

[◀ Return to Classroom](#)[DISCUSS ON STUDENT HUB](#)

Recommendations with IBM

REVIEW

CODE REVIEW

HISTORY

Meets Specifications

Congratulations!

The project is now complete and meets all the expectations, I hope you had a great learning time doing this project.

All the best

Code Functionality & Readability

All the project code is contained in a Jupyter notebook or script. If you use a notebook, it demonstrates successful execution and output of the code. All tests have passing remarks.

Code is well documented and uses functions and classes as necessary. All functions include document strings. DRY principles are implemented.

Data Exploration

Explore the data to understand the number of users, articles, and information about the interactions that take place.

Create Rank Based Recommendations

Tests will ensure that your functions will correctly pull the top articles. The two functions should pull the top ids and the top names.

Collaborative Filtering

Create a matrix with users on the rows and articles on the columns. There should be a 1 if a user-article interacted with one another and a zero otherwise.

Find similar users needed for user-user collaborative filtering model. Write a function that finds similar users.

Make recommendations using user-user based collaborative filtering. Complete the functions needed to make recommendations for each user.

Improve your original method of using collaborative filtering to make recommendations by ranking the collaborative filtering results by users who have the most article interactions first and then by articles with the most interactions.

Provide recommendations for new users, which will not be able to receive recommendations using our user-user based collaborative filtering method.

That's right! For new users, we should recommend using the most popular items. You correctly identified this idea and implemented it here. Good work.

Matrix Factorization

Perform SVD on user-item matrix. Provides U, Sigma, and V-transpose matrices, as well as an

Perform SVD on user-item matrix. Provides U, sigma, and V-transpose matrices, as well as an explanation of why this technique works in this case.

Split the user-item matrix into training and testing segments. Identify the users in the test set that are also in the training.

Perform assessment of the predicted vs. the actual values.

Provide a discussion about the results, as well as a method by which you could test how well your recommendation engine is working in practice.

Great job on the discussion !

I liked the fact you were also able to focus and define the metrics for this exercise Evaluation metric, Invariant metric,

 [DOWNLOAD PROJECT](#)

[RETURN TO PATH](#)