

PROJECT SPECIFICATION

Recommendations with IBM

Code Functionality & Readability

CRITERIA	MEETS SPECIFICATIONS
Code is functional and passes all tests.	All the project code is contained in a Jupyter notebook or script. If you use a notebook, it demonstrates the successful execution and output of the code. All tests have passing remarks.
Write code that is well documented and uses functions and classes as necessary.	Code is well documented and uses functions and classes as necessary. All functions include document strings. DRY principles are implemented.

Part I: Data Exploration

CRITERIA	MEETS SPECIFICATIONS
Explore the data to understand the number of users, articles, and information about the interactions that take place.	Correct values provided for variables (median_val, user_article_interactions, max_views_by_user, max_views, most_viewed_article_id, unique_articles, unique_users, to verified by correct output for the solution 1 dictionary test(t.sol_1_test(sol_1_dict)) at the end of Part 1

Part II: Create Rank Based Recommendations

CRITERIA	MEETS SPECIFICATIONS
Create ranked based recommendation model.	Tests will ensure that your functions will correctly pull the top articles. The two functions should pull the top ids and the top names.

Part III: Collaborative Filtering

CRITERIA	MEETS SPECIFICATIONS

CRITERIA	MEETS SPECIFICATIONS
Create the user item matrix.	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 zero otherwise.
Find similar users needed for user-user collaborative filtering model.	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.	Make recommendations using user-user based collaborative filtering. Complete the functions needed to make recommendations for each user.
Improve recommendations using user-user based collaborative filtering.	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.	Provide recommendations for new users, which will not be able to receive recommendations using our user-user based collaborative filtering method.

Part V: Matrix Factorization

CRITERIA	MEETS SPECIFICATIONS
Perform SVD on user-item matrix.	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.
Perform split of training and testing datasets.	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 results.	Perform assessment of the predicted vs. the actual values.
Results discussion.	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.

Suggestions to Make Your Project Stand Out!

- Move your notebook into a web application and blend your code into a class that can be easily used with a web application.

- Package your recommendation engine code to be pip installed by others.
 - Build out a content-based recommendation system using your NLP skills.
-
-