Recommender System Choice Report

Team Cautious Spoon

Researched Recommender Systems

We decided to investigate five types of recommender systems which are the most used in the media and entertainment industry for recommending content that users would want and would apply in a similar way to books as it would with films and other digital media.

(Neighbourhood-based) Collaborative recommender system

This system is the most commonly implemented and works by collecting ratings and finding users with similar interests based on ratings. It assumes that users who have previously had similar interests will carry on doing so in the future.

Pros:

- No domain knowledge needed
- Can help the user find a new interest
- Simple to implement

Cons:

- Cold-start problem – cannot handle new items
- Hard to include data from outside the item

Content-based recommender system

This system works by filtering information. Based on the objects a used has rated, it will find features from these, for example the author or genre from a book, and find similar books with the same features. This often works by recommending based on past objects a user has liked and ones that the user is also looking at.

Pros:

- Doesn’t need data about users other than the one getting the recommendation
- Very specific for a user, finds niches other users might not be interested in

Cons:

- Requires a lot of domain knowledge
- Hard to add future interests a user may have

Demographic based recommender system

This system groups users based on demographics and is popular due to its simplicity. Generally, research is done with a survey in a location, and similar to collaborative systems, groups of people are found. However, different data is used.

Pros:

- Easy to implement
- Does not require user ratings

Cons:

- Users may get stereotyped
- Not fully specific to a user

Utility based recommender system

This system makes a suggestion based on a utility generated for a user through a custom function and can factor non-product attributes such as availability. This would be helpful if the book club system was for a library (digital or physical) where books can only be given to one club at a time.

Pros:

- Can factor in non-product attributes

Cons:

- Harder to implement

Knowledge based recommender system

This system suggests recommendations based on a user's preferences. It establishes a relationship between an item and user by looking at how the item is used. For example, how long a book is read for and how many times.

Pros:

- Does not require user ratings
- Finds best similar recommendations even with limited options to recommend based on criteria

Cons:

- Cold-start problem – can't predict when a user has no reviews
- Needs access to query a user makes

Tested Recommender Systems

From the five researched recommender systems, we chose 2 main types to try out and test. Both also have multiple algorithms that can be used allowing us to get a wide range of testing to establish which exact system will be the best.

Content based recommender system

This system was chosen as it is very well suited for a book, being able to use details of them book such as its title, author, publisher, and genre category. As it is also very specific to a user it will be very helpful to give books that a user is guaranteed to want to read.

For this system, two different methods were tested based on the different information that a book contains. One method uses a 'combined' property that takes into account the title, author, publisher, and category of each book, whereas the other will use the 'summary'. Both approaches use cosine similarity based on the frequency of words within these properties. Additionally, the combination of both properties was also tested and used.

To test these algorithms a database of 31,000 ratings and 500 books was used. Although this does provide initial insight into the performance of each algorithm, ideally, they would have been tested against a much larger set of data and/or online testing used. However, this was not possible, and testing was limited by the computational power of the computers available.

Algorithm	MAE	RMSE
	Mean Absolute Error	Root Mean Squared Error
	Lower values mean better accuracy	Lower values mean better accuracy
Content KNN – Combined property	3.4035	3.9326
Content KNN - Summary	3.5182	3.9726
Content KNN – Using both Summary and Combined property	3.5318	3.9800
Random	3.8021	4.3834

Ultimately, the content based KNN algorithm using only **the combined property** was chosen. As well as having the best MAE and RMSE values this algorithm seemed to result in the best recommendations, providing a larger range of books, which avoided giving the exact same recommendations for similar books.

Neighbourhood based recommender system

For this system, seven different methods mentioned below were tested against 40,000 ratings to get the Mean Absolute Error and Root Mean Squared Error to establish which algorithms work the best (give the least errors).

Half of the algorithms tested look at the ratings from a User-based approach (giving recommendations based off similar users) whereas the others look at it from an Item-based approach (giving recommendations based off similar books). KNN is used which separates the data into clusters and relies on item feature similarity rather than making assumptions about the underlying distribution. The User and Item approaches are then broken down into the Cosine (where the cosine of an angle between two vectors is used), MSD (Means Squared Displacement – where the distance moved over time is used), and Pearson (using the Pearson Correlation Coefficient approach to measuring linear correlation) approaches. We also tested a random approach which randomly gave a book without using any specific approach. As expected, this had a much higher RMSE value but did not give a too worse off MAE value.

We believe that 40,000 ratings is a suitable number to test as it gives us a wide range but also makes sure that it does not have a too bad performance when using the evaluator tool.

The Mean Absolute Error metric is where we work out the absolute (positive) difference between the ratings given and what we expect to get the error margins which are then summed and divided by how many ratings we tested, in our case 40,000. The lower the value means that there is less error, therefore greater accuracy.

The Root Mean Squared Error metric is like MAE but helps be more accurate in working out the total error by penalising those ratings that have a large error and for those that only have a small error, minimises the penalty. Instead of getting the absolute difference, the differences are squared (giving positive values again), which makes large errors have an inflated value. The squared differences are then summed and divided by the total ratings and finally square rooted to get back to a familiar value.

Algorithm	MAE	RMSE
	Mean Absolute Error	Root Mean Squared Error
	Lower values mean better accuracy	Lower values mean better accuracy
User Cosine KNN	3.6687	3.9332
User MSD KNN	3.6306	3.9595
User Pearson KNN	3.6692	3.9247
Item Cosine KNN	3.6399	3.8936
Item MSD KNN	3.5769	3.8768
Item Pearson KNN	3.6620	3.8991
Random	3.6752	4.2585

The **Item MSD KNN** system was chosen as it has the lowest values for both the RMSE and MAE tests.

Implemented Recommender Systems

From the recommender systems tested, we decided to use both two types as they can be used in different places in our book club system due to the type of recommendation they give.

Content based recommender system

This was chosen because it focuses on the details of the book itself and recommends books like the one the user is currently viewing. Therefore, we have chosen to implement this system on the page where a user can search for books and view the details. Similar books will be shown at the side of the page that would apply to any user regardless of their interests but the most suitable for a user will be shown. So, for instance, if a user was looking at a book from a genre they don't usually read from, their recommendation will not be totally influenced by the genre they usually read, but instead similar books will be shown with more of a focus on similar publishers and authors.

Neighbourhood based recommender system

This system will be implemented when a club has a meeting. Based on the random chosen member, some books will be shown that this user would probably like to read based on the users' characteristics, taking into consideration the characteristic of the club (such as its genre category) so that it is suitable for all members of the club.

This system will also be used on the dashboard to suggest books that the user may like to read.

The Item MSD KNN algorithm was used which means that recommendations will be closely linked to the book itself that was previously read, making it suitable to find similar books for the whole group. This means that it is also a good way to recommend books to users generally and features on the main dashboard.