## Book Rental Recommendations

#### DESCRIPTION

Book Rent is the largest online and offline book rental chain in India. They provide books of various genres, such as thrillers, mysteries, romances, and science fiction. The company charges a fixed rental fee for a book per month. Lately, the company has been losing its user base. The main reason for this is that users are not able to choose the right books for themselves. The company wants to solve this problem and increase its revenue and profit.

### Project Objective:

You, as an ML expert, should focus on improving the user experience by personalizing it to the user's needs. You have to model a recommendation engine so that users get recommendations for books based on the behavior of similar users. This will ensure that users are renting the books based on their tastes and traits.

# **New Section**

Following operations should be performed:

- 1.Read the books dataset and explore it
- 2.Clean up NaN values
- 3. Read the data where ratings are given by users
- 4. Take a quick look at the number of unique users and books
- 5. Convert ISBN variables to numeric numbers in the correct order
- 6.Convert the user\_id variable to numeric numbers in the correct order
- 7. Convert both user\_id and ISBN to the ordered list, i.e., from 0...n-1
- 8.Re-index the columns to build a matrix
- 9. Split your data into two sets (training and testing)
- 10. Make predictions based on user and item variables
- 11.Use RMSE to evaluate the predictions

#### Importing required libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

Books=pd.read\_csv('BX-Books.csv',encoding='latin-1')

Books

		isbn	book_tit	tle book_autho	r year_of_publication	n publishe
	<b>0</b> 19515	53448	Classical Mytholo	ogy Mark P. C Morfor	200	2 Oxford Universit Pres
ks.h	ead()			Dishard Davis	-	HamarFlamina
	isbn		book_title	book_author	year_of_publication	publishe
0	195153448		Classical Mythology	, Mark P. O. Morford	2002	Oxford University Pres
1	2005018		Clara Callan	Richard Bruce Wright	2001	HarperFlamingo Canad
2	60973129	[	Decision in Normandy	Carlo D'Este	1991	HarperPerenni
3	374157065		The Story of the Great Influenza Pandemic		1999	Farrar Straus Giro
In	'publish		e', 'book_author'	, 'year_of_publica	rtion',	
	acype- or					
ks.i	nfo()					
<c Ra</c 	nfo() lass 'pandas. ngeIndex: 158 ta columns (1	8836 entrie	s, 0 to 158835	Dtype		

###cleanup NaN values
Books.dropna()

Book\_rating

		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	
			***				
	158830	321125606	SF Writer (APA Update) (2nd Edition)	John Ruszkiewicz	2002	Prentice Hall	
	158831	1563923971	Toyota Celica 1986-1999: Front Wheel Drive Mod	Larry Warren	2000	Haynes Publications	
	158832	1556114990	A Father's Kisses: A Novel	Bruce Jay	1996	Penguin USA	
Book_rating=pd.read_csv('BX-Book-Ratings.csv',encoding='latin-1')							



Book\_rating.head()

	user_id	isbn	rating
0	276725	034545104X	0.0
1	276726	155061224	5.0
2	276727	446520802	0.0
3	276729	052165615X	3.0
4	276729	521795028	6.0

##2. clean up NaN values
Book\_rating.dropna()

	user_id	isbn	rating
0	276725	034545104X	0.0
1	276726	155061224	5.0
2	276727	446520802	0.0
3	276729	052165615X	3.0
4	276729	521795028	6.0
110103	25436	330427660	0.0
110104	25436	345319672	8.0
110105	25436	349101779	9.0
110106	25436	374529035	8.0
110107	25436	380002930	0.0

110108 rows × 3 columns

3###book users

Book\_users=pd.read\_csv('BX-Users.csv', encoding='latin-1')

/usr/local/lib/python3.8/dist-packages/IPython/core/interactiveshell.py:3326: DtypeWarning: Columns (0) have mixed types.Specify dtype c exec(code\_obj, self.user\_global\_ns, self.user\_ns)

Book\_users



Book\_users.head()

Age	Location	user_id		
NaN	nyc, new york, usa	1	0	
18.0	stockton, california, usa	2	1	
NaN	moscow, yukon territory, russia	3	2	
17.0	porto, v.n.gaia, portugal	4	3	
NaN	farnborough, hants, united kingdom	5	4	

2##Clean up NAN values
Book\_users.dropna()



168096 rows × 3 columns

## 4. Unique users and books

```
4.#### unique users and books
Books.book_title.unique()
```

Book\_users.user\_id.unique()

```
array([1, 2, 3, ..., '278856', '278857', '278858'], dtype=object)
```

#### 5. Convert ISBN variables to numeric numbers in the correct order

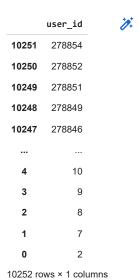
```
Books[['isbn']].groupby(['isbn'], as_index=False).mean().sort_values(by='isbn', ascending=False)
```

```
isbn
158835 B00029DGGO
158834
        B000234N76
158833
        B0001PIOX4
158832
       B0001GMSV2
158831
       B0001GDNCK
  ...
  4
         000215787X
         000171421X
  3
         000123207X
  2
         000104799X
  1
```

Convert the user\_id variable to numeric numbers in the correct order

```
158836 rows x 1 columns

Book_rating[['user_id']].groupby(['user_id'], as_index=False).mean().sort_values(by='user_id', ascending=False)
```



Convert both user\_id and ISBN to the ordered list, i.e., from 0...n-1

Book\_rating.groupby('isbn')['user\_id'].count().describe()

```
69030.000000
count
             1.595075
mean
std
             2.415267
             1.000000
min
             1.000000
25%
50%
             1.000000
             1.000000
75%
           214.000000
max
Name: user_id, dtype: float64
```

Re-index the columns to build a matrix

```
# This is formatted as code
```

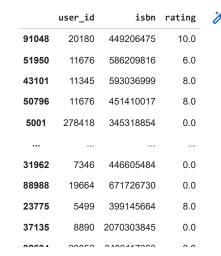
# ▼ Train Test Split

\*\* Split your data into a training set and a testing set.\*\*

Double-click (or enter) to edit

from sklearn.model\_selection import train\_test\_split train\_test\_split(Book\_rating, shuffle=False) user\_id isbn rating 276725 034545104X 276726 155061224 5.0 1 276727 446520802 2 0.0 3 276729 052165615X 3.0 4 276729 521795028 6.0 16996 044651652X 82576 0.0 82577 16996 451172027 0.0 82578 16996 452269571 0.0 16996 82579 486275434 0.0 82580 16996 517605171 9.0 [82581 rows x 3 columns], user\_id isbn rating 82581 16996 553212451 0.0 82582 16996 553213237 0.0 82583 16996 670872695 7.0 16996 671510053 82584 7.0 82585 16996 671623249 0.0 110104 25436 345319672 8.0 110105 25436 349101779 9.0 110106 25436 374529035 8.0 110107 25436 380002930 0.0 110108 25 NaN NaN [27528 rows x 3 columns]] train\_test\_split(Book\_rating) user\_id isbn rating 26974 6345 64420485 5.0 43698 11601 345272129 77064 16634 553281836 10.0 84178 17789 2020230097 0.0 13062 1517 8489618844 6.0 44359 553380400 11601 0.0 91445 20309 345396065 8.0 2951 277879 671737643 0.0 80740 16795 932592708 0.0 76359 16611 440213525 0.0 [82581 rows x = 3 columns], user id isbn rating 15163 2276 671440683 0.0 94812 21252 140235086 4.0 49746 11676 440220440 0.0 60553 12272 140620222 0.0 18207 3363 425167720 0.0 8681 1932360239 36466 8.0 87919 19119 679879579 7.0 18818 3371 440225892 0.0 77891 16755 743411323 0.0 373218095 35092 8284 0.0 [27528 rows x 3 columns]] Book\_rating\_train, Book\_rating\_test, = train\_test\_split(Book\_rating, test\_size=0.2, random\_state=20) Book\_rating\_train.shape, Book\_rating\_test.shape ((88087, 3), (22022, 3))

Book\_rating\_train



Book\_rating\_test

	user_id	isbn	rating	7
6342	278418	440204275	0.0	
53794	11676	749733330	0.0	
13803	1903	531110397	0.0	
55233	11676	944475485	5.0	
108791	25032	058604468X	10.0	
107478	24767	446525502	0.0	
92034	20501	3100485041	9.0	
3703	278144	039914465X	0.0	
21210	4131	696204800	0.0	
48700	11676	038531700X	9.0	

22022 rows × 3 columns