

# Week 07 - Lab Session Results

February 12, 2025

## 1 Evaluation of Recommender Systems

Based on the same dataset used on previous weeks, let us evaluate the Collaborative Filtering (CF) model implemented last week.

Recall that `reviewerID` corresponds to user, `asin` corresponds to item, and `overall` is the user-given rating to the item.

### Exercise 1

Based on the user-based neighborhood model that was created last week, let's make a general system that can be used to generate recommendations for all users and items. The system would take into account the mean rating of each user. We can use Scikit-Surprise for this. <https://surprise.readthedocs.io/en/stable/index.html>

Use cosine as similarity measure and try to vary the (maximum) number of neighbors to take into account when predicting ratings. Set the random state to 0 for comparable results. Keep Scikit-Surprise's default settings for all other parameters.

Is it better to use 1 or 10 neighbors? You should determine this based on the Root Mean Square Error (RMSE) over 3-fold cross-validation.

Number of neighbors with lowest validation RMSE: 10

### Exercise 2

#### 2.1

Fit the neighborhood-based model defined in exercise 1 on the full training set with cosine as similarity measure and either 1 or 10 neighbors based on what you found to be better in exercise 1. Keep Scikit-Surprise's default settings for all other parameters, but set the random state to 0 for comparable results.

Use the model to predict the unobserved ratings for the users in the training set. Remove predictions for users that are not in the test set (`test_df`).

How many predictions are there and what is the average of all the predictions (rounded to 2 decimal places)?

*Note:* there may be items in the test set that are not present in the training set; these items are not included in counting the number of predictions

There are 32 users in the unobserved ratings set that are not in the test set.  
Number of predictions: 52988  
Average of predictions: 4.73

## 2.2

Report the RMSE of the rating prediction of users and items in `test_df` (rounded to 3 decimal places).

Note that the documentation [https://surprise.readthedocs.io/en/stable/predictions\\_module.html](https://surprise.readthedocs.io/en/stable/predictions_module.html) defines `r_ui` as the true rating of user  $u$  for item  $i$ , but this can be somewhat misleading, as it depends on the input. If you run the prediction based on the anti-testset of the training set, then it won't have access to the true rating and instead use the mean rating of all users over all items, which then subsequently lands in the prediction class.

RMSE for neighborhood-based Collaborative Filtering: 0.304

## Exercise 3

Define a general method to get the top- $k$  recommendations for each user, based on the rating predictions obtained in Exercise 2.1.

Print the top- $k$  with  $k = \{5, 10, 20\}$  recommendations for the user with ID `ARARUVZ8RUF5T` and its estimated ratings. Round the printed estimated ratings to 2 decimal places.

Neighborhood based Collaborative Filtering:

TOP-5 predictions for user `ARARUVZ8RUF5T`: `[('B000WR2HB6', 5), ('B000FOI48G', 4.68), ('B000VV1Y0Y', 4.67), ('B001ET7FZE', 4.6), ('B000PKKAGO', 4.5)]`

TOP-10 predictions for user `ARARUVZ8RUF5T`: `[('B000WR2HB6', 5), ('B000FOI48G', 4.68), ('B000VV1Y0Y', 4.67), ('B001ET7FZE', 4.6), ('B000PKKAGO', 4.5), ('B00EF1QRMU', 4.47), ('B016V8YWBC', 4.46), ('B00W259T7G', 4.42), ('B00CZH3K1C', 4.33), ('B000GLRREU', 4.23)]`

TOP-20 predictions for user `ARARUVZ8RUF5T`: `[('B000WR2HB6', 5), ('B000FOI48G', 4.68), ('B000VV1Y0Y', 4.67), ('B001ET7FZE', 4.6), ('B000PKKAGO', 4.5), ('B00EF1QRMU', 4.47), ('B016V8YWBC', 4.46), ('B00W259T7G', 4.42), ('B00CZH3K1C', 4.33), ('B000GLRREU', 4.23), ('B00N2WQ2IW', 4.22), ('B00EYZY6LQ', 4.2), ('B01BNEYGQU', 4.17), ('B002GP80EU', 4.04), ('B0009RF9DW', 4.0), ('B000FI4S1E', 4.0), ('B000URXP6E', 4.0), ('B00006L9LC', 4.0), ('B0012Y0ZG2', 4.0), ('B0010HV1H4', 4.0)]`

## Exercise 4

Report Precision@ $k$  ( $P@k$ ), MAP@ $k$  and the MRR@ $k$  with  $k = \{5, 10, 20\}$  averaged across users for the CF model. Round the scores to 3 decimal places. When computing  $P@k$  and MAP@ $k$ , we consider as relevant items those with an observed rating  $\geq 4.0$  (i.e., those items from the test set with a rating  $\geq 4.0$ ). Thus, in this exercise, if a user receives an item that is present in the user's test split, the item is considered relevant since the test split only contains items with ratings  $\geq 4.0$ . Reflect on the differences obtained between the metrics and the different cut-off  $k$ .

Metrics for Neighborhood based CF:

Averaged P@5: 0.148

MAP@5: 0.167

MRR@5: 0.167

Averaged P@10: 0.078

MAP@10: 0.173

MRR@10: 0.173

Averaged P@20: 0.040

MAP@20: 0.174

MRR@20: 0.174

## Exercise 5

Based on the top-5, top-10 and top-20 predictions from Exercise 3, compute the system's hit rate averaged over the total number of users in the test set.

Hit Rate for Neighborhood based CF:

Hit Rate (top-5): 0.740

Hit Rate (top-10): 0.781

Hit Rate (top-20): 0.797