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

```
books = pd.read_csv('/content/drive/MyDrive/bookrecommender/Books.csv.zip')
users = pd.read_csv('/content/drive/MyDrive/bookrecommender/Users.csv.zip')
ratings = pd.read_csv('/content/drive/MyDrive/bookrecommender/Ratings.csv.zip')
```

```
/tmp/ipython-input-5-492682612.py:1: DtypeWarning: Columns (3) have mixed types. Specify dtype option on import or set low_memory=False.
  books = pd.read_csv('/content/drive/MyDrive/bookrecommender/Books.csv.zip')
```

```
books.head()
```

	ISBN	Book-Title	Book-Author	Year-Of-Publication	Publisher	Image-URL-S	
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
1	0002005018	Clara Callan	Richard Bruce Wright	2001	HarperFlamingo Canada	http://images.amazon.com/images/P/0002005018.0...	http://images.amazon.com/images
2	0060973129	Decision in Normandy	Carlo D'Este	1991	HarperPerennial	http://images.amazon.com/images/P/0060973129.0...	http://images.amazon.com/images
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
4	0393045218	The Mummies of Urumchi	E. J. W. Barber	1999	W. W. Norton &amp; Company	http://images.amazon.com/images/P/0393045218.0...	http://images.amazon.com/images

```
users.head()
```

	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

```
ratings.head()
```

	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

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

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

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

	0
ISBN	0
Book-Title	0
Book-Author	2
Year-Of-Publication	0
Publisher	2
Image-URL-S	0
Image-URL-M	0
Image-URL-L	3

dtype: int64

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

	0
User-ID	0
Location	0
Age	110762

dtype: int64

```
ratings.isnull().sum()
```

	0
User-ID	0
ISBN	0
Book-Rating	0

dtype: int64

```
books.duplicated().sum()
```

```
np.int64(0)
```

```
ratings.duplicated().sum()
```

```
np.int64(0)
```

```
users.duplicated().sum()
```

```
np.int64(0)
```

POPULARITY BASED RECOMMENDER SYSTEM

```
ratings_with_name=ratings.merge(books,on='ISBN')
```

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

	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

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

	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

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

	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

```
popular_df=popular_df[popular_df['num_ratings']>=250].sort_values('avg_rating',ascending=False).head(50)
```

```
popular_df=popular_df.merge(books,on='Book-Title').drop_duplicates('Book-Title')[['Book-Title','Book-Author','Image-URL-M','num_ratings','av
```

```
popular_df['Image-URL-M'][0]
```

```
'http://images.amazon.com/images/P/0439136350.01.MZZZZZZZ.jpg'
```

COLLABORATIVE FILTERING BASED RECOMMENDER SYSTEM

```
x=ratings_with_name.groupby('User-ID').count()['Book-Rating']>200
educated_users=x[x].index
```

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

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

```
famous_books
```

```
Index(['1984', '1st to Die: A Novel', '2nd Chance', '4 Blondes',
       'A Bend in the Road', 'A Case of Need',
       'A Child Called \It\": One Child's Courage to Survive"',
       'A Civil Action', 'A Day Late and a Dollar Short', 'A Fine Balance',
       ...
       'Winter Solstice', 'Wish You Well', 'Without Remorse',
       'Wizard and Glass (The Dark Tower, Book 4)', 'Wuthering Heights',
       'Year of Wonders', 'You Belong To Me',
       'Zen and the Art of Motorcycle Maintenance: An Inquiry into Values',
       'Zoya', '\O\" Is for Outlaw"'],
      dtype='object', name='Book-Title', length=706)
```

```
final_ratings=filtered_rating[filtered_rating['Book-Title'].isin(famous_books)]
```

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

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

```
pt
```

User-ID	254	2276	2766	2977	3363	4017	4385	6251	6323	6543	...	271705	273979	274004	274061	274301	274308	275970	27742
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	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	9.0	...	0.0	0.0	0.0	0.0	0.0	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	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	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	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	0.0	0.0	0.0	0.0	0.0	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	0.0	0.0	0.0	0.0	0.0	0
Zen and the Art of Motorcycle Maintenance:	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

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

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

```
similarity_scores.shape
```

```
(706, 706)
```

```
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[1],reverse=True)[1:11]
  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'].values))

    data.append(item)
    return data
```

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

```
[['Animal Farm',
  'George Orwell',
  'http://images.amazon.com/images/P/0451526341.01.MZZZZZZZ.jpg']]
```

```
pt.index[545]
```

```
'The Handmaid's Tale'
```

```
import pickle
pickle.dump(popular_df,open('popular.pkl','wb'))
```

```
pickle.dump(pt,open('pt.pkl','wb'))
pickle.dump(books,open('books.pkl','wb'))
pickle.dump(similarity_scores,open('similarity_scores.pkl','wb'))
```

Start coding or generate with AI.