BOOK RECOMMENDATION SYSTEM

Problem Statement

During the last few decades, with the rise of Youtube, Amazon, Netflix, and many other such web services, recommender systems have taken more and more place in our lives. From e-commerce (suggest to buyers articles that could interest them) to online advertisement (suggest to users the right contents, matching their preferences), recommender systems are today unavoidable in our daily online journeys. In a very general way, recommender systems are algorithms aimed at suggesting relevant items to users (items being movies to watch, text to read, products to buy, or anything else depending on industries).

types of recommender system

1. Popularity Based (formula based)
2. Content Based (based on attr , find related)
3. Collaborative filtering based (like k neighbour , based on user rating)
4. Hybrid (mixture of all this)

Dataset:

Books.csv = Books are identified by their respective ISBN. Invalid ISBNs have already been removed from the dataset. Moreover, some content-based information is given (Book-Title, Book-Author, Year-Of-Publication, Publisher), obtained from Amazon Web Services. Note that in case of several authors, only the first is provided. URLs linking to cover images are also given, appearing in three different flavours (Image-URL-S, Image-URL-M, Image-URL-L), i.e., small, medium, large. These URLs point to the Amazon web site.

Ratings.csv = Contains the book rating information. Ratings (Book-Rating) are either explicit, expressed on a scale from 1-10 (higher values denoting higher appreciation), or implicit, expressed by 0.

Users.csv = Contains the users. Note that user IDs (User-ID) have been anonymized and map to integers. Demographic data is provided (Location, Age) if available. Otherwise, these fields contain NULL-values.

In [1]:
```
import pandas as pd
import numpy as np
```

In [2]:
```
books = pd.read_csv('books.csv')
users = pd.read_csv('users.csv')
ratings = pd.read_csv('ratings.csv')
```

```
C:\Users\Lenovo\AppData\Local\Temp\ipykernel_12488\3819754567.py:1: DtypeWarning: Columns (3) have mixed types. Specify dtype option on imp
ort or set low_memory=False.
  books = pd.read_csv('books.csv')
```

In [3]:
```
books.head()
```

Out[3]:

	ISBN	Book-Title	Book-Author	Year-Of-Publication	Publisher	Image-URL-S	Image-URL-M	
0	0195153448	Classical Mythology	Mark P. O. Morford	2002	Oxford University Press	http://images.amazon.com/images/P/0195153448.0...	http://images.amazon.com/images/P/0195153448.0...	ht
1	0002005018	Clara Callan	Richard Bruce Wright	2001	HarperFlamingo Canada	http://images.amazon.com/images/P/0002005018.0...	http://images.amazon.com/images/P/0002005018.0...	ht
2	0060973129	Decision in Normandy	Carlo D'Este	1991	HarperPerennial	http://images.amazon.com/images/P/0060973129.0...	http://images.amazon.com/images/P/0060973129.0...	ht
3	0374157065	Flu: The Story of the Great Influenza Pandemic...	Gina Bari Kolata	1999	Farrar Straus Giroux	http://images.amazon.com/images/P/0374157065.0...	http://images.amazon.com/images/P/0374157065.0...	ht
4	0393045218	The Mummies of Urumchi	E. J. W. Barber	1999	W. W. Norton & Company	http://images.amazon.com/images/P/0393045218.0...	http://images.amazon.com/images/P/0393045218.0...	ht

In [4]:
```
users.head()
```

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

In [5]:
```
ratings.head()
```

Out[5]:

	User-ID	ISBN	Book-Rating
0	276725	034545104X	0
1	276726	0155061224	5
2	276727	0446520802	0
3	276729	052165615X	3
4	276729	0521795028	6

In [6]:
```
print(books.shape)
print(users.shape)
print(ratings.shape)
```

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

In [7]:
```
books.isnull().sum()
```

Out[7]:
```
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 [8]:
```
users.isnull().sum()
```

Out[8]:
```
User-ID          0
Location         0
Age         110762
dtype: int64
```

In [9]:
```
ratings.isnull().sum()
```

Out[9]:
```
User-ID        0
ISBN           0
Book-Rating    0
dtype: int64
```

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

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

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

Out[11]:
```
User-ID          0
Location         0
Age         110762
dtype: int64
```

In [12]:
```
ratings.isnull().sum()
```

Out[12]:
```
User-ID        0
ISBN           0
Book-Rating    0
dtype: int64
```

Popularity based Recommender System

we will displaying 50 popular books based highest average rating which having min 250 ratings

In [13]:
```
rating_with_naame=pd.merge(ratings,books,on='ISBN',how='inner')
rating_with_naame.head()
```

Out[13]:

	User-ID	ISBN	Book-Rating	Book-Title	Book-Author	Year-Of-Publication	Publisher	Image-URL-S	Image-URL
0	276725	034545104X	0	Flesh Tones: A Novel	M. J. Rose	2002	Ballantine Books	http://images.amazon.com/images/P/034545104X.0...	http://images.amazon.com/images/P/034545104X.
1	2313	034545104X	5	Flesh Tones: A Novel	M. J. Rose	2002	Ballantine Books	http://images.amazon.com/images/P/034545104X.0...	http://images.amazon.com/images/P/034545104X.
2	6543	034545104X	0	Flesh Tones: A Novel	M. J. Rose	2002	Ballantine Books	http://images.amazon.com/images/P/034545104X.0...	http://images.amazon.com/images/P/034545104X.
3	8680	034545104X	5	Flesh Tones: A Novel	M. J. Rose	2002	Ballantine Books	http://images.amazon.com/images/P/034545104X.0...	http://images.amazon.com/images/P/034545104X.
4	10314	034545104X	9	Flesh Tones: A Novel	M. J. Rose	2002	Ballantine Books	http://images.amazon.com/images/P/034545104X.0...	http://images.amazon.com/images/P/034545104X.

In [14]:

```
rating_with_naame.groupby('Book-Title').count()
#on book title insted of ISBN because of , some books are liste with same namebut diff ISBN , therefore we want only book
```

Out[14]:

Book-Title	User-ID	ISBN	Book-Rating	Book-Author	Year-Of-Publication	Publisher	Image-URL-S	Image-URL-M	Image-URL-L
A Light in the Storm: The Civil War Diary of Amelia Martin, Fenwick Island, Delaware, 1861 (Dear America)	4	4	4	4	4	4	4	4	4
Always Have Popsicles	1	1	1	1	1	1	1	1	1
Apple Magic (The Collector's series)	1	1	1	1	1	1	1	1	1
Ask Lily (Young Women of Faith: Lily Series, Book 5)	1	1	1	1	1	1	1	1	1
Beyond IBM: Leadership Marketing and Finance for the 1990s	1	1	1	1	1	1	1	1	1
...	...	...	...	...	...	...	...	...	...
Ã?Â?lpiraten.	2	2	2	2	2	2	2	2	2
Ã?Â?rger mit Produkt X. Roman.	4	4	4	4	4	4	4	4	4
Ã?Â?sterlich leben.	1	1	1	1	1	1	1	1	1
Ã?Â?stlich der Berge.	3	3	3	3	3	3	3	3	3
Ã?Â?thique en toc	2	2	2	2	2	2	2	2	2

241071 rows × 9 columns

In [15]:

```
rating_with_naame.groupby('Book-Title').count()['Book-Rating']
```

Out[15]:

```
Book-Title
 A Light in the Storm: The Civil War Diary of Amelia Martin, Fenwick Island, Delaware, 1861 (Dear America)    4
 Always Have Popsicles                                                                                        1
 Apple Magic (The Collector's series)                                                                         1
 Ask Lily (Young Women of Faith: Lily Series, Book 5)                                                         1
 Beyond IBM: Leadership Marketing and Finance for the 1990s                                                   1
                                                                                                             ..
Ã?Â?lpiraten.                                                                                                 2
Ã?Â?rger mit Produkt X. Roman.                                                                                4
Ã?Â?sterlich leben.                                                                                           1
Ã?Â?stlich der Berge.                                                                                         3
Ã?Â?thique en toc                                                                                             2
Name: Book-Rating, Length: 241071, dtype: int64
```

In [16]:

```
rating_with_naame.groupby('Book-Title').count()['Book-Rating'].reset_index()
```

Out[16]:

	Book-Title	Book-Rating
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 [17]:

```
num_rating_df=rating_with_naame.groupby('Book-Title').count()['Book-Rating'].reset_index()
num_rating_df.rename(columns={'Book-Rating':'num_rating_on _book'},inplace=True)
num_rating_df
```

Out[17]:

	Book-Title	num_rating_on _book
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 [18]:

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

Out[18]:

	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
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 [19]:

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

Out[19]:

	Book-Title	num_rating_on _book	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 [20]:

```
popular_df=popular_df[popular_df['num_rating_on _book']>=250].sort_values(by='avg_rating',ascending=False)
popular_df
```

Out[20]:

	Book-Title	num_rating_on _book	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
80414	Harry Potter and the Chamber of Secrets (Book 2)	556	5.183453
...	...	...	...
227701	Vinegar Hill (Oprah's Book Club (Paperback))	265	2.245283
233635	Whispers	286	2.199301
143377	Presumed Innocent	294	2.139456
94382	Isle of Dogs	288	2.000000
234951	Wild Animus	2502	1.019584

186 rows × 3 columns

In [21]:

```
#popular_df.merge(books,on='Book-Title')
a=pd.merge(popular_df,books,on='Book-Title',how='inner').drop_duplicates('Book-Title')
a.head()
```

Out[21]:

	Book-Title	num_rating_on _book	avg_rating	ISBN	Book-Author	Year-Of-Publication	Publisher	Image-URL-S	
0	Harry Potter and the Prisoner of Azkaban (Book 3)	428	5.852804	0439136350	J. K. Rowling	1999	Scholastic	http://images.amazon.com/images/P/0439136350.0...	http://images.amazon.com/imag
3	Harry Potter and the Goblet of Fire (Book 4)	387	5.824289	0439139597	J. K. Rowling	2000	Scholastic	http://images.amazon.com/images/P/0439139597.0...	http://images.amazon.com/imag
5	Harry Potter and the Sorcerer's Stone (Book 1)	278	5.737410	0590353403	J. K. Rowling	1998	Scholastic	http://images.amazon.com/images/P/0590353403.0...	http://images.amazon.com/imag
9	Harry Potter and the Order of the Phoenix (Boo...	347	5.501441	043935806X	J. K. Rowling	2003	Scholastic	http://images.amazon.com/images/P/043935806X.0...	http://images.amazon.com/imag
13	Harry Potter and the Chamber of Secrets (Book 2)	556	5.183453	0439064872	J. K. Rowling	2000	Scholastic	http://images.amazon.com/images/P/0439064872.0...	http://images.amazon.com/imag

In [22]:
```
a.shape
```

Out[22]:
```
(186, 10)
```

In [23]:
```
popular_df_final=a[['Book-Title','Book-Author','Image-URL-M','num_rating_on _book','avg_rating','Year-Of-Publication']]
popular_df_final.head()
```

Out[23]:

	Book-Title	Book-Author	Image-URL-M	num_rating_on _book	avg_rating	Year-Of-Publication
0	Harry Potter and the Prisoner of Azkaban (Book 3)	J. K. Rowling	http://images.amazon.com/images/P/0439136350.0...	428	5.852804	1999
3	Harry Potter and the Goblet of Fire (Book 4)	J. K. Rowling	http://images.amazon.com/images/P/0439139597.0...	387	5.824289	2000
5	Harry Potter and the Sorcerer's Stone (Book 1)	J. K. Rowling	http://images.amazon.com/images/P/0590353403.0...	278	5.737410	1998
9	Harry Potter and the Order of the Phoenix (Boo...	J. K. Rowling	http://images.amazon.com/images/P/043935806X.0...	347	5.501441	2003
13	Harry Potter and the Chamber of Secrets (Book 2)	J. K. Rowling	http://images.amazon.com/images/P/0439064872.0...	556	5.183453	2000

In [24]:
```
popular_df_final.shape
#186 popular books
```

Out[24]:
```
(186, 6)
```

In [25]:
```
import random
```

In [26]:
```
# random_sample = random_sample.head(0)
random_sample = popular_df_final.sample(n=5, random_state= random.randint(1, 100))
# Assuming you have the following lists
a = list(random_sample['Book-Title'])
b = list(random_sample['Image-URL-M'])
c = list(random_sample['Book-Author'])
```

In [27]:
```
a
```

Out[27]:
```
['Wild Animus',
 'Midnight in the Garden of Good and Evil: A Savannah Story',
 'The Nanny Diaries: A Novel',
 'Silence of the Lambs',
 'The God of Small Things']
```

In [28]:
```
b
```

Out[28]:
```
['http://images.amazon.com/images/P/0971880107.01.MZZZZZZZ.jpg',
 'http://images.amazon.com/images/P/0679429220.01.MZZZZZZZ.jpg',
 'http://images.amazon.com/images/P/0312278586.01.MZZZZZZZ.jpg',
 'http://images.amazon.com/images/P/0312924585.01.MZZZZZZZ.jpg',
 'http://images.amazon.com/images/P/0060977493.01.MZZZZZZZ.jpg']
```

In [29]:
```
c
```

Out[29]:
```
['Rich Shapero',
 'John Berendt',
 'Emma McLaughlin',
 'Thomas Harris',
 'Arundhati Roy']
```

Content Based

In [30]:
```
a=input("enter a movie name")
```

```
enter a movie name1984
```

In [31]:
```
#books.drop_duplicates('Book-Author')
books[books['Book-Title']==a].head(1)
```

Out[31]:

	ISBN	Book-Title	Book-Author	Year-Of-Publication	Publisher	Image-URL-S	Image-URL-M	
2104	0451524934	1984	George Orwell	1990	Signet Book	http://images.amazon.com/images/P/0451524934.0...	http://images.amazon.com/images/P/0451524934.0...	http://ima

In [32]:
```
author=books[books['Book-Title']==a]['Book-Author'].head(1)
year=books[books['Book-Title']==a]['Year-Of-Publication'].head(1)
Publisher=books[books['Book-Title']==a]['Publisher'].head(1)
print(author,year,Publisher)
```

```
2104    George Orwell
Name: Book-Author, dtype: object 2104    1990
Name: Year-Of-Publication, dtype: object 2104    Signet Book
Name: Publisher, dtype: object
```

In [33]:
```
books[books['Book-Author'].isin(author)].head(1)
```

Out[33]:

	ISBN	Book-Title	Book-Author	Year-Of-Publication	Publisher	Image-URL-S	Image-URL-M	
285	0451526341	Animal Farm	George Orwell	2004	Signet	http://images.amazon.com/images/P/0451526341.0...	http://images.amazon.com/images/P/0451526341.0...	http://imag

In [34]:
```
books[books['Year-Of-Publication'].isin(year)].head(1)
```

Out[34]:

	ISBN	Book-Title	Book-Author	Year-Of-Publication	Publisher	Image-URL-S	Image-URL-M	
82	087113375X	Modern Manners: An Etiquette Book for Rude People	P.J. O'Rourke	1990	Atlantic Monthly Press	http://images.amazon.com/images/P/087113375X.0...	http://images.amazon.com/images/P/087113375X.0...	http://ir

In [35]:
```
books[books['Publisher'].isin(Publisher)].head(1)
```

Out[35]:

	ISBN	Book-Title	Book-Author	Year-Of-Publication	Publisher	Image-URL-S	Image-URL-M	
75	0451625889	The Prince	Niccolo Machiavelli	1952	Signet Book	http://images.amazon.com/images/P/0451625889.0...	http://images.amazon.com/images/P/0451625889.0...	http://im

In [36]:
```
books[books['Year-Of-Publication'].isin(year)].tail(1)
```

Out[36]:

	ISBN	Book-Title	Book-Author	Year-Of-Publication	Publisher	Image-URL-S	Image-URL-M
271330	3320016822	Urteil ohne Prozess: Margot Honecker gegen Oss...	JÃ¶rn Kalkbrenner	1990	Dietz	http://images.amazon.com/images/P/3320016822.0...	http://images.amazon.com/images/P/3320016822.0...

In [37]:
```
books[books['Book-Author'].isin(author)].tail(1)
```

Out[37]:

	ISBN	Book-Title	Book-Author	Year-Of-Publication	Publisher	Image-URL-S	Image-URL-M	
269379	0582060184	1984	George Orwell	1991	Longman Publishing Group	http://images.amazon.com/images/P/0582060184.0...	http://images.amazon.com/images/P/0582060184.0...	http://i

Collaborative Based

considering only users who have rated 100 books , and selecting those books who have 100 books book.csv + ratings.csv = rating_with_naame , rating_with_naame from this filter those user who write 100 ratings min and make it "filtered_user", filtered_user from this filter those books which have 100 ratings min and make it "filterd_book" df, filterd_book with their users is "final rating" df

In [38]:
```
b=rating_with_naame.groupby('User-ID').count()['Book-Rating']>100
b[b]
```

Out[38]:
```
User-ID
254       True
507       True
882       True
1424      True
1435      True
          ...
277478    True
277639    True
278137    True
278188    True
278418    True
Name: Book-Rating, Length: 1648, dtype: bool
```

In [39]:
```
exp_users=b[b].index
```

In [40]:
```
filtered_user=rating_with_naame[rating_with_naame['User-ID'].isin(exp_users)]
#this are experienced users , who have give min 100 book rating
filtered_user
```

Out[40]:

	User-ID	ISBN	Book-Rating	Book-Title	Book-Author	Year-Of-Publication	Publisher	Image-URL-S	
2	6543	034545104X	0	Flesh Tones: A Novel	M. J. Rose	2002	Ballantine Books	http://images.amazon.com/images/P/034545104X.0...	http://images.amazon.com/imag
4	10314	034545104X	9	Flesh Tones: A Novel	M. J. Rose	2002	Ballantine Books	http://images.amazon.com/images/P/034545104X.0...	http://images.amazon.com/imag
5	23768	034545104X	0	Flesh Tones: A Novel	M. J. Rose	2002	Ballantine Books	http://images.amazon.com/images/P/034545104X.0...	http://images.amazon.com/imag
7	28523	034545104X	0	Flesh Tones: A Novel	M. J. Rose	2002	Ballantine Books	http://images.amazon.com/images/P/034545104X.0...	http://images.amazon.com/imag
14	77480	034545104X	8	Flesh Tones: A Novel	M. J. Rose	2002	Ballantine Books	http://images.amazon.com/images/P/034545104X.0...	http://images.amazon.com/imag
...	...	...	...	...	...	...	...	...	
1031115	276680	0812881311	0	Streets of Gold	Evan Hunter	1985	Madison Books	http://images.amazon.com/images/P/0812881311.0...	http://images.amazon.com/imag
1031116	276680	0865055041	0	Kitchen (Historic Communities (Paperback))	Bobbie Kalman	1990	Crabtree Publishing Company	http://images.amazon.com/images/P/0865055041.0...	http://images.amazon.com/imag
1031117	276680	086505505X	0	Home Crafts (Historic Communities (Paperback))	Bobbie Kalman	1990	Crabtree Publishing Company	http://images.amazon.com/images/P/086505505X.0...	http://images.amazon.com/imag
1031118	276680	1564407284	0	The Old Sturbridge Village Cookbook, 2nd : Aut...	Caroline Sloat	1995	Globe Pequot	http://images.amazon.com/images/P/1564407284.0...	http://images.amazon.com/imag
1031119	276680	1582343594	0	City of Masks : A Cree Black Novel	Daniel Hecht	2004	Bloomsbury USA	http://images.amazon.com/images/P/1582343594.0...	http://images.amazon.com/imag

591896 rows × 10 columns

In [41]:

```
a=filtered_user.groupby('Book-Title').count()['Book-Rating']>100
#filtering only those books , which have a 100 ratings on it
filterd_book=a[a].index
filterd_book
```

Out[41]:

```
Index(['1984', '1st to Die: A Novel', '2nd Chance', 'A Bend in the Road',
       'A Case of Need',
       'A Child Called \It\": One Child's Courage to Survive"',
       'A Is for Alibi (Kinsey Millhone Mysteries (Paperback))',
       'A Map of the World', 'A Painted House', 'A Prayer for Owen Meany',
       ...
       'Where the Heart Is (Oprah's Book Club (Paperback))',
       'While I Was Gone', 'Whispers', 'White Oleander : A Novel',
       'White Oleander : A Novel (Oprah's Book Club)',
       'Wicked: The Life and Times of the Wicked Witch of the West',
       'Wild Animus', 'Wish You Well', 'Wuthering Heights',
       '\O\" Is for Outlaw"'],
      dtype='object', name='Book-Title', length=299)
```

In [42]:

```
final_rating=filtered_user[filtered_user['Book-Title'].isin(filterd_book)]
final_rating
```

Out[42]:

	User-ID	ISBN	Book-Rating	Book-Title	Book-Author	Year-Of-Publication	Publisher	Image-URL-S	
63	278418	0446520802	0	The Notebook	Nicholas Sparks	1996	Warner Books	http://images.amazon.com/images/P/0446520802.0...	http://images.amazon.com/images/P
65	3363	0446520802	0	The Notebook	Nicholas Sparks	1996	Warner Books	http://images.amazon.com/images/P/0446520802.0...	http://images.amazon.com/images/P
66	7158	0446520802	10	The Notebook	Nicholas Sparks	1996	Warner Books	http://images.amazon.com/images/P/0446520802.0...	http://images.amazon.com/images/P
67	8253	0446520802	10	The Notebook	Nicholas Sparks	1996	Warner Books	http://images.amazon.com/images/P/0446520802.0...	http://images.amazon.com/images/P
69	11676	0446520802	10	The Notebook	Nicholas Sparks	1996	Warner Books	http://images.amazon.com/images/P/0446520802.0...	http://images.amazon.com/images/P
...	...	...	...	...	...	...	...	...	
1025422	264317	0833563505	0	The Queen of the Damned (Vampire Chronicles (P...	Anne Rice	1999	Sagebrush Bound	http://images.amazon.com/images/P/0833563505.0...	http://images.amazon.com/images/P
1026724	266865	0531001725	10	The Catcher in the Rye	Jerome David Salinger	1973	Scholastic Library Pub	http://images.amazon.com/images/P/0531001725.0...	http://images.amazon.com/images/P
1028777	271284	0440910927	0	The Rainmaker	John Grisham	1995	Island	http://images.amazon.com/images/P/0440910927.0...	http://images.amazon.com/images/P
1029070	271705	B0001PIOX4	0	Fahrenheit 451	Ray Bradbury	1993	Simon & Schuster	http://images.amazon.com/images/P/B0001PIOX4.0...	http://images.amazon.com/images/P
1030868	275970	1586210661	9	Me Talk Pretty One Day	David Sedaris	2001	Time Warner Audio Major	http://images.amazon.com/images/P/1586210661.0...	http://images.amazon.com/images/P

45305 rows × 10 columns

In [43]:
```
pt=final_rating.pivot_table(index='Book-Title',columns='User-ID',values='Book-Rating')
pt
```

Out[43]:

User-ID	254	507	882	1424	1435	1733	1903	2033	2110	2276	...	275020	275970	276463	276680	277427	277478	277639	278137	278188	27841
Book-Title																					
1984	9.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	0.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Nal
1st to Die: A Novel	NaN	0.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Nal
2nd Chance	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	10.0	...	NaN	NaN	NaN	NaN	NaN	NaN	0.0	NaN	NaN	Nal
A Bend in the Road	0.0	NaN	0.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0.0	Nal
A Case of Need	NaN	NaN	NaN	NaN	7.0	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0.
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
Wicked: The Life and Times of the Wicked Witch of the West	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Nal
Wild Animus	NaN	NaN	NaN	NaN	5.0	1.0	0.0	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	0.0	0.0	NaN	0.0	NaN	Nal
Wish You Well	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Nal
Wuthering Heights	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Nal
\O\" Is for Outlaw"	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0.0	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Nal

299 rows × 1609 columns

In [44]:
```
pt.fillna(0,inplace=True)
```

In [45]:
```
pt
```

Out[45]:

User-ID	254	507	882	1424	1435	1733	1903	2033	2110	2276	...	275020	275970	276463	276680	277427	277478	277639	278137	278188	278418
Book-Title																					
1984	9.0	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	0.0	0.0	0.0	0.0
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	0.0	0.0	0.0	0.0
2nd Chance	0.0	0.0	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	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	0.0	0.0	0.0	0.0
A Case of Need	0.0	0.0	0.0	0.0	7.0	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
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
Wicked: The Life and Times of the Wicked Witch of the West	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	0.0	0.0	0.0	0.0
Wild Animus	0.0	0.0	0.0	0.0	5.0	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
Wish You Well	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	0.0	0.0	0.0	0.0
Wuthering Heights	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	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	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

299 rows × 1609 columns

In [46]:
```
from sklearn.metrics.pairwise import cosine_similarity
#Cosine similarity is a measure of similarity between two non-zero vectors of an inner product
#space based on the cosine of the angle between them, resulting in a value between -1 and 1
```

In [47]:
```
similarity_score=cosine_similarity(pt)
similarity_score
```

Out[47]:
```
array([[1.        , 0.06550754, 0.00805409, ..., 0.00705809, 0.04180813,
        0.03029366],
       [0.06550754, 1.        , 0.23629915, ..., 0.10249585, 0.05748184,
        0.12388301],
       [0.00805409, 0.23629915, 1.        , ..., 0.1449341 , 0.01093082,
        0.07992925],
       ...,
       [0.00705809, 0.10249585, 0.1449341 , ..., 1.        , 0.00957907,
        0.10082854],
       [0.04180813, 0.05748184, 0.01093082, ..., 0.00957907, 1.        ,
        0.02916002],
       [0.03029366, 0.12388301, 0.07992925, ..., 0.10082854, 0.02916002,
        1.        ]])
```

In [48]:
```
similarity_score.shape
```

Out[48]:
```
(299, 299)
```

In [49]:
```
def recommend(book_name):
    #index fetch in pt
    index=np.where(pt.index==book_name)[0][0]
    similar_item=list(enumerate(similarity_score[index]))
    #similar_item
    #sort them on basis on similarity_score
    similar_item=sorted(similar_item,key=lambda x:x[1],reverse=True)
    #get top 5 ,0th is itself distance
    similar_item=similar_item[1:6]
    similar_item

    listt=[]
    for i in similar_item:
        a=pt.index[i[0]]
        listt.append(a)
    return listt
```

In [50]:
```
mv=input("enter a movie name")
ans=recommend(mv)
ans
```

```
enter a movie name1984
```

Out[50]:
```
["The Handmaid's Tale",
 'The Catcher in the Rye',
 'Lord of the Flies',
 'The Vampire Lestat (Vampire Chronicles, Book II)',
 'Timeline']
```

Recommendation of book

In [51]:
```
recommended_books_data = books[books['Book-Title'].isin(ans)]
recommended_books_data.drop_duplicates(['Book-Title'],inplace=True)
recommended_books_data
```

```
C:\Users\Lenovo\AppData\Local\Temp\ipykernel_12488\68673933.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  recommended_books_data.drop_duplicates(['Book-Title'],inplace=True)
```

Out[51]:

	ISBN	Book-Title	Book-Author	Year-Of-Publication	Publisher	Image-URL-S	Image-URL-M	
28	0345417623	Timeline	MICHAEL CRICHTON	2000	Ballantine Books	http://images.amazon.com/images/P/0345417623.0...	http://images.amazon.com/images/P/0345417623.0...	h
90	0316769487	The Catcher in the Rye	J.D. Salinger	1991	Little, Brown	http://images.amazon.com/images/P/0316769487.0...	http://images.amazon.com/images/P/0316769487.0...	h
608	0345313860	The Vampire Lestat (Vampire Chronicles, Book II)	ANNE RICE	1986	Ballantine Books	http://images.amazon.com/images/P/0345313860.0...	http://images.amazon.com/images/P/0345313860.0...	h
1757	0399501487	Lord of the Flies	William Gerald Golding	1959	Perigee Trade	http://images.amazon.com/images/P/0399501487.0...	http://images.amazon.com/images/P/0399501487.0...	h
1985	0449212602	The Handmaid's Tale	Margaret Atwood	1989	Fawcett Books	http://images.amazon.com/images/P/0449212602.0...	http://images.amazon.com/images/P/0449212602.0...	h

if book is not in pt dataset , means we cannot recommand for it , but show similar matches of it

In [56]:
```
mv='19'
search = pt[pt.index.str.contains(mv)]
recommendations=list(search.index)[:5]
recommended_books_data = books[books['Book-Title'].isin(recommendations)]
recommended_books_data.drop_duplicates(['Book-Title'], inplace=True)
aa = list(recommended_books_data['Book-Title'])
bb = list(recommended_books_data['Image-URL-M'])
cc = list(recommended_books_data['Book-Author'])
```

```
C:\Users\Lenovo\AppData\Local\Temp\ipykernel_12488\4274341269.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  recommended_books_data.drop_duplicates(['Book-Title'], inplace=True)
```

In [57]:
```
recommended_books_data
```

Out[57]:

	ISBN	Book-Title	Book-Author	Year-Of-Publication	Publisher	Image-URL-S	Image-URL-M	
2104	0451524934	1984	George Orwell	1990	Signet Book	http://images.amazon.com/images/P/0451524934.0...	http://images.amazon.com/images/P/0451524934.0...	http://ima