

[Return to "Data Scientist Nanodegree" in the classroom](#)

[DISCUSS ON STUDENT HUB](#)

Recommendations with IBM

REVIEW

CODE REVIEW

HISTORY

Requires Changes

5 SPECIFICATIONS REQUIRE CHANGES

Well Done !!! This is quite a good submission. However, some of the rubrics need changes. Please refer to the rubrics below.

All the best !!! We are looking forward to your next submission.

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.

You have successfully completed the dictionary associated with exploring the user-item matrix. Excellent job! You passed all of the tests to identify the number of users, articles, and explore their interactions.

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.

Nice work pulling the top rank based recommendations. Your function passed all of the tests related to pulling the top 5, 10, and 20 articles in the dataset. Good work!

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.

Nice work setting up the user-article matrix. All tests passed. Good work !!!

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.

Nicely Done !!! However, some changes are required. Currently, you are recommending all articles that a similar user has read. However, there would be many articles that would have already been read by the user. You need to get rid of these articles. You can do something like below:

```
def user_user_recs(user_id, m=10):  
    ...  
  
    INPUT:  
    user_id - (int) a user id  
    m - (int) the number of recommendations you want for the user  
  
    OUTPUT:  
    recs - (list) a list of recommendations for the user
```

Description:

Loops through the users based on closeness to the input user_id

For each user - finds articles the user hasn't seen before and provides them as recs

Does this until m recommendations are found

Notes:

Users who are the same closeness are chosen arbitrarily as the 'next' user

For the user where the number of recommended articles starts below m and ends exceeding m, the last items are chosen arbitrarily

```
...  
# Your code here  
most_similar_users = find_similar_users(user_id, user_item)  
  
recs = []  
  
user_seen_articles, _ = get_user_articles(user_id, user_item)  
user_seen_articles = set(user_seen_articles)  
  
for user in most_similar_users:  
    if len(recs) < m:  
        article_ids = set(get_user_articles(user, user_item)[0])  
        recommended_ids = article_ids.difference(user_seen_articles)  
        req_rec = m - len(recs)  
        recommended_ids = list(recommended_ids)[:req_rec]  
        recs.extend(recommended_ids)  
        user_seen_articles = user_seen_articles.union(recommended_ids)  
  
return recs
```

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.

The same issue as discussed in the previous rubric still exists in this section too. Please consider using an approach similar to the one mentioned in the previous rubric.

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

Good Job recommending the top 10 articles to a new user. However, you need to justify your choice with an answer to the question asked in this section marked in the cell, "Provide your response here."

6. If we were given a new user, which of the above functions would you be able to use to make recommendations? Explain. Can you think of a better way we might make recommendations? Use the cell below to explain a better method for new users.

Provide your response here.

Matrix Factorization

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.

Good Job calculating the matrices. However, you should provide a response to the question about the difference between the user-item matrices in this project assignment and the one shown in the lesson exercises?

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

Nice work. You have accurately pulled the training and testing ids. You have correctly identified the different values associated with the training and testing user and article ids. You also passed all of the tests. Great job.

Perform assessment of the predicted vs. the actual values.

You have created a nice visual of the training and testing accuracy values with differing numbers of latent features. Awesome Job !!! Also, you have plotted the f1 score for both training and test sets. Well Done !!!

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.

Good Job discussing your results based on the f1 score and reaching a conclusion for the number of latent features to be used. However, will it be sufficient to make good recommendations? Do you have a way to evaluate all these algorithms? Please elaborate on these topics a little more in this section.

Hint: Think about using A/B testing for evaluation.

 RESUBMIT

 [DOWNLOAD PROJECT](#)



Best practices for your project resubmission

Ben shares 5 helpful tips to get you through revising and resubmitting your project.

[▶ Watch Video](#) (3:01)

RETURN TO PATH

Rate this review