Only for You: Relevant Board Game Suggestions from BoardGameGeek.com

Sanni Isokirmo[1], Ahti Katiska[1], and Miika Mäki[1]

[1]*University of Helsinki*

October 24, 2025 4:04pm

Introduction

Play—coming together to engage in a common recreational activity—is a human universal [1] and essential part in human development [2, 3]. Board games are a popular and accessible form of play that bring family, friends and strangers alike together, and promote well-being across the life-span [4, 5].

But what if a group of people have outplayed the games they own and would like to find some new ones? There are at least 150.000 different board games out there [6] so finding a good match can be a time consuming process. A service that would suggest new board games based on the individual taste could come in handy on such occasions.

This technical report will outline the steps we made and lessons learned to create a recommendation system for new board games based on the user's ratings in BoardGameGeek.com (BGG) database as well as the technical details of setting up a webpage *LINK TO WEBSITE* through which users can find recommendations for new board games. To achieve this, we used non-negative matrix factorization (NMF) with a selected users' review scores to create tailored suggestions for new board games.

Data

We chose the biggest board game ratings database BoardGameGeek.com to fetch user ratings and board game metadata, such as playing mechanics and category. As it is openly available, it having a data access was simple and there were no data privacy issues to be addressed.

Since there are 2.7 million users and 150.000 boardgames in BGG [6, 7], fetching all of them would not be feasible with the API interface, at least within this project. Since many users have left no or few reviews, the data would be too sparse if we had taken a random sample. To get a compromise between training matrix sparsity, time constraints and selection bias, we decided to choose x guilds (online discussion groups) that represent different geographic regions, target audiences (teens, parents, seniors), genres as well as general and special interest groups. This way we could get X reviews from Y reviewers about Z games. The sampling strategy did bias our distribution towards active users, but also allows for reliable training. Users with only few reviews would make the estimation computationally intensive and possibly unreliable.

We decided to load our data via the BGG XML API [8]. While the data is well preprocessed and clean, getting the data was no simple task.

First, the API has rate limits that we had to find out by trial and error. We used a base delay of .75 seconds and if the rate was limited, increased the delay exponentially and waited for the maximum of one minute. The maximum batch size for games was 20 and it took some time to find out why the game metadata coverage was so low on larger test runs.

Second, since the game metadata fetching was unreliable even with the exponential backoff time, we used a game metadata cachefile to make sure that if a game metadata had been fetched successfully once, it was stored on the project home directory.

Finally, we included only users with at least 25 reviews and fetched the most popular games first to avoid training data matrix sparsity. Our final dataset of users was quite small, since user discovery is a nontrivial problem.

Methods

Using review scores is the single best way of generating recommendations [9]. It seems counterintutive as one would think that categories, genres, sales or other available metadata would be equally helpful. Based on this empirical finding and the scope of the project, we chose to focus solely on the review scores.

The method we chose to use for data analysis was non-negative matrix factorization (NMF), which is an unsupervised learning algorithm [10]. We chose NMF as the basis of our recommender system because it is a well known method for such use.

We combined the usernames and game names into a matrix with the corresponding ratings from each user and imputed NaN's with SoftImpute. For the NMF model we also had to calculate the optimal rank, which we did by iteratively running multiple NMF models (Selitetään miten meidän defaulttiin päädyttiin.). With the calculated default optimal rank we then ran the NMF model with sklearn's implementation and the resulting recommendation matrix giving us estimates on how the user may rate games they haven't rated yet. We used these estimations to show the user the top games they haven't rated yet as well as the actual estimated rating they might give them.

We create a item-user matrix from the reviews. By the simple fact that the quantity of boardgames far surpass the number of users, the matrix is very sparse. In our first version, we imputed the missing values with zeros. This lead to the NMF biasing the rating estimations heavily towards 0. We decided to impute the missing values with fancyimpute packages SoftImpute, which is a SVD based imputation algorithm.We used scikits nmf implementation.

We selected the NMF rank hyperparameter via testing multiple rank values, and comparing the root mean square error of the models. In our final version the rank of the NMF was 25. We did not tune the SoftImpute hyperparameters due to timeconstrainst, but this and other aspects to improve upon are mentioned in the discussion.

Regarding the UI we kept it very simple with a Boostrap table to show the details for each game and a Chart.js bar table to show the rating distribution within our dataset. Otherwise no other visualization methods were used in our project.

Results

The website can be found on http://ec2-13-60-174-144.eu-north-1.compute.amazonaws.com/. The user can write their, or someone elses username in the field, the website lists usernames that match the query which are already available in our scraped database. If the user proceeds with a username that is not in our scraped data, a warning is shown, that their BGG user must have at least 1 review, and that the computation will take time. Given the lack of computational power on the EC2 instance the app is running on, the recalculation of the recommendation matrix takes multiple minutes. The user is then directed to a page where they can view their top recommendations.

The model (NMF) in testing had a RMSE of 0.267 on our dataset, which given that the values are from 1-10, is an acceptable error rate.

Discussion

We created a board game recommendation webpage based on a sample of 200.000 board game reviews from 1000 active BoarGameGeek.com users to help BoardGameGeek users—or anyone—find new boardgames that match their taste. Below we will reflect on the lessons learned from our project.

We found it interesting how simple it is to create a recommendation algorithm that works fairly well. It is reassuring that one can first build something that works and then make it more complex. That is essentially what we did, and will definitely use a similar routine for future data science projects.

Setting up API was a valuable learning experience. If we were to redo this, it would probably be better to rely on a ready made data dump. This is because the rate limits were very restrictive and getting the data in tidy format involved quite a bit of work. If this project was expanded, we would use the whole BGG database as a training data to get even better estimates. For the scope of this mini-project, 200,000 reviews should be enough.

Generally the project ended up being less complex than we had wished, though we didn't exactly depart from our initial plan. The final state of the project quite close to the minimum viable product we were expecting to get. Mainly we weren't able to implement the ideas we for non-users being able to also get recommendations. Other ideas we had included more visualization as well as using the categories in some meaningful way. Ultimately this was mainly due to scheduling problems as well as partly due to setting such a clear division for work tasks. Perhaps if we had found time to work as a team rather than waiting on each person to finish their indivitual tasks, we could've progressed more steadily and had time for other additions.

The application as it stands, does not validate the SoftImpute results, and probably with correct tuning of the SoftImpute hyperparameters, the NMF portion of the algorithm stack could entirely be removed. While researching the options of handling the missing values in the item-user matrix, we also considered using a weighted NMF. However this didn't have an actively maintained library on python, so it would have required us to create the implementation ourself.

Regarding the data scraping process, now in retrospect it would have probably been easier to scrape reviews via games, not via users. since the games are more easily discoverable.

We also learned how using everyone's strengths can contributed into a great product. One of us had solid experience from front end and boardgames; another from creating appealing user interface and keeping repositories organized; yet another from statistics and managing projects more generally. Combining these abilities allowed us to have a clear division of labor among the group and make a steady progress with weekly checkpoints.

Other than using a larger training data, future work could try out different recommendation algorithms and optimize them with more rigor.

This project serves as a good example that when the problem definition is clear cut and the product creates true added value for the user, it is easy to manage a complex workflow like ours. With a clear purpose and goal in mind, it is easier to stay on track and strive for a product that will bring added value to users.

References

[1] D. E. Brown, "Human Universals, Human Nature & Human Culture," *Daedalus*, vol. 133, no. 4, pp. 47–54, 2004, ISSN: 0011-5266. JSTOR: 20027944. Accessed: Sep. 18, 2025. [Online]. Available: https://www.jstor.org/stable/20027944.

[2] P. K. Smith, "Play: Types and Functions in Human Development," in *Origins of the Social Mind: Evolutionary Psychology and Child Development*, New York, NY, US: The Guilford Press, 2005, pp. 271–291, ISBN: 978-1-59385-103-3.

[3] A. D. Pellegrini, *The Role of Play in Human Development*. Oxford University Press, 2009, 289 pp., ISBN: 978-0-19-536732-4. Google Books: yO3zr_0iFYIC.

[4] L. Dell'Angela, A. Zaharia, A. Lobel, O. Vico Begara, D. Sander, and A. C. Samson, "Board Games on Emotional Competences for School-Age Children," *Games for Health Journal*, vol. 9, no. 3, pp. 187–196, Jun. 2020, ISSN: 2161-783X. DOI: 10.1089/g4h.2019.0050. Accessed: Sep. 18, 2025. [Online]. Available: https://www.liebertpub.com/doi/full/10.1089/g4h.2019.0050.

[5] S. Solway, "Wellness through board games," M.S. thesis, The Chicago School of Professional Psychology, 2011.

[6] WordsRated. "Board Games Statistics – WordsRated," Accessed: Sep. 18, 2025. [Online]. Available: https://wordsrated.com/board-games-statistics/.

[7] M. Didymus-True. "BoardGameGeek's support drive numbers fall for third year running -," Accessed: Sep. 18, 2025. [Online]. Available: https://boardgamewire.com/index.php/2024/01/09/boardgamegeeks-support-drive-numbers-fall-for-third-year-running/.

[8] BBG. "BGG XML API2," BoardGameGeek, Accessed: Sep. 19, 2025. [Online]. Available: https://boardgamegeek.com/wiki/page/BGG_XML_API2.

[9] D. Epstein, *Range: Why Generalists Triumph in a Specialized World*. New York: Riverhead Books, 2021, 368 pp., ISBN: 978-0-7352-1450-7.

[10] Q. Daly. "Step-by-Step NMF Example in Python," Medium, Accessed: Oct. 23, 2025. [Online]. Available: https://medium.com/@quindaly/step-by-step-nmf-example-in-python-9974e38dc9f9.

Appendix