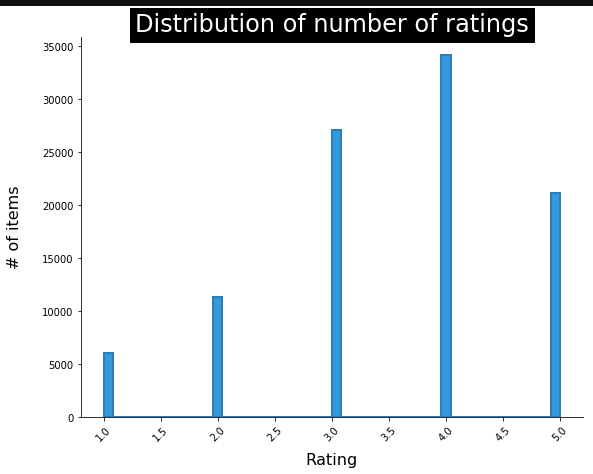
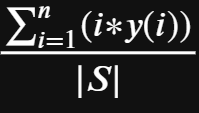
**Question 1 – Data Exploration**

* Dataset sparsity = 0.936 🡪 93.6% of the matrix is 0’s
* Distribution of ratings:



We can see that '4' is the dominant rating (34,174 ratings).  
The average rating is: , where i is a rating value (1-5), y(i) is the amount of ratings per user/item, and |S| is the size of the dataset (100K).  
We can also see that the users are more prone to give a perfect rating (5) rather than less than 3.

* The average rating is 3.52986

This dataset seems to be very large, so we expect the run with all the parameters to take a long time, especially with Stochastic Gradient Descent, which is slower than **A**lternating **L**east **S**quares.

**Question 2 – Matrix factorization model implementation and evaluation**

Best runs: (pinch to enlarge)

|  |  |  |  |
| --- | --- | --- | --- |
|  | Bias-only | SGD | ALS |
| RMSE  -  **min** training values | k=5, alpha=0.01, item\_factor\_reg=1, user\_factor\_reg=1, item\_bias=1, user\_bias=1 | k=20, alpha=0.01, item\_factor\_reg=0.01, user\_factor\_reg=0.01, item\_bias=0.01, user\_bias=0.01 | k=20, , item\_reg=0.1, user\_reg=0.01 |
| MRR  -  **max** training values | k=5, alpha=0.001, item\_factor\_reg=0.01, user\_factor\_reg=1, item\_bias=1, user\_bias=1 | k=10, alpha=0.01, item\_factor\_reg=0.1, user\_factor\_reg=0.01, item\_bias=1, user\_bias=1 | k=20, , item\_reg=0.01, user\_reg=0.01 |
| nDCG  -  **max** training values | k=20, alpha=0.001, item\_factor\_reg=0.01, user\_factor\_reg=0.01, item\_bias=0.01, user\_bias=0.01 | k=5, alpha=0.001, item\_factor\_reg=0.1, user\_factor\_reg=0.1, item\_bias=1, user\_bias=0.01 | k=5, , item\_reg=0.1, user\_reg=0.1 |

* SGD and bias-only models were not able to finish running on all the folds before the submission deadline, so this is still an estimation.
* Cutoff value used above is 5 (for MRR and nDCG), although it can be easily changed to any other number (up to maximal number of items in the dataset).

Challenges and findings:

1. SGD with learning rate of 0.1 and larger causes an overflow (discussed in the forum also).
2. SGD is relatively slow, and when including all permutations for k, alpha, number of iterations per run, 4 regularization parameters and all their permutations with one another we get 486 runs per fold, so it takes many hours to finish. Maybe we should have fixed a single value per regularization parameter, but we chose to be more robust.

(Each of these values is assigned every possible value from [0.01, 0.1, 1.0]:

'item\_fact\_reg', 'user\_fact\_reg', 'item\_bias\_reg', 'user\_bias\_reg')

1. Using regularizations prevents overfitting of the user and item vectors, but is also making the run much slower (more permutations between all parameters in the model, as discusses in section 2 above).
2. The regularization narrowed the gap between our training and test data.

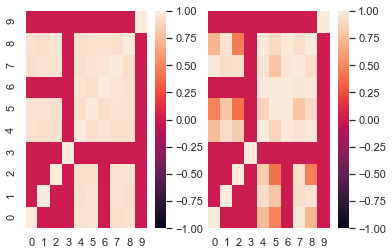
Popularity-based model:

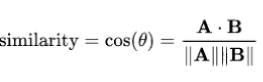
See notebook for more details.

* RMSE value: 1.068822755652113
* MRR value: 0.986744432661718
* nDCG value: 0.913142792960928

**Question 3 – Item similarity and model explainability**

1. We calculated the cosine similarity for every pair of movies (“filtered” such that they are rated by the same subset of users – i.e. user X rated both movies) from the top 10 movies (sorted by most number of ratings and then by highest average rating) in 2 flavors:
2. The straightforward calculation (left in Heatmap)
3. The calculation on the **ratings histogram** (how many ratings of 1/2/… per cell) for each of these “filtered” items (right side)





* One drawback is that popular movies tend to “go together” in ratings, although the genres might be different.

1. The meaning of latent dimensions of our matrix factorization model is the number of "clusters" we decided to divide our data into.

If we choose k which is divisible by 19 (the number of genres we have in the file "u.genre"), then we estimate that it the clusters will fit closely to the original classes (genres)

It is similar to do a k-mean clustering and then add the labels' data.

Overall, it was a very good hands-on learning experience, but it also felt too extensive to fit in just one assignment.