Book Recommender System: Matrix Factorization via Singular Value Decomposition

Data Science Capstone Project - BrainStation

Angel Wang

7/18/2021

Problem statement

Reading has made such a great positive impact on my life. As a reader, I would love to help others discover amazing experiences through reading. As a data scientist, I use tools more efficient and exploratory than simply browsing through each book on the shelves. I believe readers find certain books enjoyable not just because of who the author is or what genre it belongs to. To quantify those abstract features, I choose matrix factorization algorithm to build my book recommender system. The hope is this system can reduce the amount of time people spend looking for enjoyable books, and therefore use positive feedback to encourage people to spend more time reading.

Background

Recommender systems have been widely used on platforms including streaming services, social media, and e-commerce. They offer tailored recommendations to each individual according to their own taste. The users can find the items they like in a shorter time, and the platform can profit more from more transactions or just interactions. With a growing number of users and items, data science is needed to support this system and optimize it with intensive calculation.

Currently, there are two types of recommender systems. The content-based filtering system collects demographic data and tries to conclude the features of items that each user reacts positively to (e.g. the genre of books, the starring of movies), and it makes recommendations solely based on each user's own data. The collaborative filtering system vectorizes the user's feedback on various items, find the users that have similar taste, and recommend one user to items the other user like.

The collaborative filtering system can then be classified to memory-based collaborative filtering and model-based collaborative filtering, and matrix factorization belongs to the latter. It factorizes the interaction matrix into two lower dimensionality matrices and uses them to generate back the interaction matrix to predict the unknown values.

This family of methods became widely known during the Netflix prize challenge. Its effectiveness was reported by Simon Funk in his blog post in 2006, and the prediction results can be improved by assigning different regularization weights to the latent factors based on items' popularity and users' activeness using singular value decomposition.

Source of data

The data used in this project was collected in late 2017 by the lab of Julian McAuley, UCSD from www.goodreads.com. All the data is in compressed JSON format. Since the entire dataset is very large, it was divided by genre and this project used the Poetry genre. Three datasets were available for each genre's data and this project used two of them: Interactions and Books.

Each row in Interactions represents a review or rating performed by a user on a book. Each column is a feature of the interaction, including user id, book id, review id, rating, timestamps, etc.

Each row in Books represents a unique book. Each column is a feature describing the book, including title, ISBN, publisher, popular shelves, etc.

Preprocessing

EDA shows the majority of the rating column has value 0. However, it's found that 0 is just a placeholder when no rating score is provided. Therefore, all the 0 ratings are removed from the dataset before modeling, as they will be treated the same as other unknown values in the interaction matrix. This step removes 1.5 million rows from the total 2.7 million rows in Interactions.

EDA also illustrates that a small number of users contributed to most of the ratings, while the majority of users don't rate frequently. The same pattern applies to books: a small portion of books are rated many times, while most of the books are rarely rated. This pattern indicates that we should filter out the less popular books and less active users in data cleaning to release the stress on calculation and improve the quality of prediction. The 1.2 million interactions remaining from the last step contain 36,182 books and 267,821 users. After the filtration, there are 133,080 interactions left, containing 2,474 books and 2,917 users.

EDA is also performed on some other features to learn about the dataset more, but after feature selection, only 3 features are saved for modeling: user id, book id, rating. Another 2 features describing books are also saved in order to provide more information when making the recommendation: title and link.

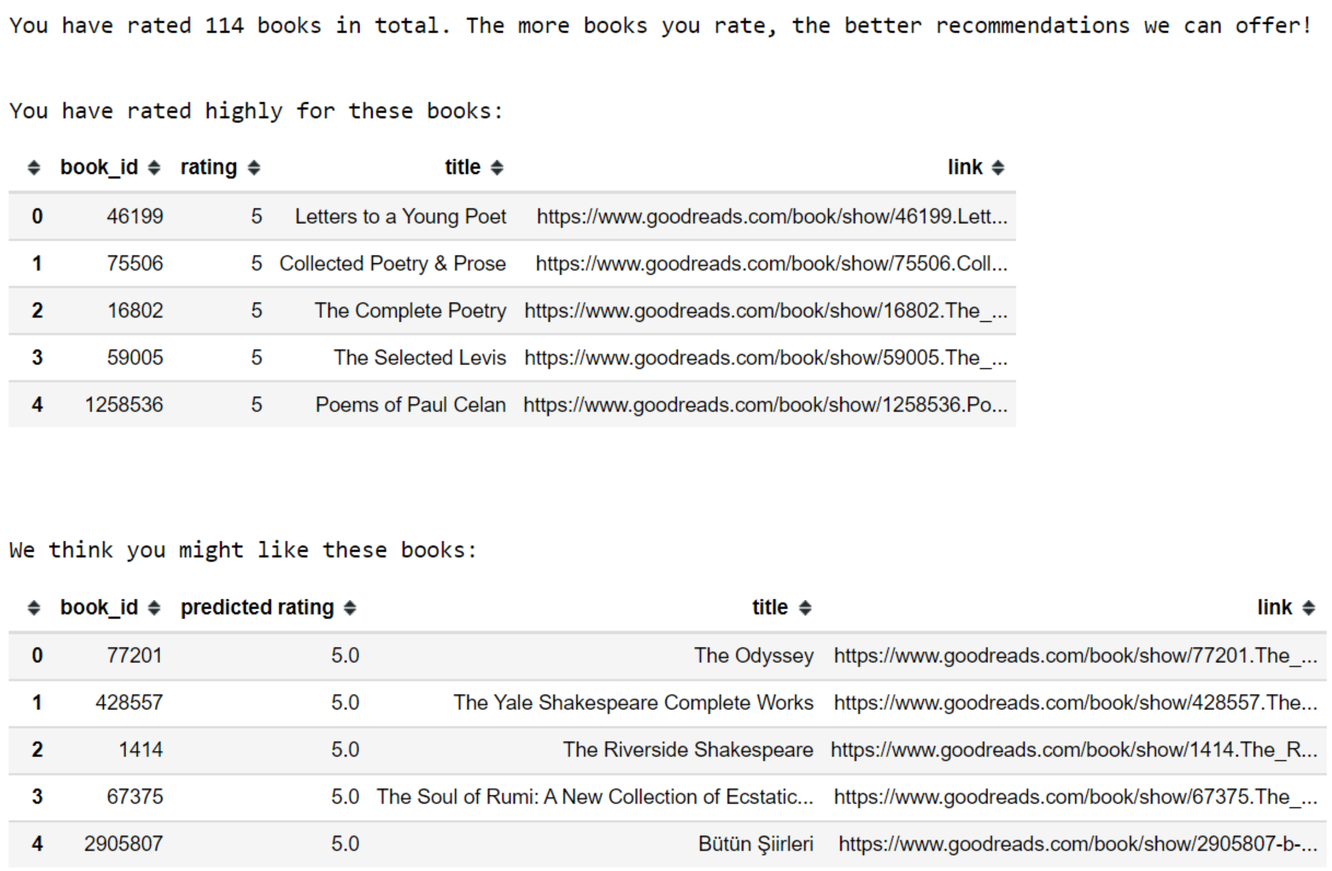
Modeling

Using the surprise package, the data is first split into the train set and the test set. The SVD model is fitted on the train set and then gives predictions on the test set.

	SVD model	Base rate (by average)	Base rate (by mode)
RMSE	0.8100	0.8419	1.0066
MAE	0.6269	0.6472	0.6363

The model is then optimized by hyperparameter tuning through visualization and grid search. The surprise package is designed to use all the known values as the train set at this step with cross-validation, and predict for all the unknown empty values as "test set". For this test set, the RMSE is 0.5681 and the MAE is 0.4504. However, these numbers should not be interpreted similarly as before, since they are calculated against the average rating across the entire dataset. The FCP score is also calculated to be 0.5991. Further performance evaluation of the recommender system should be done by tracking the usage, and how many interactions the recommendations prompt.

The predictions made by the model can give a list of recommended books to each user. If the user has interaction records in the train set, the recommendations will be calculated based on personal preference. If the user is new to the system, a list of the most popular books will be recommended.



```
You have rated 114 books in total. The more books you rate, the better recommendations we can offer!
```

```
You have rated highly for these books:
```

	book_id	rating	title	link
0	46199	5	Letters to a Young Poet	https://www.goodreads.com/book/show/46199.Lett...
1	75506	5	Collected Poetry & Prose	https://www.goodreads.com/book/show/75506.Coll...
2	16802	5	The Complete Poetry	https://www.goodreads.com/book/show/16802.The_...
3	59005	5	The Selected Levis	https://www.goodreads.com/book/show/59005.The_...
4	1258536	5	Poems of Paul Celan	https://www.goodreads.com/book/show/1258536.Po...

```
We think you might like these books:
```

	book_id	predicted rating	title	link
0	77201	5.0	The Odyssey	https://www.goodreads.com/book/show/77201.The_...
1	428557	5.0	The Yale Shakespeare Complete Works	https://www.goodreads.com/book/show/428557.The...
2	1414	5.0	The Riverside Shakespeare	https://www.goodreads.com/book/show/1414.The_R...
3	67375	5.0	The Soul of Rumi: A New Collection of Ecstatic...	https://www.goodreads.com/book/show/67375.The_...
4	2905807	5.0	Bütün Şiirleri	https://www.goodreads.com/book/show/2905807-b-...

Conclusions

The recommender system uses the SVD model to implement matrix factorization algorithm. The optimal model with minimum error can make personalized recommendation to help people find books they are likely to enjoy.

Applications and Next Steps

For business applications, this recommender system can be used for more than just books. With properly identified users and items, and rating scores between them, the system can give predictions and recommendations for any desired scenario.

To improve on the quality of recommendations, the potential next step is to build a hybrid system, including demographic filtering, content-based filtering (NLP), and other approaches. For user experience, an interactive user interface can be constructed, where users can put in their user id to receive a list of recommendations.