Book Recommendation System

Akshay Adlakha
University at Buffalo
akshayad@buffalo.edu

Nirjhar Agarwal
University at Buffalo
nirjhara@buffalo.edu

***Abstract*—Recommendation systems attempt to predict the preference or rating that a user would give to an item. Knowledge discovery techniques can be applied to the problem of making personalized recommendations about items or information during a user's visit to a website. Collaborative Filtering algorithms give recommendations to a user based on the ratings of other users in the system. In the proposed framework, prediction using item based collaborative filtering is combined with prediction in an adaptive weighted scheme. The proposed solution will be scalable.**

I. INTRODUCTION

In this project, We are working on a Recommendation Engine which helps to get the similar items to users or items to similar users to recommend. This is based on collaborative filtering. Today we see on any E-commerce websites, we look for some products and we start getting recommendations of similar items. So, this gives us the curiosity to know how they recommend items to other users. The idea came to us because we see this everywhere but don't really see it. When we started this course on machine learning, it opened a whole new world of algorithms. Anything we do as simple as netflix or Amazon apply the recommendation system and then we decide to make a project in it to see the inner mechanisms for it.

The project has two different independent measurements (User based and Item based) performed on the entire dataset in which a recommendation is made based on User(Its personal preference) or Item(Specific rating given to a book).

The prediction of books is calculated using cosine similarity for both the User based and Item based filtering.

$$Cos\theta = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\| \, \|\vec{b}\|} = \frac{\sum_1^n a_i b_i}{\sqrt{\sum_1^n a_i^2} \sqrt{\sum_1^n b_i^2}}$$

where, $\vec{a} \cdot \vec{b} = \sum_1^n a_i b_i = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n$ is the dot product of the two vectors.

User-User Collaborative Filtering

The method identifies users similar to other users in terms of rating a particular entity. If a user has a certain rating that is similar to the rating of another user then we can recommend that other user’s item to the aforementioned user.

Item-Item Collaborative Filtering

The method applies items similar to other items based on the User rating. If a particular user has a certain rating for a particular item then the system would recommend items based on similar rating(or higher rating depending upon the rating) to the user.

II. PROCEDURE

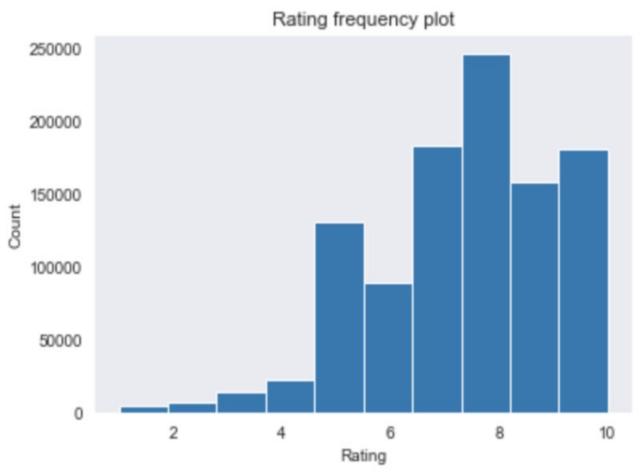
A. Data Fetch

There are three input files(Users, Books, Book-Ratings), all of them in csv format. The files are loaded into python using pandas and are merged using the primary key. The final table contains UserId, Book rating, Location, Author, Publisher columns etc.

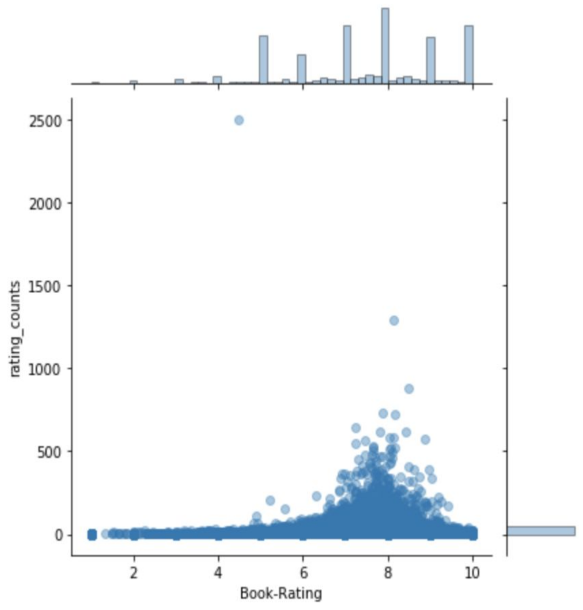
B. Data Manipulation

The data is loaded into python from three different csv and merged into one dataframe and after that pivot table is used in both User and Item filtering to calculate recommendations.

EDA plots:



This is the plot of the number of times a particular rating 1-10 is given by the users.



This plot shows the total sum of each book rated by the users and how many times an individual rating (1-10) given by the users.

The NA values are replaced by 0 in the table for calculation purposes.

C. Calculating Recommendations

The recommendation for User is calculated using cosine similarity metric in both the filtering User-based and Item-based. Recommendation books are based on the query given by the user. Results are derived from the similarity matrix in both the filtering methods.

D. Prediction using Cosine Similarity

Cosine similarity is a metric used to determine how similar the Users are or Items are.

Mathematically, it measures the cosine angle between two vectors projected in a multi-dimensional space. In this context, the two vectors are arrays containing the ratings of books.

The data has been filtered to include books which have been rated more oftenly because books which have been rated only once will not matter much to the overall engine and it becomes easier for the system to compute the pivot table as there are thousands upon thousands of records to link with each other.

When plotted on a multi-dimensional space, where each dimension corresponds to a ratings, the cosine similarity captures the orientation (the angle) of the ratings and not the magnitude.

Let us take an example we have two sets as following

A = [2, 1, 1, 2, 1, 0]

B = [1, 1, 1, 1, 1, 1]

We can apply cosine similarity on the above sets as following

$$Cos(A,B) = \frac{(2*2 + 1*1 + 1*1 + 2*1 + 1*1)}{\sqrt{(2*2 + 1*1 + 2*2 + 1*1 + 1*1)}*\sqrt{(2*2 + 1*1 + 1*1 + 1*1 + 1*1 + 1*1)}}$$

=> 9/9.9498 =0.9045

We are calculating cosine angle between two vectors projected in a multi-dimensional space.

E. User Interface

For our user interface we used the TKINTER library available in python. It is a robust and platform independent toolkit. It has a wide variety of features that lets you handle almost anything. It is implemented as a python wrapper around TCL interpreter embedded in python interpreter. On a broad level it contains Windows, Widget, Frame, Child and Parent. Widget can be anything ranging from buttons, checkboxes, labels etc.

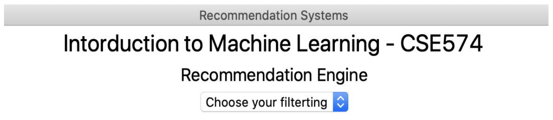
The widget is created in four stages namely Create, Configure, Pack and Bind.

In our project we have used some of the features from the tkinter library like Window, frame, optionmenu, buttons, dropdowns, entry, label etc.

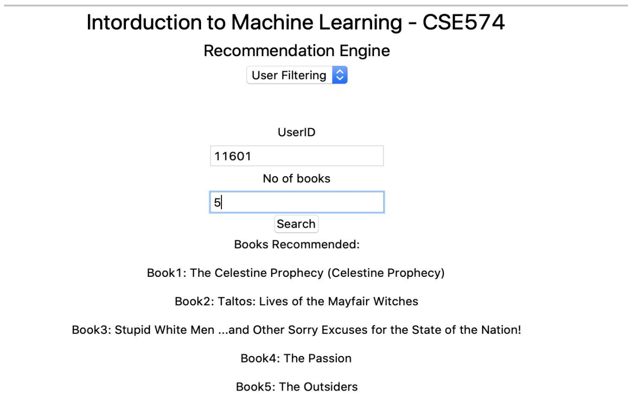
III Results

Following results achieved in building this recommendation system.

This is the main screen where the user gets landed the first time. It asks the user to select the type of filtering.



After making the selection the user is asked to input relevant details for eg. if the user selects a User based filtering then it would simply have to enter UserId and number of books. The system would then compute the User-User filtering and compare the particular user to other users and based on that the books will be recommended to that user.



User Filtering

UserID: 11601

No of books: 5

Books Recommended:

Book1: The Celestine Prophecy (Celestine Prophecy)

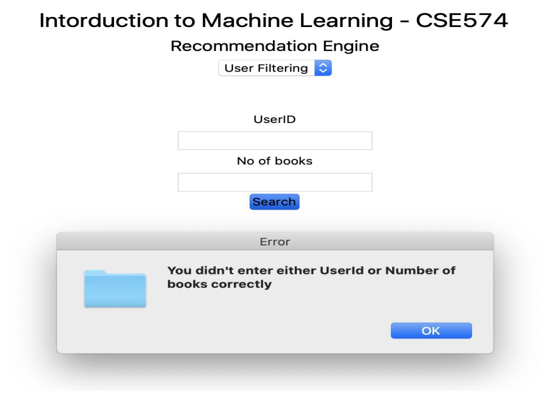
Book2: Taltos: Lives of the Mayfair Witches

Book3: Stupid White Men ...and Other Sorry Excuses for the State of the Nation!

Book4: The Passion

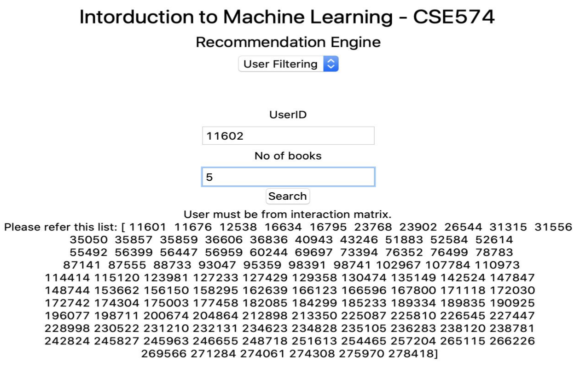
Book5: The Outsiders

System can take a maximum of books. If a user forgets to enter values or tries to enter any negative value or a value greater than 10, then gets an error.



You didn't enter either UserId or Number of books correctly

If a user enters a UserId which is not in the pivot matrix then gets a message and will be asked to enter the UserId from the specified list.



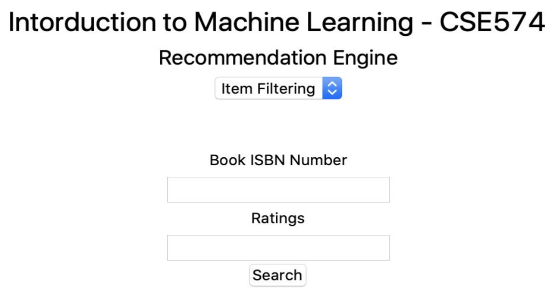
User Filtering

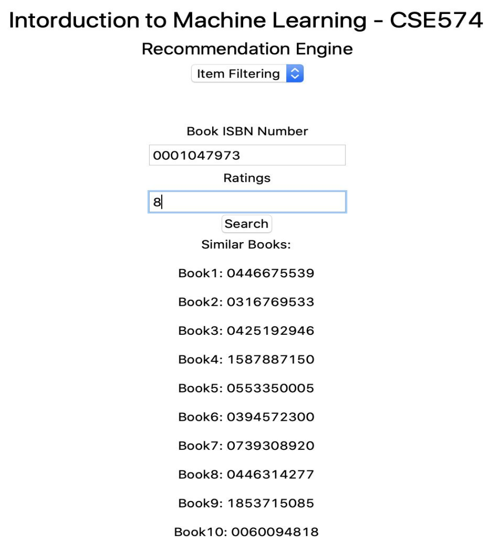
UserID: 11602

No of books: 5

```
User must be from interaction matrix.
Please refer this list: [ 11601  11676  12538  16634  16795  23768  23902  26544  31315  31556
 35050  35857  35859  36606  36836  40943  43246  51883  52584  52614
 55492  56399  56447  56959  60244  69697  73394  76352  76499  78783
 87141  87555  88733  93047  95359  98391  98741 102967 107784 110973
114414 115120 123981 127233 127429 129358 130474 135149 142524 147847
148744 153662 156150 158295 162639 166123 166596 167800 171118 172030
172742 174304 175003 177458 182085 184299 185233 189334 189835 190925
196077 198711 200674 204864 212898 213350 225087 225810 226545 227447
228998 230522 231210 232131 234623 234828 235105 236283 238120 238781
242824 245827 245963 246655 248718 251613 254465 257204 265115 266226
269566 271284 274061 274308 275970 278418]
```

If the user selects Item based filtering then the system will ask for the ISBN no of the book and what rating it wants to give to that particular book and based on that rating the system will recommend ten other similar books.




Intorduction to Machine Learning - CSE574

Recommendation Engine

Item Filtering

Book ISBN Number

0001047973

Ratings

8

Search

Similar Books:

Book1: 0446675539

Book2: 0316769533

Book3: 0425192946

Book4: 1587887150

Book5: 0553350005

Book6: 0394572300

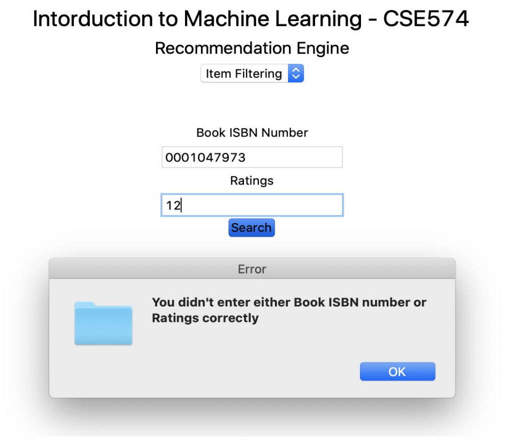
Book7: 0739308920

Book8: 0446314277

Book9: 1853715085

Book10: 0060094818

If a user forgets to enter values or tries to enter a negative rating or greater than 10, then will get an error.


Intorduction to Machine Learning - CSE574

Recommendation Engine

Item Filtering

Book ISBN Number

0001047973

Ratings

12

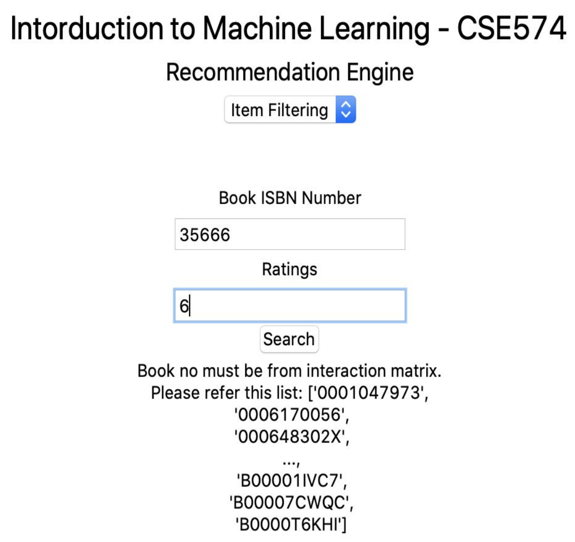
Search

Error

You didn't enter either Book ISBN number or Ratings correctly

OK

If a user tries to enter the ISBN number which is not in the pivot matrix then gets a message to enter the ISBN from the specified list.


Intorduction to Machine Learning - CSE574

Recommendation Engine

Item Filtering

Book ISBN Number

35666

Ratings

6

Search

Book no must be from interaction matrix.
Please refer this list: ['0001047973',
'0006170056',
'000648302X',
...,
'B00001IVC7',
'B00007CWQC',
'B0000T6KHI']

User Filtering:

Input - User Id, No of Books(1 - 10)

Output - Recommendations of specified number of books.

Item Filtering

Input - ISBN(book code), Rating(user given).

Output - Recommendations of 10 books.

IV CONCLUSION

We have successfully implemented the Book Recommendation system using collaborative filtering in both User to User and Item to Item and ran the program through various testing phases and have concluded that it is working properly.

V SCOPE

The algorithm for collaborative filtering can also be implemented using other methods like apriori algorithm. The concept of deep learning can be theoretically applied to the recommendation system which can improve scalability that can handle much more data than currently possible. This is still a topic of debate among many big companies as they are continually trying to perfect the algorithm so they can specifically target their audience in various demographics.

.

VI CONTRIBUTION

We distributed the work based on the background and interests. Akshay started with collecting dataset and did exploratory data analysis. We had two types of filtering in our project. We both understood their part and Akshay did user-user filtering and Nirjhar worked on item-item filtering. We divided the work of building the UI.
Nirjhar implemented the functionality of dropdown selection. Akshay worked on the querying part i.e. text label and search functionality. We wrote the report together through google docs.

We were constantly helping each other wherever we got struck. We worked effectively and efficiently. We enjoyed working on this project.

We were asking the queries we had during the project from TA Tiehang Duan. He was very helpful and answered our queries.

VII References

1. https://towardsdatascience.com/introduction-to-recommender-systems-6c66cf15ada
2. https://towardsdatascience.com/recommender-systems-in-practice-cef9033bb23a
3. https://tryolabs.com/blog/introduction-to-recommender-systems/
4. https://medium.com/recombee-blog/machine-learning-for-recommender-systems-part-1algorithms-evaluation-and-cold-start-6f696683d0ed
5. https://docs.python.org/3/library/tkinter.html
6. https://scikit-learn.org/stable/modules/generated/sklearn.metrics.pairwise.cosine_similarity.html
7. https://pypi.org/project/sklearn-pandas/1.5.0/