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_title	book_author	year_of_publication	publisher
0	195153448	Classical Mythology	Mark P. O. Morford	2002	Oxford University Press

```
Books.head()
```

	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

```
Books.columns
```

```
Index(['isbn', 'book_title', 'book_author', 'year_of_publication',
       'publisher'],
      dtype='object')
```

```
Books.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 158836 entries, 0 to 158835
Data columns (total 5 columns):
 #   Column               Non-Null Count   Dtype 
---  ------               --------------   ----- 
 0   isbn                 158836 non-null  object
 1   book_title           158836 non-null  object
 2   book_author          158835 non-null  object
 3   year_of_publication  158835 non-null  object
 4   publisher            158833 non-null  object
dtypes: object(5)
memory usage: 6.1+ MB
```

```
###cleanup NaN values
Books.dropna()
```

	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
```

	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
...	...	...	...
110104	25436	345319672	8.0
110105	25436	349101779	9.0
110106	25436	374529035	8.0

```
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
```

	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
...	...	...	...

```
Book_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

```
2##Clean up NAN values
Book_users.dropna()
```

	user_id	Location	Age
1	2	stockton, california, usa	18.0
3	4	porto, v.n.gaia, portugal	17.0
5	6	santa monica, california, usa	61.0
9	10	albacete, wisconsin, spain	26.0
10	11	melbourne, victoria, australia	14.0
...	...	...	...
278849	278849	georgetown, ontario, canada	23.0
278851	278851	dallas, texas, usa	33.0
278852	278852	brisbane, queensland, australia	32.0
278853	278853	stranraer, n/a, united kingdom	17.0
278855	278855	tacoma, washington, united kingdom	50.0

168096 rows × 3 columns

4.Unique users and books

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

```
array(['Classical Mythology', 'Clara Callan', 'Decision in Normandy', ...,
       'Where Do I Put the Decimal Point How To',
       "Never Fight Fair!: Navy Seals' Stories of Combat and Adventure",
       'Beside the Still Waters (Colour Gift Boo'], dtype=object)
```

```
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
3	000171421X
2	000123207X
1	000104799X

Convert the user_id variable to numeric numbers in the correct order

158836 rows × 1 columns

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

	user_id
10251	278854
10250	278852
10249	278851
10248	278849
10247	278846
...	...
4	10
3	9
2	8
1	7
0	2

10252 rows × 1 columns

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

```
Book_rating.groupby('isbn')['user_id'].count().describe()
```

```
count    69030.000000
mean         1.595075
std          2.415267
min          1.000000
25%          1.000000
50%          1.000000
75%          1.000000
max        214.000000
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
 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
 ...        ...         ...     ...
 82576    16996  044651652X     0.0
 82577    16996   451172027     0.0
 82578    16996   452269571     0.0
 82579    16996   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
 82584     16996  671510053     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     0.0
 77064    16634   553281836    10.0
 84178    17789  2020230097     0.0
 13062     1517  8489618844     6.0
 ...        ...         ...     ...
 44359    11601   553380400     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
 ...        ...         ...     ...
 36466     8681  1932360239     8.0
 87919    19119   679879579     7.0
 18818     3371   440225892     0.0
 77891    16755   743411323     0.0
 35092     8284   373218095     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
```

	user_id	isbn	rating
91048	20180	449206475	10.0
51950	11676	586209816	6.0
43101	11345	593036999	8.0
50796	11676	451410017	8.0
5001	278418	345318854	0.0
...	...	...	...
31962	7346	446605484	0.0
88988	19664	671726730	0.0
23775	5499	399145664	8.0
37135	8890	2070303845	0.0

Book_rating_test

	user_id	isbn	rating
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