Book Recommendation System

Abstract

With traditional methods of reading being replaced by the world on the web, recommendation systems have become a necessity more than just a luxury. More and more companies are investing in building recommendation systems to engage users by providing them with personalized recommendations which depend on the users' own behaviors. These systems have proved to be a significant part in boosting sales of various companies. Our aim was to build a recommendation system using various techniques such as content based filtering and collaborative filtering.

Introduction

The aim of this project was to recommend books to users based on different techniques, depending on the user preference. The book crossing dataset was downloaded from Kaggle. The dataset underwent pre-processing, and a few insights were made from the dataset by performing exploratory data analysis on the dataset. Using these insights and various techniques, recommendation systems were designed.

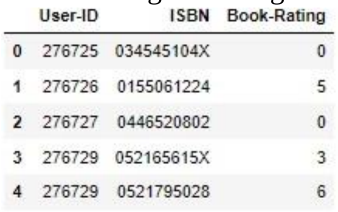
Every technique does not necessarily recommend the same set of books to the users for the same input book. Every technique takes a book as an input and yields recommendations based on a different similarity metric. These metrics include cosine similarity, correlation, or k-nearest neighbors.

1. Datasets, Pre-processing and Exploratory Data Analysis

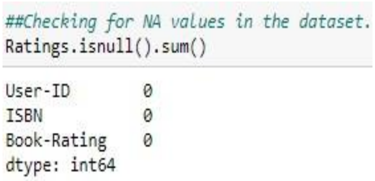
The datasets used for the Book Recommendation system are as follows:

1.1 <u>Ratings Data</u>

The Ratings dataset contains the User ID, ISBN of the book and the respective book ratings by the users. It consists of 1,149,780 rows and 3 columns with duplicates as the same users have rated different books and unique users have rated the same books. The book ratings are in the scale 0 to 10 where 0 is the lowest rating and 10 is the highest rating. There are no missing values in the ratings dataset



	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



```
##Checking for NA values in the dataset.
Ratings.isnull().sum()

User-ID        0
ISBN           0
Book-Rating    0
dtype: int64
```

The number of unique users in this dataset are 77,805 and the unique number of books are 185,973.

The following plot shows the most number of ratings given by the users. Fig 1 shows that most of the users gave 0 ratings to the books. It causes a highly uneven distribution. Whereas Fig 2 shows the most number of given ratings without the 0 rating and it can be seen that rating 8 was the most frequently given rating by the users. Here, 0 rating will be considered an implicit rating and 1-10 ratings will be considered as explicit ratings. For the book recommendation system only explicit ratings will be considered.

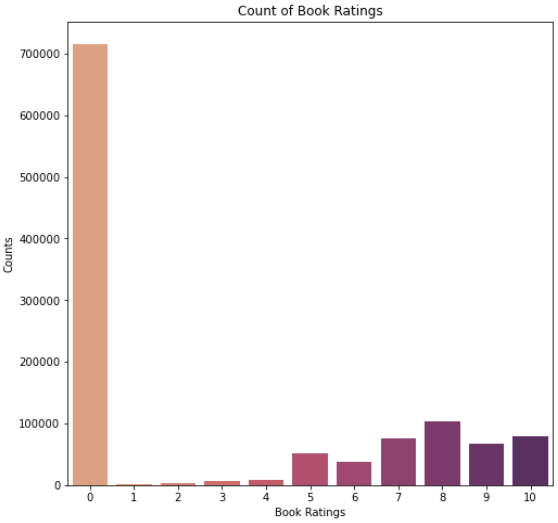


Figure 1

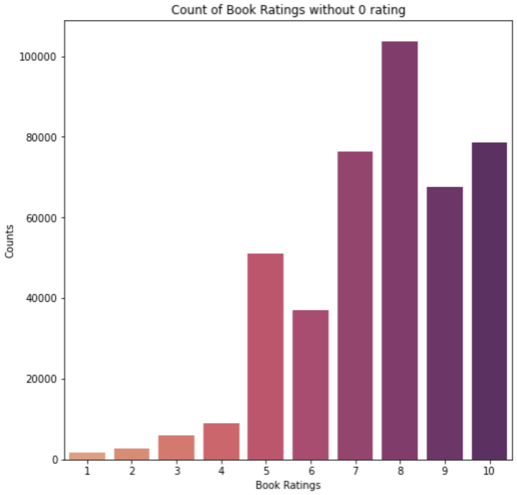
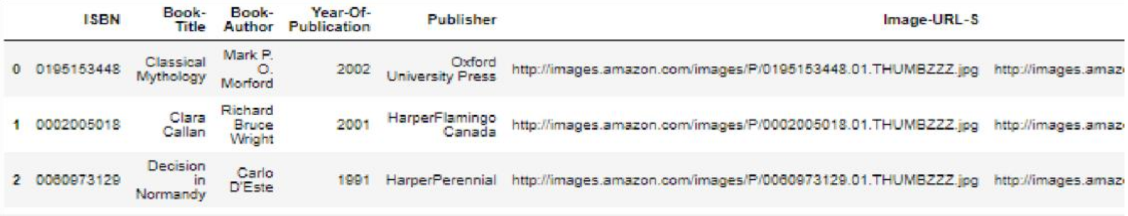


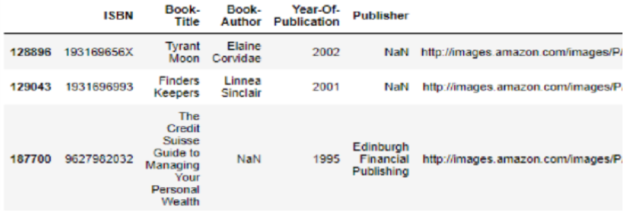
Figure 2

1.2 Books Data

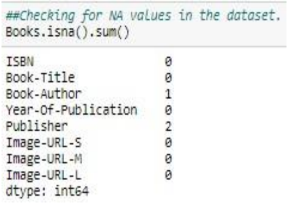
The Books dataset contains all the attributes of each book. It contains the book id, which is the ISBN of the book, the book title, author of the book, publisher, year of publication and the image urls of the book covers. The dataset consists of 27,179 rows and 8 columns. As the only image url used by the system was the small size, the other 2 columns for different sizes of book images were dropped.



	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/0195153448.01.THUMBZZZ.jpg	http://images.amaz
1	0002005018	Clara Callan	Richard Bruce Wright	2001	HarperFlamingo Canada	http://images.amazon.com/images/P/0002005018.01.THUMBZZZ.jpg	http://images.amaz
2	0060973129	Decision in Normandy	Carlo D'Este	1991	HarperPerennial	http://images.amazon.com/images/P/0060973129.01.THUMBZZZ.jpg	http://images.amaz



	ISBN	Book-Title	Book-Author	Year-Of-Publication	Publisher	
128896	193169656X	Tyrant Moon	Elaine Corvidae	2002	NaN	http://images.amazon.com/images/P/
129043	1931696993	Finders Keepers	Linnea Sinclair	2001	NaN	http://images.amazon.com/images/P.
187700	9627982032	The Credit Suisse Guide to Managing Your Personal Wealth	NaN	1995	Edinburgh Financial Publishing	http://images.amazon.com/images/P.

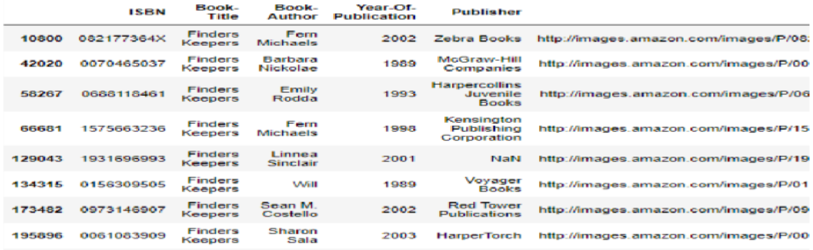


```
##Checking for NA values in the dataset.
Books.isna().sum()
```

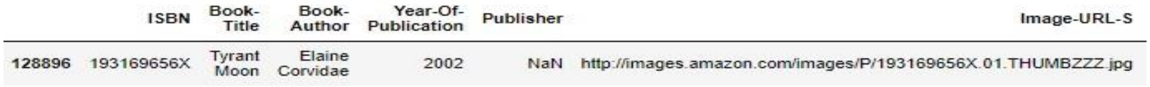
```
ISBN                   0
Book-Title             0
Book-Author            1
Year-Of-Publication    0
Publisher              2
Image-URL-S            0
Image-URL-M            0
Image-URL-L            0
dtype: int64
```

It was observed that there were 2 missing values under publishers and 1 missing value for book authors. The books “Tyrant Moon” and “Finders Keepers” have a missing Publisher and the book “The Credit Suisse Guide to Managing Your Personal Wealth” has a missing Book-Author.

It was found that the book "Finders Keepers" has various ISBN values and each one has a different Author and Publisher. These books were written by different authors and published by various publishers under the same book title. Therefore, as the publisher was unknown this book was dropped from the dataset. For the book “Tyrant Moon” there was no other row with the same book title and the publisher was unknown, therefore, this book was dropped as well.

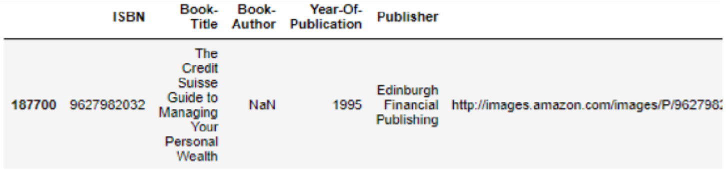


	ISBN	Book-Title	Book-Author	Year-Of-Publication	Publisher	
10800	082177364X	Finders Keepers	Fern Michaels	2002	Zebra Books	http://images.amazon.com/images/P/08:
42020	0070465037	Finders Keepers	Barbara Nickolae	1989	McGraw-Hill Companies	http://images.amazon.com/images/P/00
58267	0688118461	Finders Keepers	Emily Rodda	1993	Harpercollins Juvenile Books	http://images.amazon.com/images/P/06
66681	1575663236	Finders Keepers	Fern Michaels	1998	Kensington Publishing Corporation	http://images.amazon.com/images/P/15
129043	1931696993	Finders Keepers	Linnea Sinclair	2001	NaN	http://images.amazon.com/images/P/19
134315	0156309505	Finders Keepers	Will	1989	Voyager Books	http://images.amazon.com/images/P/01
173482	0973146907	Finders Keepers	Sean M. Costello	2002	Red Tower Publications	http://images.amazon.com/images/P/09
195896	0061083909	Finders Keepers	Sharon Sala	2003	HarperTorch	http://images.amazon.com/images/P/00

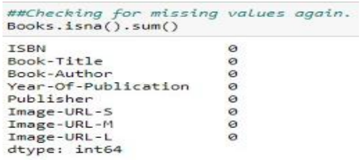


	ISBN	Book-Title	Book-Author	Year-Of-Publication	Publisher	Image-URL-S
128896	193169656X	Tyrant Moon	Elaine Corvidae	2002	NaN	http://images.amazon.com/images/P/193169656X.01.THUMBZZZ.jpg

For the book “The Credit Suisse Guide to Managing Your Personal Wealth” , there was only one row with the missing author, therefore, as the author remained unknown, this book was dropped as well. There were no missing values in the books dataset after the pre-processing.



	ISBN	Book-Title	Book-Author	Year-Of-Publication	Publisher	
187700	9627982032	The Credit Suisse Guide to Managing Your Personal Wealth	NaN	1995	Edinburgh Financial Publishing	http://images.amazon.com/images/P/962798:



```
##Checking for missing values again.
Books.isna().sum()
```

```
ISBN                      0
Book-Title                0
Book-Author               0
Year-Of-Publication       0
Publisher                 0
Image-URL-S               0
Image-URL-M               0
Image-URL-L               0
dtype: int64
```

Exploratory data analysis was applied on the books dataset and the findings are shown in figures below

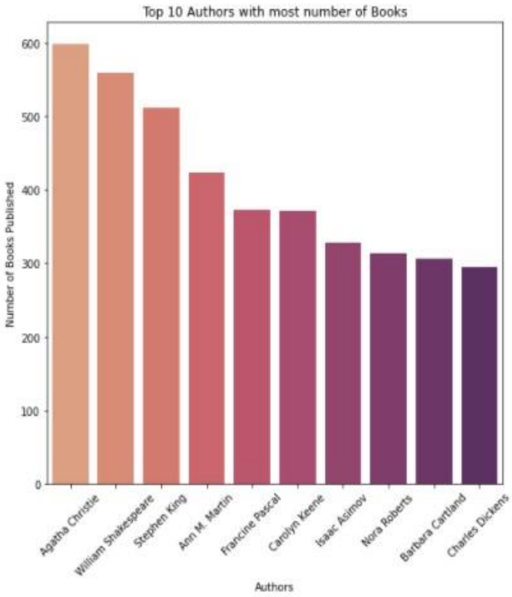


Figure 3

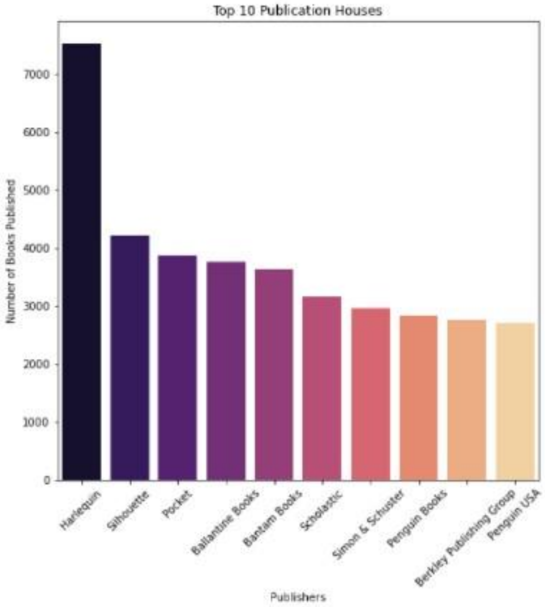


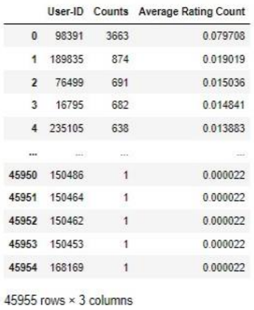
Figure 4

For the top 10 book authors with the most number of books, “Agatha Christine” has close to 600 books in the books dataset. Looking at the top 10 publishers, “Harlequin” has the most number of books published.

The year of publication column in the books dataset had a lot of invalid years such as ‘0’, ’1378’ , ‘2030’ , ‘2037’ , ‘2050’ , ‘2024’ and ‘1376’ and they had 0 number of books published, therefore, these particular entries were dropped.

In Fig 7 (Appendix) , the books published each year and a left-skewed distribution can be seen which shows an increase in the number of books published with increasing year. The most number of books published were in 2002 with the number of published books as 17,627.

The dataframe below shows the average rating count per user. The first entry shows that 1 user with user id ‘98391’ has rated 1,554 books which is the maximum number of books rated by a single user. The average rating per user can also be seen in the Fig 5 which clearly shows a spike in books rated and refers to the number 1,554



	User-ID	Counts	Average Rating Count
0	98391	3663	0.079708
1	189835	874	0.019019
2	76499	691	0.015036
3	16795	682	0.014841
4	235105	638	0.013883
...	...	...	...
45950	150486	1	0.000022
45951	150464	1	0.000022
45952	150462	1	0.000022
45953	150453	1	0.000022
45954	168169	1	0.000022

45955 rows × 3 columns

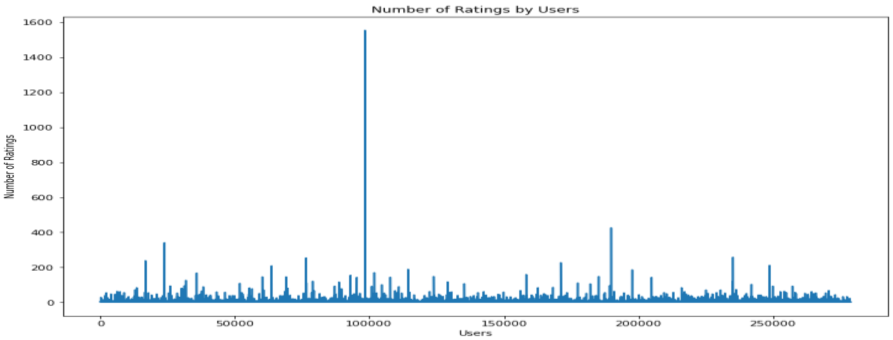
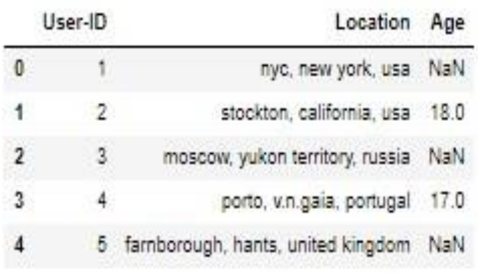


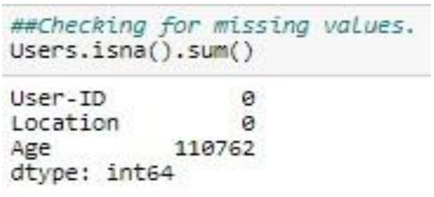
Figure 5

1.3 Users Data

The User data has the demographic information of the users, which include user-id, the location of user and the age of user. The location of the user was further split into city and county. The age column had a lot of missing values but as the age of the user was not used in any analysis or in the other algorithms it was left as it is.



	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



```
##Checking for missing values.
Users.isna().sum()
```

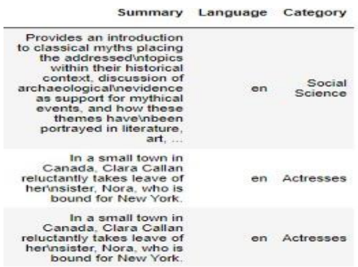
```
User-ID          0
Location         0
Age         110762
dtype: int64
```

An analysis was performed using the location of the user Fig 8 (Appendix) and it was observed that the majority of the users were from the USA, followed by a few users from Canada, UK and Germany, which had similar numbers of users.

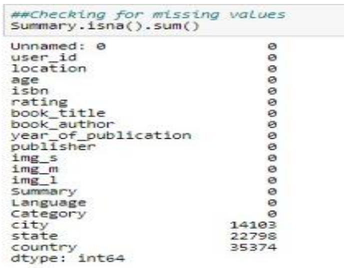
1.4 Summary Data

The Summary data has the 2 main features which are used in the recommendation system, one being the summary of the book and other being the category of the book. The summary explains the gist of the book in just 1-2 sentences. The category column is used to group similar books together which is useful in recommending similar books to users based on their interest. This makes it easier to recommend books which are in the same category.

The summary dataset consists of 1,031,175 rows and 19 columns which is basically a merged dataset of all the datasets introduced above. The missing values in the dataset are only in the city, state and county of the user. As the location of the user was already analysed in the User data, the missing values were not removed from this dataset, simply to avoid dropping essential summary and categories of different books.



Summary	Language	Category
Provides an introduction to classical myths placing the addressed\ntopics within their historical context, discussion of archaeological\nevidence as support for mythical events, and how these themes have\nbeen portrayed in literature, art, ...	en	Social Science
In a small town in Canada, Clara Callan reluctantly takes leave of her\nsister, Nora, who is bound for New York.	en	Actresses
In a small town in Canada, Clara Callan reluctantly takes leave of her\nsister, Nora, who is bound for New York.	en	Actresses



```
##Checking for missing values
Summary.isna().sum()
```

```
Unnamed: 0                 0
user_id                    0
location                   0
age                        0
isbn                       0
rating                     0
book_title                 0
book_author                0
year_of_publication        0
publisher                  0
img_s                      0
img_m                      0
img_l                      0
Summary                    0
Language                   0
Category                   0
city                   14103
state                  22798
country                35374
dtype: int64
```

Fig 9 (Appendix) shows us the top 10 categories of books. Around 78% of books fall in the ‘Fiction’ category, followed by 8% in ‘Juvenile Fiction’, 4% in ‘Biography and Autobiography’ and the remaining 1% books fall in categories like ‘Humor’, ‘Religion’, ‘Juvenile Nonfiction’ and ‘Social Science’

Figure 6 shows the highest number of ratings given by the user under different categories. It is observed that under the ‘Fiction’ category, most users have given a rating of 8 followed by a rating of 7 and then rating 9. This clearly shows the popularity of different categories and the reading preferences of users.

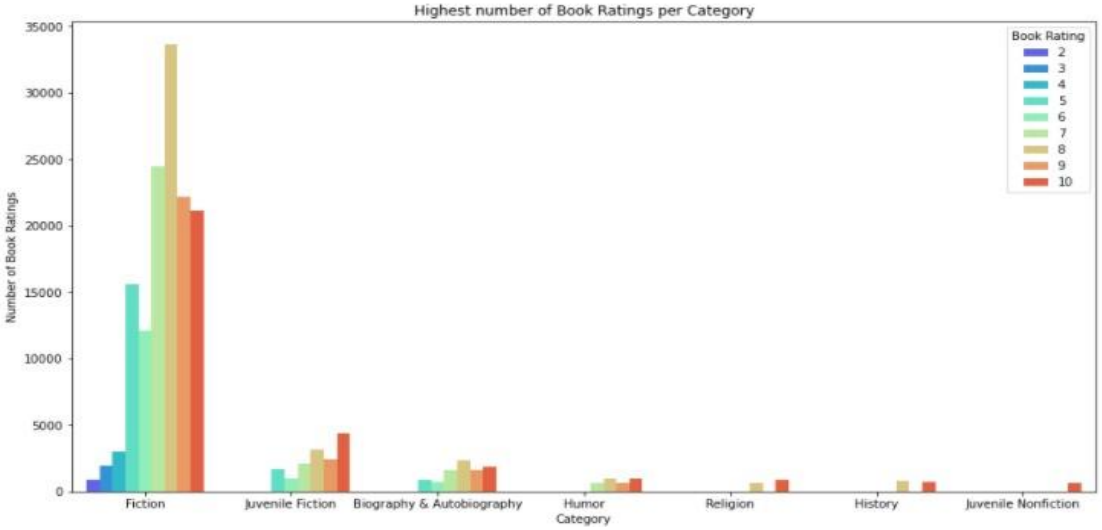


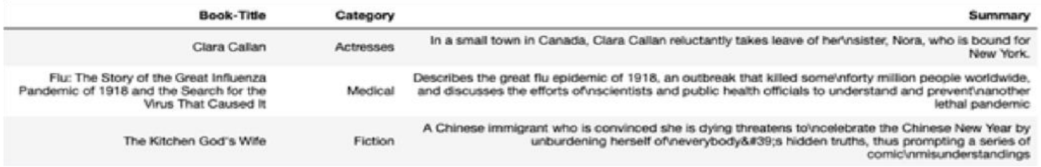
Figure 6

2. Techniques

2.1 Content Based Filtering: A Naïve Approach

Recommendation systems designed using the content-based filtering approach are built upon the assumption that a user prefers to use (in this case, read) similar kinds of items. This method either uses a feature of the user to create a user profile or a feature of the item to create an item profile and use these profiles to recommend items.

The dataset used in the project has a Summary table which has a summary column and a category column that describes each book in the dataset in one to two sentences and the category of the book respectively.



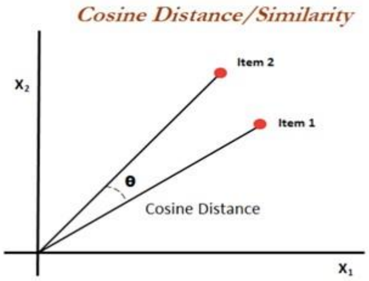
Book-Title	Category	Summary
Clara Callan	Actresses	In a small town in Canada, Clara Callan reluctantly takes leave of her\nsister, Nora, who is bound for New York.
Flu: The Story of the Great Influenza Pandemic of 1918 and the Search for the Virus That Caused It	Medical	Describes the great flu epidemic of 1918, an outbreak that killed some\nforty million people worldwide, and discusses the efforts of\nscientists and public health officials to understand and prevent\nanother lethal pandemic
The Kitchen God's Wife	Fiction	A Chinese immigrant who is convinced she is dying threatens to\ncelebrate the Chinese New Year by unburdening herself of\neverybody&#39;s hidden truths, thus prompting a series of comic\nmisunderstandings

The idea was to build a system that uses the summary column as a feature of the book (item) and build a book profile (item profile) and recommend books based on similarity between their profiles.

The method uses cosine similarity as a metric to compute similarities between the books. Cosine similarity can be used to measure similarity between items irrespective of their size.

$$Cosine(A, B) = \overline{||A||^{A\cdot B}||B||}$$

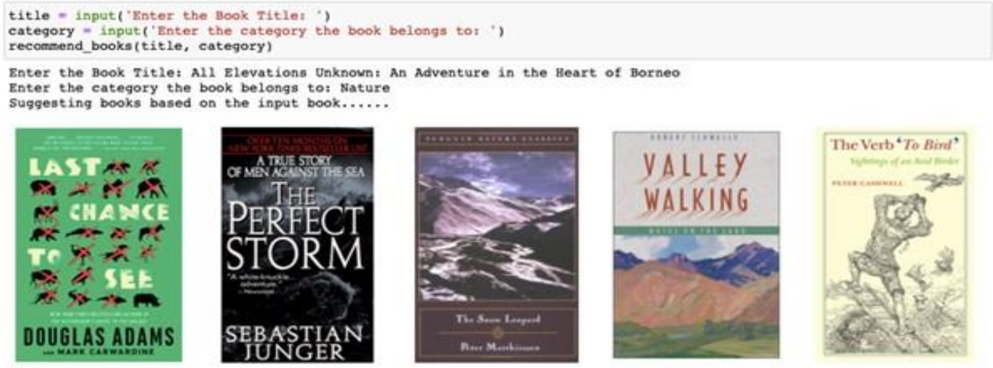
where A and B are items in a d-dimensional space.



Steps:

1. Take the book title and category of the book as input from the user.
2. Subset data from the df based on books with the same category. The data now contains rows where the category of the book is the input category (let's call it reduced_df for simplicity).
3. Convert the summary column of the reduced_df to TF-IDF matrix using the TfidfVectorizer from the Scikit-learn library.
4. Compute the cosine similarity between the summaries and storing the indices of the 5 most similar books based on cosine similarity.

5. Use the indices to display book covers as recommendations.
6. If the book title given as input to the function isn't present in the df_reduced data, the system simply informs the user and recommends top 5 books with the highest rating belonging to the input category.



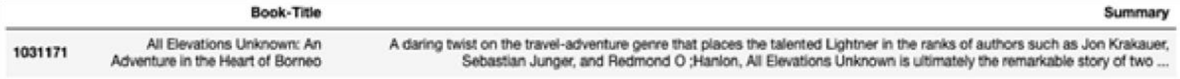
```
title = input('Enter the Book Title: ')
category = input('Enter the category the book belongs to: ')
recommend_books(title, category)
```

```
Enter the Book Title: All Elevations Unknown: An Adventure in the Heart of Borneo
Enter the category the book belongs to: Nature
Suggesting books based on the input book......
```

Recommended books using content based filtering

<u>The input book for the above recommendation</u>:

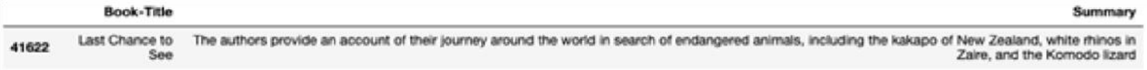
All Elevations Unknown: An Adventure in the Heart of Borneo



	Book-Title	Summary
1031171	All Elevations Unknown: An Adventure in the Heart of Borneo	A daring twist on the travel-adventure genre that places the talented Lightner in the ranks of authors such as Jon Krakauer, Sebastian Junger, and Redmond O ;Hanlon, All Elevations Unknown is ultimately the remarkable story of two ...

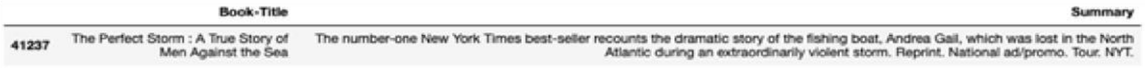
<u>Books recommended to the user based on similarity between their summaries;</u>

1. Last Chance to See



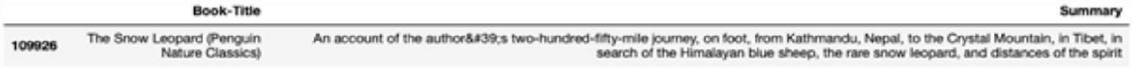
	Book-Title	Summary
41622	Last Chance to See	The authors provide an account of their journey around the world in search of endangered animals, including the kakapo of New Zealand, white rhinos in Zaire, and the Komodo lizard

2. The Perfect Storm : A True Story of Men Against the Sea



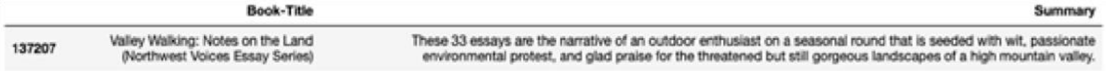
	Book-Title	Summary
41237	The Perfect Storm : A True Story of Men Against the Sea	The number-one New York Times best-seller recounts the dramatic story of the fishing boat, Andrea Gail, which was lost in the North Atlantic during an extraordinarily violent storm. Reprint. National ad/promo. Tour. NYT.

3. The Snow Leopard (Penguin Nature Classics)



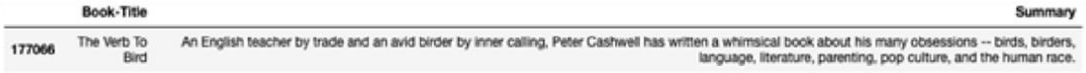
	Book-Title	Summary
109926	The Snow Leopard (Penguin Nature Classics)	An account of the author&#39;s two-hundred-fifty-mile journey, on foot, from Kathmandu, Nepal, to the Crystal Mountain, in Tibet, in search of the Himalayan blue sheep, the rare snow leopard, and distances of the spirit

4. Valley Walking: Notes on the Land (Northwest Voices Essay Series)



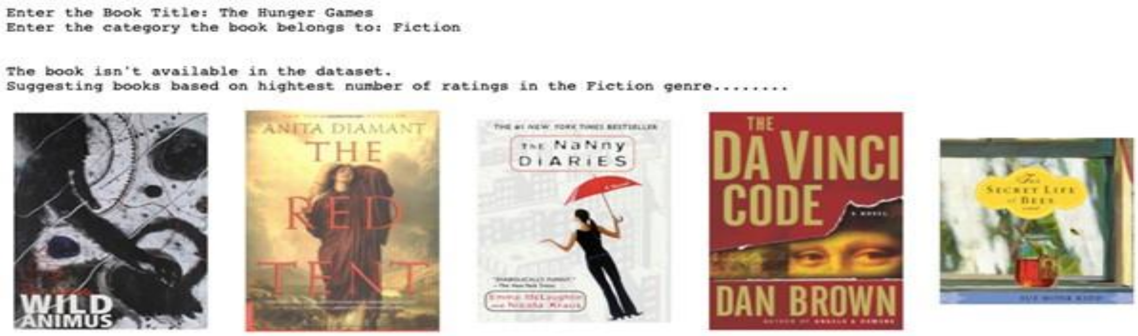
	Book-Title	Summary
137207	Valley Walking: Notes on the Land (Northwest Voices Essay Series)	These 33 essays are the narrative of an outdoor enthusiast on a seasonal round that is seeded with wit, passionate environmental protest, and glad praise for the threatened but still gorgeous landscapes of a high mountain valley.

5. The Verb To Bird



	Book-Title	Summary
177066	The Verb To Bird	An English teacher by trade and an avid birder by inner calling, Peter Cashwell has written a whimsical book about his many obsessions -- birds, birders, language, literature, parenting, pop culture, and the human race.

Taking a glance at the summaries of all the books, it does look like they may be similar as they all belong to the same genre but the guesswork and glancing at the summary isn’t really the best way to arrive at a conclusion about the quality of recommendation made by the system.



```
Enter the Book Title: The Hunger Games
Enter the category the book belongs to: Fiction

The book isn't available in the dataset.
Suggesting books based on hightest number of ratings in the Fiction genre........
```

Output when the book title isn’t present in the dataset

2.2 Content Based Filtering: Bag of Words Approach

The last filtering technique focuses only on the summary column. The summary describes the book in two to three sentences. The underlying question remains: Are two sentences enough to capture the essence of a book? The obvious answer to the question is NO.
Therefore, the aim was to build a better book profile (item profile) and then use cosine similarity on these profiles.

This approach uses the **RAKE algorithm**. RAKE stands for **R**apid **A**utomatic **K**eyword **E**xtraction, as the name suggests it is a keyword extraction algorithm. It is independent of domain knowledge and simply removes stopwords from raw text. It generates candidate expressions. It is different from TF-IDF of words or TF-IDF of n grams in the sense that it does not need a lot of effort or meticulous logic for implementation.

RAKE works in a very simple yet efficient way. It first converts all the words in text to lowercase, for example it will convert MACHINE to machine, USML to usml and so on. It then tokenizes the raw text using delimiters. For example,

I am a graduate student at the Northeastern University and I study Data Science at the Northeastern University.

will be converted to

[‘i’, ‘am’, ‘a’, ‘graduate’, ‘student’, ‘at’, ‘the’, ‘northeastern’, ‘university’, ‘and’, ‘i’, ‘study’, ‘data’, ‘science’, ‘at’, ‘the’, ‘northeastern’, ‘university’, ‘.’]

The array is then split into sequences of words by delimiters or position of stopwords. The words in a sequence are assigned the same position in the text and are considered a candidate keyword.
[‘i’,
‘am’,
‘a’,
‘graduate’,

‘student’, [‘graduate’, ‘student’]
‘at’,
‘the’,
‘northeastern’,
‘university’, [‘northeastern’, ‘university’]
‘and’,
‘i’,
‘study’,
‘data’,
‘science’, [‘study’, ‘data’, ‘science’]
‘at’,
‘the’,
‘northeastern’,
‘university’, [‘northeastern’, ‘university’]
‘.’]
**stopwords marked in yellow

So the candidate words for the example are:
[graduate, student]
[northeastern, university] * 2 [study,
data, science]

The candidate words are then converted to keyword scores using a matrix which is very similar to a document term matrix.

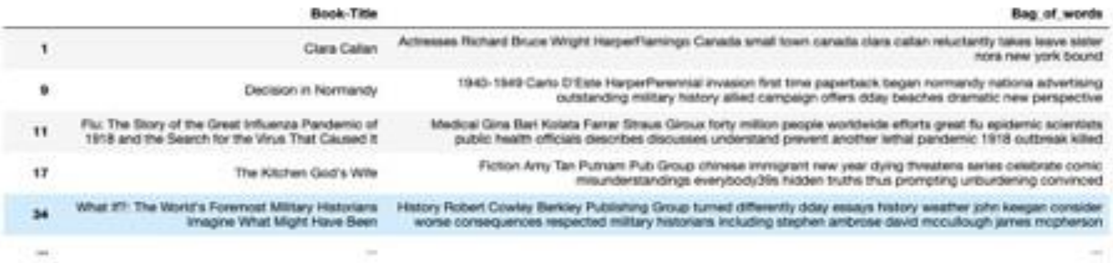
	graduate	student	northeastern	university	study	data	science
graduate	1	1					
student	1						
northeastern			2	1			
university			1	2			
study					1	1	1
data					1	1	1
science					1	1	1

Then, it defines word frequencies, word degrees and keyword scores.

- **Word frequencies** are the number of times a word appears in all candidate keywords. For example, the word frequency of graduate = 1 and that of university = 2.
- **Word degree** is the sum of each row. For example, the degree of graduate = 2 and that of university = 3.
- **Keyword Score** is defined as word frequency by word degree. For example, the keyword score of graduate = ½ and that of university = 2/3.

Steps:

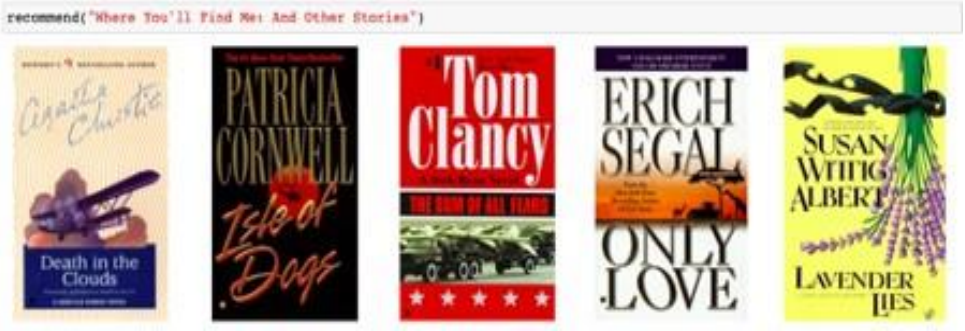
1. Convert the Summary column to Key_words column after extracting the key words using the RAKE algorithm.
2. Convert the rest of the columns containing book features (Book-Author, Publisher, Language, Category) to lists.
3. Append all the lists to make a Bag_of_words column that contains all the features of the book.



	Book-Title	Bag_of_words
1	Clara Callan	Actresses Richard Bruce Wright HarperFlamingo Canada small town canada clara callan reluctantly takes leave sister nora new york bound
9	Decision in Normandy	1940-1949 Carlo D'Este HarperPerennial invasion first time paperback began normandy nations advertising outstanding military history allied campaign offers dday beaches dramatic new perspective
11	Flu: The Story of the Great Influenza Pandemic of 1918 and the Search for the Virus That Caused It	Medical Gina Bari Kolata Farrar Straus Giroux forty million people worldwide efforts great flu epidemic scientists public health officials describes discusses understand prevent another lethal pandemic 1918 outbreak killed
17	The Kitchen God's Wife	Fiction Amy Tan Putnam Pub Group chinese immigrant new year dying threatens series celebrate comic misunderstandings everybody39s hidden truths thus prompting unburdening convinced
34	What If?: The World's Foremost Military Historians Imagine What Might Have Been	History Robert Cowley Berkley Publishing Group turned differently dday essays history weather john keegan consider worse consequences respected military historians including stephen ambrose david mccullough james mcpherson
...	...	...

4. Create a word count matrix using the CountVectorizer from the sklearn library. This creates a document term matrix.
5. Calculate cosine similarity again using the sklearn library on the word count matrix.

6. Recommend books that are the most similar to the input book based on cosine similarity.



```
recommend("Where You'll Find Me: And Other Stories")
```

Recommended books using bag of words approach

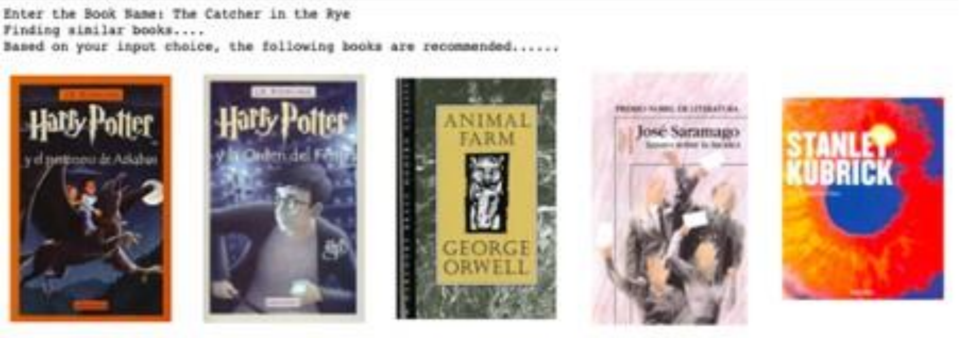
This approach is different from the last one, which was only based on the summary of the book. This approach uses additional features of the book like its author, publisher, category and language. This helps build a better and more descriptive item profile and hence is a good replacement of the first approach.

2.3 Collaborative Filtering using KNN

This filtering technique uses similarity between books using ratings given by the readers. The underlying assumption of this technique is that ‘**a reader gives similar ratings to similar books**.’ This approach uses NearestNeighbors from the sklearn library. To keep up with the computing efficiency of the system, only books with at least 100 ratings and users who have rated at least 100 books are considered.

Steps:

1. Create a user-item matrix with users as columns and books as rows
2. Convert the dense matrix to a sparse matrix. Machine learning algorithms perform the best on sparse matrices.
3. Construct the kNN model and fit it on the sparse matrix.
4. Take a book title as input from the user, find its nearest neighbors from the kNN model fitted on the sparse matrix and return the nearest neighbors as recommendations to the user.



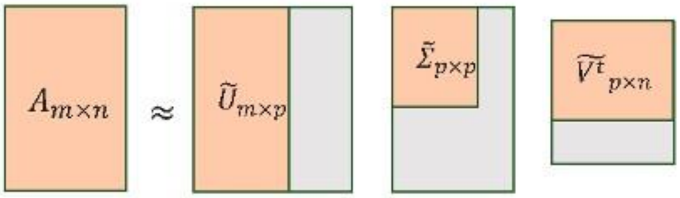
```
Enter the Book Name: The Catcher in the Rye
Finding similar books....
Based on your input choice, the following books are recommended......
```

Recommended books using KNN

2.4 Collaborative Filtering using Matrix Factorization

The matrix factorization technique ‘factorizes’ the user-item matrix into rectangular matrices of lower dimension. In particular, this approach uses Singular Value Decomposition. SVD first performs dimensionality reduction on the user-item matrix and transforms it from N dimensions to k dimensions, where k is less than N.

The dimensions of the user-item matrix are reduced using truncated SVD. Truncated SVD is different from the conventional SVD in a way that it does not centre the data before computing SVD and hence can work with sparse matrices efficiently.



Truncated SVD illustration

Steps:

1. Construct the user item matrix where the users are columns and books are rows.
2. Fit the truncated SVD model (from the sklearn library) on the data.
3. Compute the correlation of the truncated SVD fit matrix.
4. Find similar books based on the correlation matrix and recommend the top 5 books to the user.

2.5 <u>Collaborative Filtering using Surprise package</u>

Surprise package stands for ‘Simple Python Recommendation System Engine’. it is a recommendation system library and is used as one of the scikit series. It is a simple technique which is easy to use and supports a large variety of recommendation algorithms such as basic algorithm, collaborative filtering, matrix decomposition and many more.

The surprise package provides us with a variety of prediction algorithms, like matrix factorization (SVD, KNN, SVD ++, BaselineOnly) , baseline algorithm and neighborhood method.

It also has the tools for evaluating, analyzing, and comparing the performance of different algorithms. Using a powerful iterator such as Cross Validation, it gives detailed results and evaluation for a set of parameters. These cross validation programs can be run very easily and efficiently.

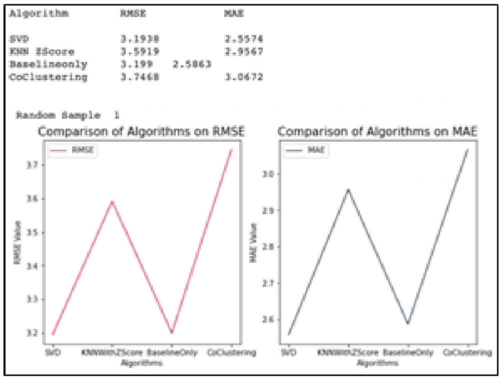
The following tasks are are performed using Scikit-Surprise Package:

1. Comparing the performance of different algorithms like SVD, KNNWithZScore, BaselineOnly, CoClustering with different samples but same data size
2. Comparing the performance of different algorithms like SVD, KNNWithZScore, BaselineOnly, CoClustering with different data sizes
3. Recommendation using the SVD algorithm, using Matrix Factorization techniques.It is usually a more effective technique as it discovers the latent features and the underlying interactions between users and items.

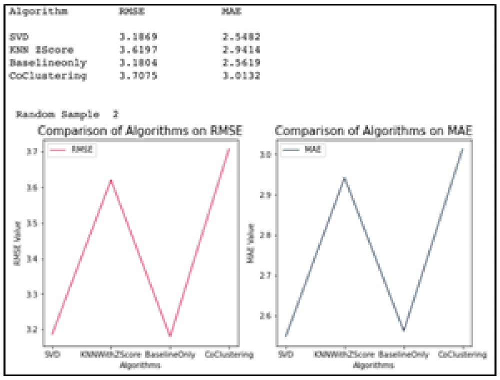
Evaluation

1. <u>Comparing algorithms based on different samples but same data size</u>

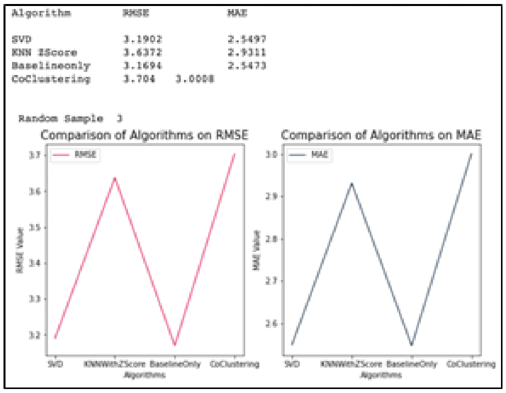
   The Plots below demonstrates the comparison between RMSE(root-mean-square error) and MAE (Mean Absolute error) calculated for different algorithms for random sample**s**



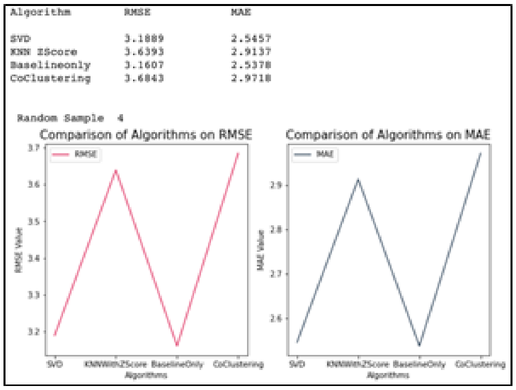
```
Algorithm          RMSE               MAE

SVD                3.1938             2.5574
KNN ZScore         3.5919             2.9567
Baselineonly       3.199    2.5863
CoClustering       3.7468             3.0672


 Random Sample  1
```



```
Algorithm          RMSE               MAE

SVD                3.1869             2.5482
KNN ZScore         3.6197             2.9414
Baselineonly       3.1804             2.5619
CoClustering       3.7075             3.0132


 Random Sample  2
```



```
Algorithm          RMSE               MAE

SVD                3.1902             2.5497
KNN ZScore         3.6372             2.9311
Baselineonly       3.1694             2.5473
CoClustering       3.704    3.0008


 Random Sample  3
```

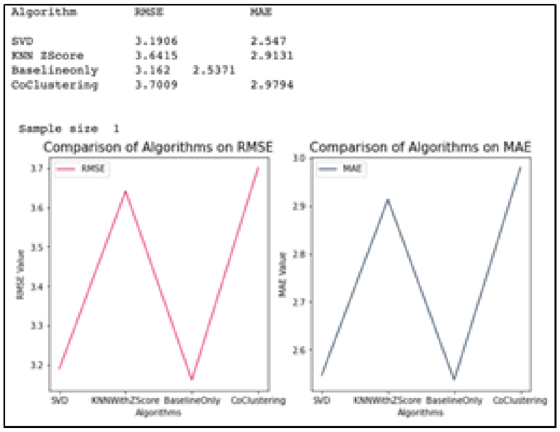


```
Algorithm          RMSE               MAE

SVD                3.1889             2.5457
KNN ZScore         3.6393             2.9137
Baselineonly       3.1607             2.5378
CoClustering       3.6843             2.9718


 Random Sample  4
```

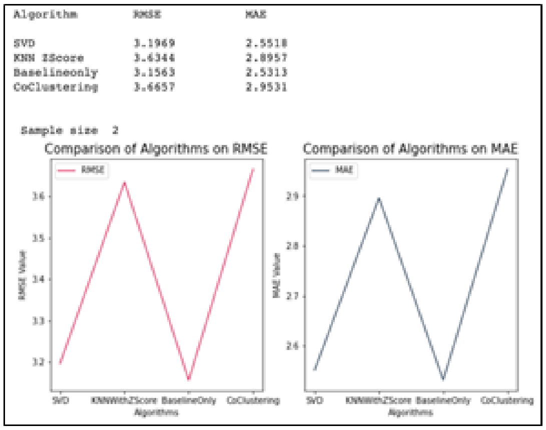
We can see that SVD and BaseOnly algorithms perform better in comparison to other algorithms, but it is unclear which one performs best.

2. <u>Comparing algorithms based on different data sizes</u>

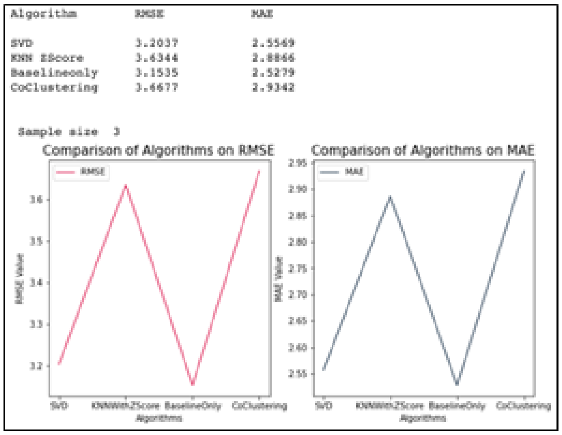
To get more detailed insight, we compared the algorithm’s performance on different dataset sizes.



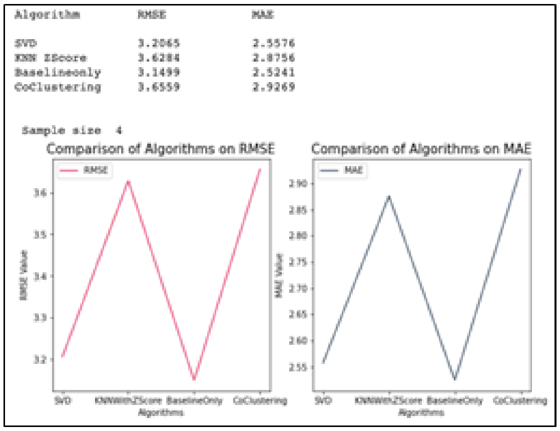
```
Algorithm          RMSE              MAE

SVD                3.1906            2.547
KNN ZScore         3.6415            2.9131
Baselineonly       3.162   2.5371
CoClustering       3.7009            2.9794


 Sample size  1
```



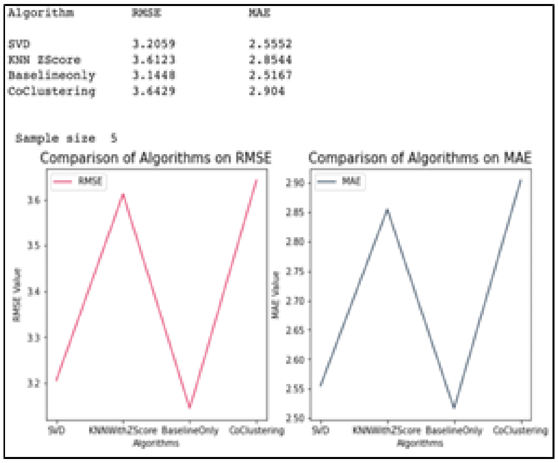
```
Algorithm          RMSE              MAE

SVD                3.1969            2.5518
KNN ZScore         3.6344            2.8957
Baselineonly       3.1563            2.5313
CoClustering       3.6657            2.9531


 Sample size  2
```



```
Algorithm          RMSE              MAE

SVD                3.2037            2.5569
KNN ZScore         3.6344            2.8866
Baselineonly       3.1535            2.5279
CoClustering       3.6677            2.9342


 Sample size  3
```



```
Algorithm          RMSE              MAE

SVD                3.2065            2.5576
KNN ZScore         3.6284            2.8756
Baselineonly       3.1499            2.5241
CoClustering       3.6559            2.9269


 Sample size  4
```



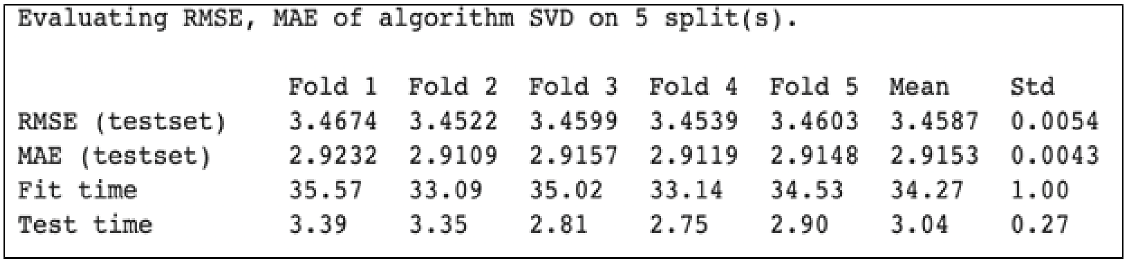
```
Algorithm          RMSE              MAE

SVD                3.2059            2.5552
KNN ZScore         3.6123            2.8544
Baselineonly       3.1448            2.5167
CoClustering       3.6429            2.904


 Sample size  5
```

3. Recommendation system using Surprise package

Steps :

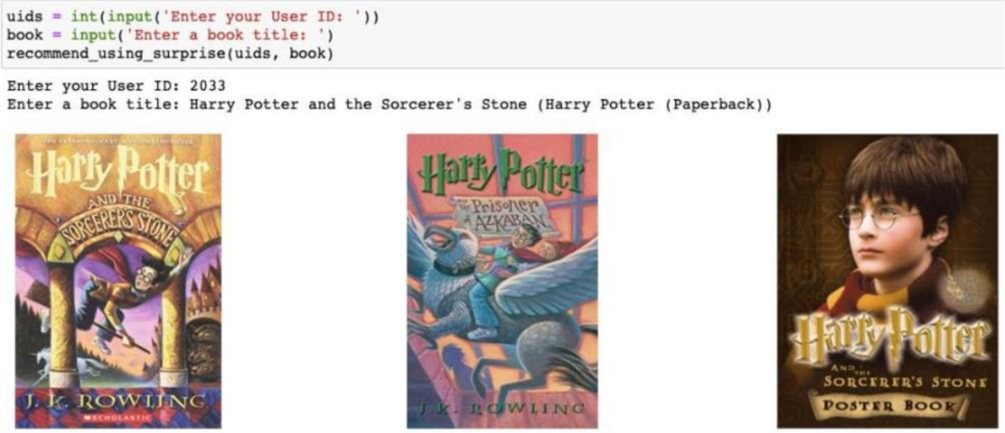
1. Load the rating dataset using load_from_df() method, we will also need a Reader object, and the rating_scale parameter must be specified, which is between 1-10 for our dataset. Note : The data frame must have three columns, corresponding to the user ids, the item ids, and the ratings in this order. Each row thus corresponds to a given rating.

2. Cross validating the model using 5 fold cross validation, reporting accuracy and computation time



```
Evaluating RMSE, MAE of algorithm SVD on 5 split(s).

                  Fold 1  Fold 2  Fold 3  Fold 4  Fold 5  Mean    Std
RMSE (testset)    3.4674  3.4522  3.4599  3.4539  3.4603  3.4587  0.0054
MAE (testset)     2.9232  2.9109  2.9157  2.9119  2.9148  2.9153  0.0043
Fit time          35.57   33.09   35.02   33.14   34.53   34.27   1.00
Test time         3.39    3.35    2.81    2.75    2.90    3.04    0.27
```

3. After fitting the model, we can check the predicted score of, for example, user-id “276709” on Book’s ISBN using the predict method
4. To provide the recommendation for users, we iterate through all books and check the estimated rating for the book.
5. We select the book with highest rating and recommend that to the user.



```
uids = int(input('Enter your User ID: '))
book = input('Enter a book title: ')
recommend_using_surprise(uids, book)
```

```
Enter your User ID: 2033
Enter a book title: Harry Potter and the Sorcerer's Stone (Harry Potter (Paperback))
```

Collaborative filtering using the Surprise package gives us the advantage of selecting the best parameters for algorithms by cross-validating the data, which saves a lot of time when the data set is large. When implementing SVD algorithms we used 5 fold validation that runs 5 tests on data and gets the best parameters. Since, the surprise package gives us only the estimated ratings that the user might give to that book.

Conclusion :

A book recommendation system using various filtering techniques and similarity metrics was designed. Content based systems and collaborative systems worked in different yet interesting ways. While a content based systems's main focus is on the similarity between the book features', a system based on collaborative filtering takes into account the user's past behaviour.

In Content-based filtering we get top 5 books based on cosine similarity between the books, this method is very efficient as it considers the summary column as a feature of the books along with the title of the book which explores the keywords which we would have missed if we only used category column only.

We considered users who have given more than 100 ratings and Books that contain more than 100 ratings for further algorithms. Collaborative Filtering gives us more insight in user-book relationships. We explore the relationship between rating given by a user to a book to recommend books to users based on all ratings data we have and we recommend highly rated books read by similar users.

The five techniques used in the project were different from each other and sometimes also yielded different outputs for the same book as their algorithm was different. A way we could have improved this system more would have been making the recommendation system's input case insensitive. Sometimes our machine wasn't able to handle the enormous amount of data and therefore it had to be trimmed.

A few concepts were particularly very fascinating such as Truncated SVD that works efficiently with sparse matrices, the RAKE algorithm that is used to extract keywords from raw text and the surprise package used to implement recommendation systems. We tried to blend in a few techniques and built a recommendation system that doesn't necessarily work only the "Book Crossing Dataset' but can be used on any dataset, even to recommend movies or news articles.

References

[1] https://towardsdatascience.com/my-journey-to-building-book-recommendation-system-5ec959c41847

[2] Pasquale Lops, Marco De Gemmis, and Giovanni Semeraro. “Content-based recommender systems: State of the art and trends.” In: Recommender systems handbook. Springer, 2011, pp. 73–105.

[3] https://www.analyticsvidhya.com/blog/2021/06/build-book-recommendation-system-unsupervised-learning-project/

[4] https://towardsdatascience.com/breaking-down-goodreads-dataset-using-python-388e9b9d6352

Dataset

[1] https://www.kaggle.com/peternwill/book-crossing-goodread-best-book

Appendix

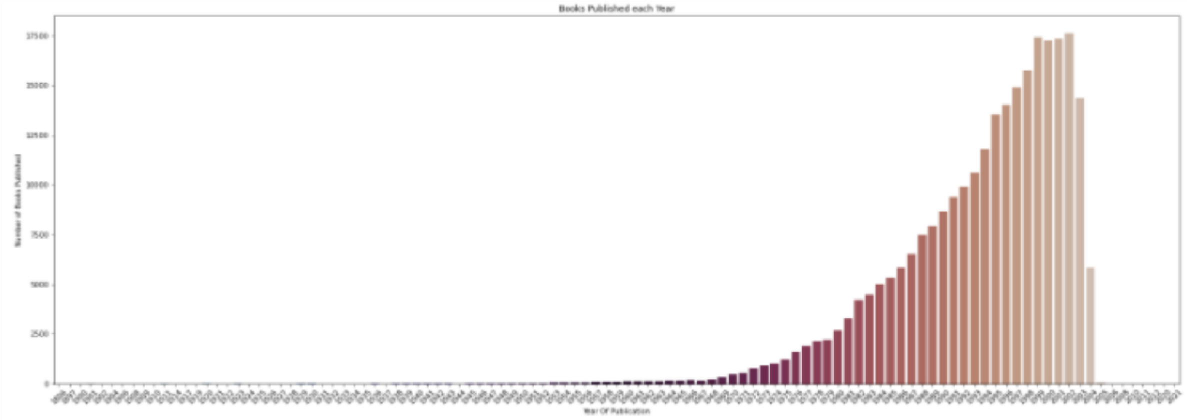


Figure 7

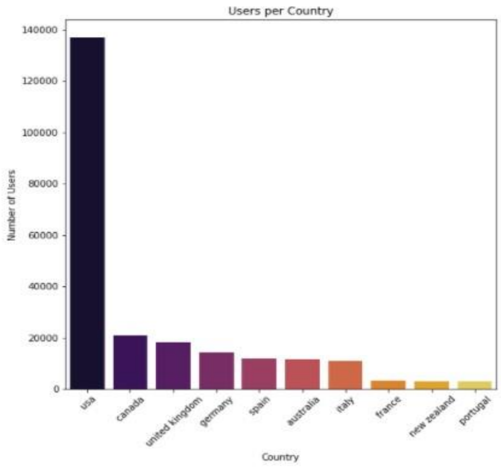


Figure 8

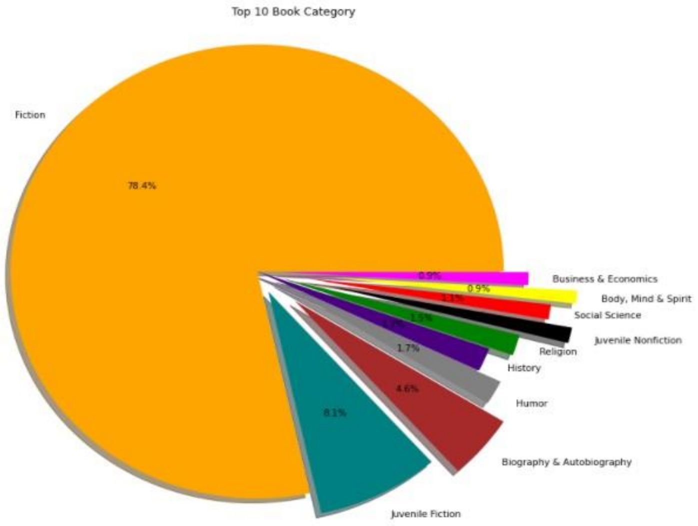


Figure 9