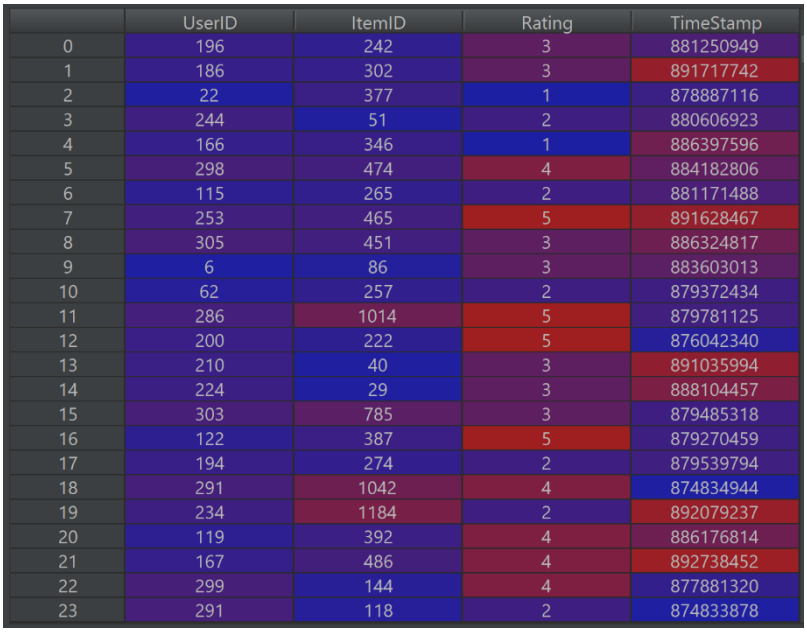
Report For EMC-dell Home Task

Omer Amar

1. **Technologies:**
   After short research about data visualization platforms and technologies I decided to use python with the next libraries:
   a. Bokeh – an interactive visualization library.
   b. Networkx – creation and manipulation of graphs.
   c. Pandas – data analysis library.
   d. Seaborn – for coloring.

2. **Handling the data:**
   I organized the dataset in pandas data frame for easy access and fast properties extraction. The data frame object:



	UserID	ItemID	Rating	TimeStamp
0	196	242	3	881250949
1	186	302	3	891717742
2	22	377	1	878887116
3	244	51	2	880606923
4	166	346	1	886397596
5	298	474	4	884182806
6	115	265	2	881171488
7	253	465	5	891628467
8	305	451	3	886324817
9	6	86	3	883603013
10	62	257	2	879372434
11	286	1014	5	879781125
12	200	222	5	876042340
13	210	40	3	891035994
14	224	29	3	888104457
15	303	785	3	879485318
16	122	387	5	879270459
17	194	274	2	879539794
18	291	1042	4	874834944
19	234	1184	2	892079237
20	119	392	4	886176814
21	167	486	4	892738452
22	299	144	4	877881320
23	291	118	2	874833878

   After that, the data was sorted by two parameters with “UserID” as the main column and “ItemID” as the secondary column.
   My goal was to arrange the data in a manner that I can easily filter the relationships between users as was instructed. I chose the following table template:

Node 1	Node 2	Items	Count	Average

   “Node 1” and “Node 2” are integers, “Items” is a list of integers, “Count” is an integer that stores the length of the “Items” list and “Average” is the average difference of ratings of the common items between “Node 1” and “Node 2”.

For that purpose, I used two methods:

a. Loop for every pair of users, find their common items, calculate the average difference of these items and append the row of data to the "Averages" data frame. The complexity of that function is $O(n^2)$, therefore I saved that data frame as a csv file "averages.csv", and I read this file from the main code. Obviously, we don't want to execute this function before every time we want to see the network graph, therefore in real time implementation I would add a function that finds changes in the data set (users may clear their ratings or rate another items), and overwrite the csv file. In this way I needed to execute the $O(n^2)$ only once, every change in rating will be updated fast in the averages data frame.
b. For every user, find his common items with every other user and loop through the items. This was the method that I didn't choose to use because it was more complex to arrange by items.

After executing the first function I was able to efficiently filter the required average difference threshold. I set the default value to 0 because otherwise the number of edges is too large. I added a number of common items threshold for extra filtering.

3. Data Visualization:

Bokeh and Networkx libraries work well together by the "from_networkx" in bokeh.models.graphs. In the code, I create a graph using this libraries combination. Firstly, I add all of the users as nodes. After that I filter the undesired rows from the averages data frame with the default thresholds and add edges. Then, I define a size and a color to every node according to its degree. Last, I add hover tools to the graph and then the graph is ready to be shown.

The nodes sizes and colors indicate the degree of the node in the graph, bigger nodes have higher degree. When you hover over a link you can see the list of item IDs of the intersection. When switching inspection policy by clicking a button, you can hover over a node and see the user's ID.

Clicking on one of the nodes will highlight it and the edges the are linked to it and fade every other detail on the figure. Clicking on every dot on the figure that isn't a node will undo the node selection.

Further explanation of the UI appears in the next section.

4. The Dashboard:

The dashboard includes the main figure with the interactive network graph and a number of control buttons. There is a tool bar at the right side of the figure for control.

The buttons:

a. "Show/Hide isolated nodes" – toggle between showing and hiding the nodes that have no edges at all.

b. "Change inspection policy" – in Bokeh library's "from_networkx" you can add a hover tool for the interactive network. Unfortunately, the hovering method is either over nodes and linked edges or over edges and linked nodes. That's why I put the toggle policy button.

c. "Increase/Decrease average threshold by 0.1" – Wait about 5-8 seconds and see the change in the figure with the new network. **Increasing** the average threshold may add more edges to the network because we filter averages that are less than or equal to the average threshold.

d. "Increase/Decrease number of items threshold by 1" - Wait about 5-8 seconds and see the change in the figure with the new network. **Decreasing** the average threshold may add more edges to the network because we filter number of common items that are greater than or equal to the number of items threshold.

5. Bugs:

a. After changing one of the thresholds and then toggling the inspection policy, the bokeh figure toolbar is disappearing from the page. Changing one of the thresholds again gets the toolbar back.