris

58586 rows × 5 columns

In [22]:

```
pt = final_ratings.pivot_table(index='Book-Title',columns='User-ID',values='Book-Rat
```

In [23]:

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

In [24]:

```
pt
```

Out[24]:

User-ID	254	2276	2766	2977	3363	4017	4385	6251	6323	6543	...	271705	273979
Book-Title													
1984	9.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	10.0	0.0

User-ID	254	2276	2766	2977	3363	4017	4385	6251	6323	6543	...	271705	273979
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	9.0	...	0.0	0.0
2nd Chance	0.0	10.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0
4 Blondes	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 Bend in the Road	0.0	0.0	7.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
Year of Wonders	0.0	0.0	0.0	7.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	9.0
You Belong To Me	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0
Zen and the Art of Motorcycle Maintenance: An Inquiry into Values	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0
Zoya	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0
\O\" Is for Outlaw"	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0

706 rows × 810 columns

In [25]:

```
from sklearn.metrics.pairwise import cosine_similarity
```

In [26]:

```
similarity_scores = cosine_similarity(pt)
```

In [27]:

```
similarity_scores.shape
similarity_scores
```

Out[27]:

```
array([[1.        , 0.10255025, 0.01220856, ..., 0.12110367, 0.07347567,
        0.04316046],
       [0.10255025, 1.        , 0.2364573 , ..., 0.07446129, 0.16773875,
        0.14263397],
       [0.01220856, 0.2364573 , 1.        , ..., 0.04558758, 0.04938579,
        0.10796119],
       ...,
       [0.12110367, 0.07446129, 0.04558758, ..., 1.        , 0.07085128,
        0.0196177 ],
       [0.07347567, 0.16773875, 0.04938579, ..., 0.07085128, 1.        ,
        0.10602962],
       [0.04316046, 0.14263397, 0.10796119, ..., 0.0196177 , 0.10602962,
        1.        ]])
```

Top 10 book for user input

In [47]:

```
def recommend(book_name):
    # index fetch

    index = np.where(pt.index==book_name)[0][0]
    similar_items = sorted(list(enumerate(similarity_scores[index])),key=lambda x:x[

    data = []
    for i in similar_items:
        item = []
        temp_df = books[books['Book-Title'] == pt.index[i[0]]]
        item.extend(list(temp_df.drop_duplicates('Book-Title')['Book-Title'].values)
        item.extend(list(temp_df.drop_duplicates('Book-Title')['Book-Author'].values
        #item.extend(list(temp_df.drop_duplicates('Book-Title')['Image-URL-M'].value

        data.append(item)

    return data
```

Top Ten book realted to '1984' book based on CF

In [48]:

```
recommend('1984')
```

Out[48]:

```
[['Animal Farm', 'George Orwell'],
 ["The Handmaid's Tale", 'Margaret Atwood'],
 ['Brave New World', 'Aldous Huxley'],
 ['The Vampire Lestat (Vampire Chronicles, Book II)', 'ANNE RICE'],
 ['The Hours : A Novel', 'Michael Cunningham'],
 ['Fahrenheit 451', 'Ray Bradbury'],
 ['The Catcher in the Rye', 'J.D. Salinger'],
 ['Naked', 'David Sedaris'],
 ['The Hundred Secret Senses', 'Amy Tan']]
```

Enter Book Name:

In [49]:

```
book=input()
recommend(book)
```

```
4 Blondes
```

Out[49]:

```
[['The House of the Spirits', 'Isabel Allende'],
 ['Pride and Prejudice', 'Jane Austen'],
 ['Pleading Guilty', 'Scott Turow'],
 ['Seabiscuit', 'LAURA HILLENBRAND'],
 ['Notes from a Small Island', 'Bill Bryson'],
 ['Bridget Jones: The Edge of Reason', 'Helen Fielding'],
 ["Schindler's List", 'Thomas Keneally'],
 ['The Nanny Diaries: A Novel', 'Emma McLaughlin'],
 ['Here on Earth', 'Alice Hoffman']]
```

In [ ]: