Recommender Systems (from Homework Assignment 4)

Suppose that an online bookseller has collected ratings information from 20 past users (**U1**-**U20**) on a selection of recent books. The ratings range from 1 = worst to 5 = best. Two new users (**NU1** and **NU2**) have recently visited the site and rated some of the books ("?" represents missing ratings). The two new users' ratings given in the last two rows.

	TRUE BELIEVER	THE DA VINCI CODE	THE WORLD IS FLAT	MY LIFE SO FAR	THE TAKING	THE KITE RUNNER	RUNNY BABBIT	HARRY POTTER
U1	1	5		3			3	5
U2	5	4			3	2	1	
U3	3		1	2	2			5
U4		3			4	1		3
U5	2	4	3			2	2	
U6	5			3	1		3	1
U7	1	4	5	5	2			4
U8	2	1			4	5	1	
U9			3	2	2			5
U10	3	5	1				4	4
U11			2	1		2		3
U12	4	4		2		1	1	4
U13			2		4		4	5
U14		5	3	3	2		1	1
U15		2			3	3		2
U16		3	2	1	1		4	4
U17	1	5	1	2		4		4
U18	5		4		3	3	4	5
U19		4		2		5	1	5
U20	2	5	1	1	5	3		4
NU1	3		5	4	2	3		5
NU2		5	2	2	4		1	3

Using the *K-Nearest Neighbor* algorithm predict the ratings of these new users for each of the books they have not yet rated. Use the Pearson **correlation coefficient** (see Assignment 1) as the similarity measure.

a. First compute the correlations between the new users (**NU1** and **NU2**) and all other users (you can show these as added columns in original spreadsheet). Then for each new user compute the predicted rating for each of the unrated items using ***K*=3** (i.e., 3 nearest neighbors). Use the weighted average function (see the appendix) to compute the predictions based on ratings of the nearest neighbors. Be sure to show the intermediate steps in your work (or provide a short explanation of how you computed the predictions).

   See attached spreadsheet Qualls_ECT584_Asssignment4_Q2.xls

   - **4.360 for NU1, The DaVinci Code.**
   - **2.500 for NU1, Runny Babbit.**
   - **3.040 for NU2, True Believer.**
   - **2.319 for NU2, The Kite Runner.**

b. Measure the **Mean Absolute Error** (MAE) on the predictions using NU1 and NU2 as test users. You can compute MAE by generating predictions for items already rated by the test user (e.g., for NU1 these are all items except "**The DaVinci Code**" and "**Runny Babbit**"). Then, for each of these items you can compute the absolute value of the difference between the predicted and the actual ratings. Finally, you can average these errors across all 12 compared items (for both NU1 and NU2) to obtain the MAE.

   See attached spreadsheet Qualls_ECT584_Asssignment4_Q2.xls

   - **0.721 is MAE for NU1**
   - **0.539 is MAE for NU2**

c. **Item-Based Collaborative Filtering.** Using the same data as above and the item-based collaborative filtering algorithm (instead of user-based CF used in the previous parts), compute the predicted rating of **NU1** on the book "**The DaVinci Code**". Note that in this case, you will need to find the ***K*** most similar *items* (books) to the target item based on their rating vectors (columns in the table), and then use **NU1**'s ratings on the ***K*** neighbor items. For this problem use ***K* = 2**, and use **Cosine Similarity** to identify the most similar neighbors to "**The DaVinci Code**". In order to compute Cosine similarities, you may assume that missing values in the ratings table are considered to be zeros.

   See attached spreadsheet Qualls_ECT584_Asssignment4_Q2.xls

   - **Two highest cosine similarity values are .642 and .673**
   - **Rating for Da Vinci Code using item-based collaborative filtering is 4.512**

Appendix

User-Based Collaborative Filtering with K-NN

How to compute Predictions:

Suppose that we have a new target user ***NU*** and we want to compute the predicted rating for ***NU*** on a target item I_t (an item ***NU*** has not rated).

Assume that we have identified the ***K*** nearest neighbors, U_1, U_2, ..., U_k for ***NU***. Let us denote the rating given by user U_i to an item I_j by $r(U_i,I_j)$. Also, let us denote the similarity of user U_i to user ***NU*** as by $sim(NU, U_i)$. Note that, generally, this similarity is computed as the Pearson correlation of the two users.

Using the weighted sum approach, the predicted rating of ***NU*** on the target item I_t can be computed as follows:

$$r(NU, I_t) = \frac{\sum_{i=1}^{K} r(U_i, I_t) \times sim(NU, U_i)}{\sum_{l=1}^{K} sim(NU, U_l)}$$

In other words, the ratings of the ***K*** neighbors are weighted by their similarity to the target user, and the sum of all these weighted ratings is divided by the sum of all the similarities across the **K** neighbors.

Important Notes:

1. Generally, when the ***K*** neighbors are identified, those whose correlations with the target user less than or equal to 0 are filtered out. So, in practice, the predictions may be computed with less than ***K*** neighbors (only those with similarities greater than 0 are considered).
2. When computing the predictions (i.e., computing the weighted average), only those neighbors are considered that have actually rated the target item, I_t, are considered. For example, suppose **K = 3**, and U_1, U_2, and U_3, are the nearest neighbors to target user ***NU***. Suppose that only U_1 and U_3 have rated item I_t. Then, the predicted rating for ***NU*** is computed using only U_1 and U_3:

$$r(NU, I_t) = \frac{[r(U_1, I_t) \times sim(NU, U_1) + r(U_3, I_t) \times sim(NU, U_3)]}{[sim(NU, U_1) + sim(NU, U_3)]}$$