3 dataframe

boo	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				
1	0002005018	Clara Callan	Richard Bruce Wright	2001	HarperFlamingo Canada	http://images.amazon.com/images/P/0002005018.0				
2	0060973129	Decision in Normandy	Carlo D'Este	1991	HarperPerennial	http://images.amazon.com/images/P/0060973129.0				
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				
4	0393045218	The Mummies of Urumchi	E. J. W. Barber	1999	W. W. Norton & Company	http://images.amazon.com/images/P/0393045218.0				

books.shape(271360, 8)

	/	٠,
users.	neaa(

User-ID		Location	
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

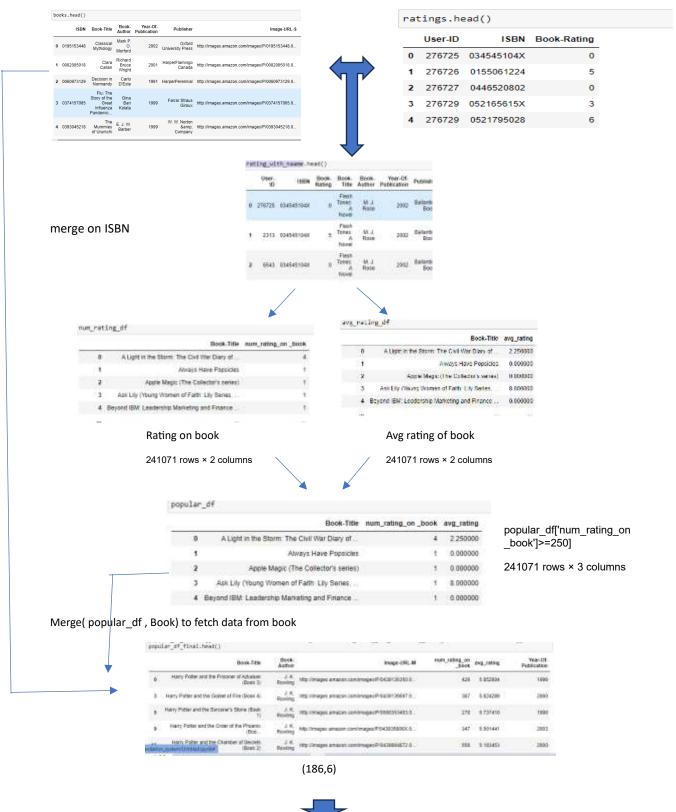
users.shape(278858, 3)

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

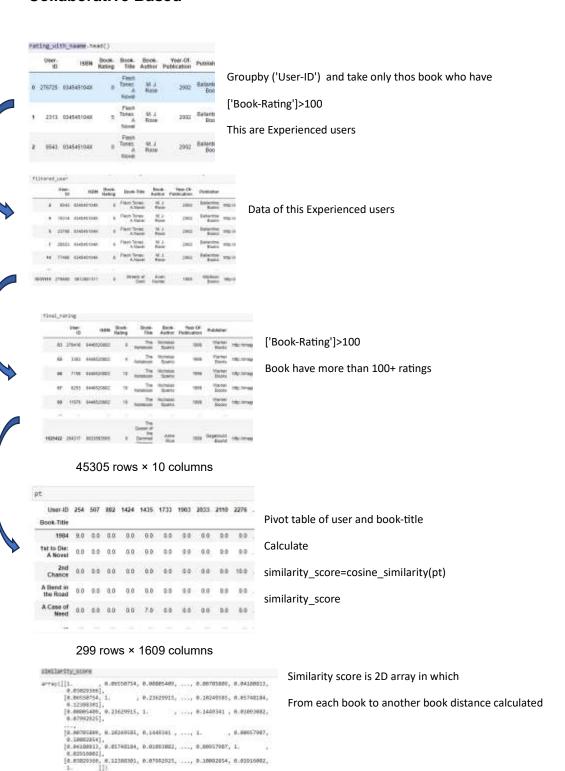
ratings.shape(1149780, 3)

Popularity based Recommender System



Random 10 as output, Trending books

Collaborative Based



```
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
  similar_item=sorted(similar_item,key=lambda x:x[1],reverse=True)
                                                                                #sort them on basis on similarity_score
  similar_item=similar_item[1:6]
                                                                               #get top 5,0th is itself distance
  similar\_item
  listt=[]
  for i in similar_item: #similar items have index and similarity score , take only index
    a=pt.index[i[0]]
                     #find book name on that index
    listt.append(a)
                      #append all this name to list
  return list
mv=input("enter a movie name") #input a book name
ans=recommend(mv) # function calling
ans
****** ans is a list of 5 recommaded books now find this book name in our books dataset and print details *******
```