Lab 4

In this lab, we keep on our thoughts on the Amazon food dataset we have been using so far, and we start using the rating the users give to the products.

A rating is a number of stars between 1 and 5, where 5 is “I love it” and 1 is “I hate it”. There are a number of philosophical investigations and research papers on what the other rating really means. For some really critical users, 3 stars could be really a good rating, maybe the maximum they would ever give to a product; others instead, have never gone below 4, and only in exceptional cases, when the product bought was really unsatisfactory.

What we will try in this laboratory is to normalize the ratings according to the user’s proclivity to give a 5-star rating.

Let’s see an example:

	A1	A2	A3	A4	A5
B1		5		4	1
B2	3				
B3		5	4	5	4
B4				5	
B5		5		2	3

The columns of this matrix are the users, while the rows are the products. User A2 has given 3 reviews: 5 stars to product B1, 5 stars to B3 and 5 stars to B5. A5 instead has given 1 star to B1, 4 stars to B3 and 3 stars to B5.

To remove user bias, we can normalize this matrix subtracting, from each column, its mean value, obtaining thus:

	A1	A2	A3	A4	A5
B1		0		0	-1.67
B2	0				
B3		0	0	1	1.33
B4				1	
B5		0		-2	0.33

Now we see that, for example, A4 has given 1 more than its personal average to product B3, that is to say she likes it, while she has given 2 less than the average to B5, so she definitely does not like it. A5 instead was not so unsatisfied by B5, since she gave 0.33 more than its average rating to this product.

Now for each of the products we can compute the average of such normalized ratings, obtaining a “normalized average rating” for each product:

B1	-0.56
B2	0
B3	0.58
B4	1
B5	-0.56

Notice how the ranking of B5 was affected by the normalization transformation.

Your task for this lab is to write a Hadoop application to compute the normalized average ratings of the products of the Amazon food dataset by considering the (sparse) products/users matrix based on the ratings available in the following HDFS file:
/data/students/bigdata-01QYD/Lab4/Reviews.csv

The format (columns) of the input data set/file is the following:
Id,ProductId,UserId,ProfileName,HelpfulnessNumerator,HelpfulnessDenominator,Score,Time,Summary,Text

Each line represents the rating given by user UserId to product ProductId. Each user rates each product at most once. However, each user can rate many products and each product can be rated by many users.
You are interested in the fields ProductId, UserId, and Score.

The output folder of your application must contain for each product its normalized average rating (one pair “product,normalized average rating” per line).

For the initial test of your application you can use the small sample dataset ReviewsSample.csv, which contains a set of reviews related to the same users, products, and ratings of the small example products/users matrix reported in this document.
ReviewSample.csv is available on the web page of the course.

Pay attention that the input file has the header. **The header of the file must be filtered**.

Hints

Beware that the products/users matrix is sparse, i.e. many of its values are null/unknown. Hence, **you can assume that the number of ratings per user is “small”, i.e., given an arbitrary user his/her list of ratings can be stored in a local Java variable**. Use this assumption in your application to avoid sending unnecessary (key, value) pairs on the network.

⚠⚠⚠ Shut down JupyterHub container ⚠⚠⚠

As soon as you complete all the tasks and activities on JupyterHub environment, please remember to shut down the container to let all your colleagues in all the sessions connect on JupyterHub and do all the lab activities.

1. Go into File -> Hub Control Panel menu
2. A new browser tab opens with the “Stop My Server” button. Click on it and wait till it disappears.