PROJECT REPORT

ON

BOOK RECOMMENDER SYSTEM (Readers Guide)

BY

Abasiekeme Attang

**READER’S GUIDE**

**NON-TECHNICAL SUMMARY**

The Reader’s guide is a book recommender system for novel and book readers that uses the book crossing dataset and implements algorithms in other to make recommendations on what books to read next .The book recommender system data set which contains a lot of users and the books they read and their ratings .This is an efficient way of making users be able to know what to read next based on the similarity of fellow users or by how the books are related to one by how they rated the books for example If they are similar users for example they always rate the same things to users observing the one one has not rated will make it easier for one

.Recommender System are new era web instrument that help client in exploring through data on the web and get data identified with their inclinations. Albeit more often than not recommender systems are connected in the range of web based shopping and stimulation spaces like motion picture and music, their pertinence can be developed different fields. With the web blast individuals are progressively utilizing on the web sites to pick the best arrangement to go

through their cash with a broadly accessible rundown of creators and classes. A great deal of sites are additionally offering books as one of the items and a few destinations are committed sites for online book shopping. In this aggressive market, heaps of sellers are utilizing distinctive methodologies to draw in clients. Proposal systems is one such innovation that help client by furnishing them with pertinent data in view of their profile. This project is an endeavor to speak to the Recommender Systems which are at present working in the area of study guides.

**WHY READER’S GUIDE**

Recommender systems plays a vital role in e commerce and subscriptions. It is known to increase profitable as we see the increase in profits of sites like amazon and Netflix .I can’t count how many times i have had to buy a product on amazon based on recommendations and i see that makes a lot of sense and I thought about a practical way to apply programming with

Machine learning applications and I figured that would make sense.

**DATA**

The data used for the project was the book crossing dataset and was downloaded from this website here it contains three csv files which is used to user book rating dictionary with that .

**Preprocessing**

I had to preprocess text data in other to make a text vector which will be efficient in performing similarities between users and similarities between books .When running the coding i used in encoding =’ISO-8859-1’ because i was running into problems opening the CSV files then replaced the necessary to be able to split my data in order to get a dictionary where user

, books -> ratings and made a function that transforms ratings.

**Methods**

In order to implement the book recommenders systems I used the following implementations tools:

1. User-Based collaborative filtering

2. Item based collaborative filtering

**User-Based Collaborative Filtering**

This algorithm provides recommendations of books by finding similar users. This is often harder to scale because of the dynamic nature of users. In the user-based approach the algorithm produces a rating for an book i by a user u by combining the ratings of other users u’ that are similar to u. Similar here means that the two user's ratings have a high Pearson correlation or Euclidean distance or cosine similarity.

**User-Based Algorithm**

For every book i that u has no preference for yet

For every other user v that has a preference for i

Compute the similarity between u and v

Ad v’s preference for i , weighted by s to a running average

Return the top books ranked by the weighted average.

**Item-Based Collaborative Filtering**

This algorithm provides recommendations by calculating the similarities between the books and providing the most similar books to one another based on their ratings .

**Item-Based Collaborative Filtering**

For every book i that u has no preference for yet

For every book j that u has a preference for

Compute similarity between i and j

Add u’s preference for j weighted by s to a running average

Return the top books ,ranked by their weighted average.

**SIMILARITY MEASURES**

Similarity is the measure of how much indistinguishable two objects are. It is generally portrayed as a separation with measurements speaking to elements of the items. On the off chance that the separation is little it will be high level of similarity while expansive separation will be low level of similarity. Similarity are measure in the range[0,1] term vectors.

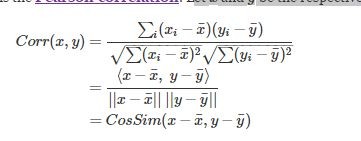
For this project I implemented Euclidean distance and Pearson correlations.

**Euclidean Distance**

This is the distance between two paths of the length connecting them. In most cases when people said about distance, they will refer to Euclidean distance. Euclidean distance is also known as simply distance. When data is dense or continuous, this is the best proximity measure.

**Pearson Correlations**

Pearson correlations uses formula like below

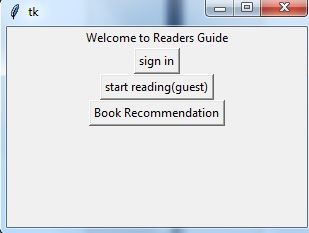


**DESIGN**

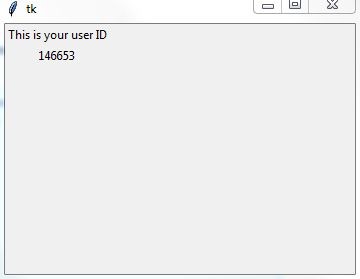
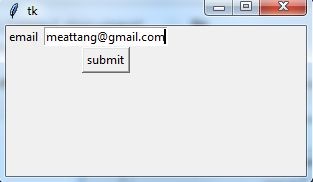
This project was implemented using python and using a GUI using tkinter .Tkinter gives a fair representation of implementation and will be used to implement user-based and item based collaborative filtering

For **user- based recommendation** i created an allocation system which when you sign in assumes gives you a user ID this was done by generating a random user from the users registered in the database this is done to avoid key errors when inputting for recommendation and with that you input the ID to get the recommendation.

**Login page**

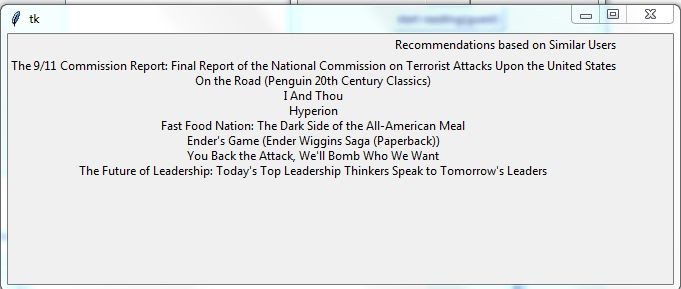


**Sign in submit Button**



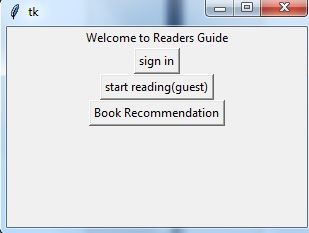
**To get user-based recommendation**

You go back to the home page and click “book recommendation” and we see the recommendation

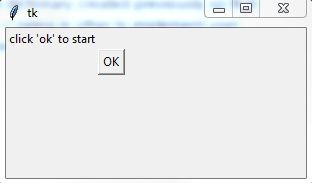


**For Item Based** I assumed the users where guest so be able to distinguish the item based from the user based .So for this i had to flip the dictionary created previously so that instead of user , book -> rating it became book ,user -> rating in other to implement user based ratings so a function was made for that purpose.So below shows how it is used.

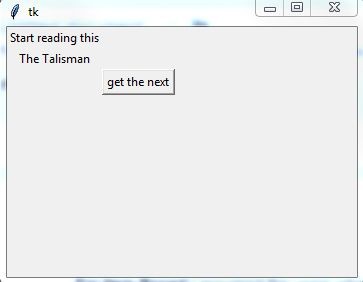
**Looking you click “start reading”**



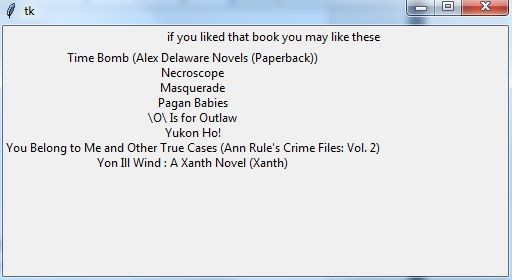
Then it goes to this for you too click ok to start



When you click this it generates a random book for you to start reading and this is done show efficiency of the system that recommendations are done random and can be any book.



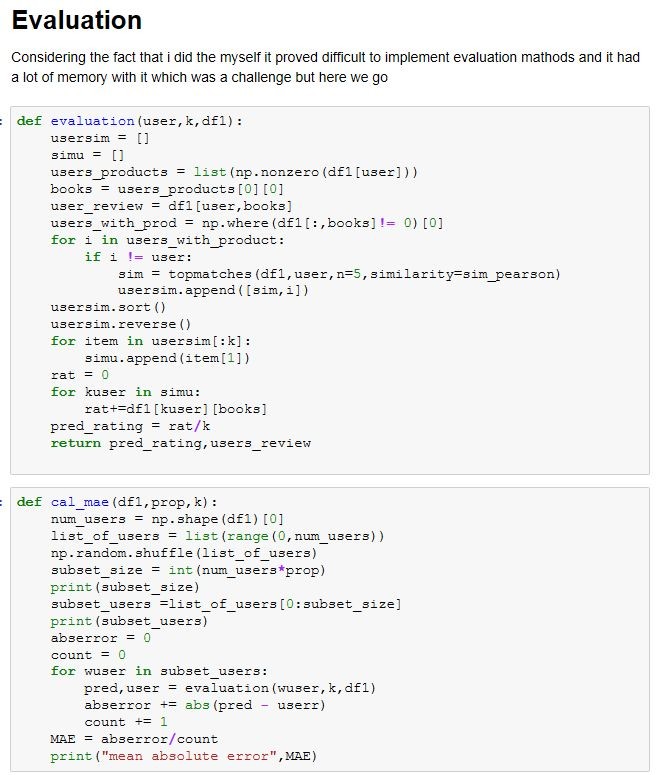
Then the recommendations are provided and this is how item based is done



The Item based provided 8 similar books to read if you like that one and gave it the highest rating so it therefore finds the most similar books and presents it in a format .

During the evaluation there was a lot of constraints with this project .I encountered a lot of errors and which made it difficult to be able to make errors.But i used MAE at first i got an

Average MAE of 2.1 then i encountered errors after that



**ANALYSIS**

**Assumptions**:

We assume that the ratings were accurate and didn’t exceed 10.

**Weaknesses**

Due to time constraints and workload I could not implement matrix factorization. But there are limitations when it comes to the methods .For user-based it can't handle sparse datasets and that poses as a problem and can't be too effective with real time data and scalability poses a challenge for user-based .For item based it poses the same challenge due to the wide list

**Strength**

I was able to apply the algorithms and prepare a good result based on that i was able to

Apply user-based and item-based and presented a nice GUI which improved upon can serve a good purpose .

**Challenges**

During this project I encountered some challenges while coding:

> Due to the large dataset I found out my code was a taking a long time to run

> The evaluation took a long time to run and still gave me errors.

**Future Works**

In the future i hope to do some matrix factorization using gradient descent and being able to make it function in real time.

**Conclusion**

The significant objective of this project is to give differing suggestion to the user regardless of the possibility that there are less ratings around a thing for a specific user. This project can be connected to different sectors too like business analytics for example being able to know your target audience to promote a product . Encourage the framework can be reached out in other measurement that incorporates fusing logical data, taking various ratings and giving a more adaptable proposal that could go into various areas too. The execution of the framework and its examination with the other comparative systems is planned for future continuous work.And i'm grateful being able to learn a lot about programming data mining applications.