

Q1.

	Training Graph-1	Test Graph-1	Item-Item Graph-1
Number of User Nodes	943	459	
Number of Item Nodes	1650	1378	1650
Number of Edges	80000	19968	852848

	Training Graph-2	Test Graph-2	Item-Item Graph-2
Number of User Nodes	943	653	
Number of Item Nodes	1648	1386	1648
Number of Edges	80000	19964	850600

	Training Graph-3	Test Graph-3	Item-Item Graph-3
Number of User Nodes	943	869	
Number of Item Nodes	1650	1391	1650
Number of Edges	80000	19964	854612

	Training Graph-4	Test Graph-4	Item-Item Graph-4
Number of User Nodes	943	923	
Number of Item Nodes	1660	1372	1660
Number of Edges	80000	19973	883331

	Training Graph-5	Test Graph-5	Item-Item Graph-5
Number of User Nodes	943	927	
Number of Item Nodes	1650	1375	1650
Number of Edges	80000	19964	880470

Q2.

Average Normalized Gain = 0.733828399378

Q3.

Method-1 :

For every user in the group preference vector was found out. Then the task was to find out a single aggregated preference vector for the whole group so that It could be plugged in the pagerank function. This aggregated preference vector was found out by, first multiplying individual preference vectors by the importance of that user in the group and then adding all these preference vectors and dividing them by sum total of importance values of all the group members, for ex, for the group {1701:1,1702:4}, the preference vector was multiplied by 1 for user 1701 and by 4 for user 1702. Then these preference vectors were summed and divided by total of importance values that is $4+1 = 5$. Then this preference vector was plugged in the pagerank and following results were obtained :

210 181 22 174 50 496 82 87 568 121 118 1 98 172 204 15

and so on...

Method-2 :

So the idea was to obtain the pagerank score (after finding the preference vector for this user) for every group member and then multiplying the pagerank score for every user by its importance value in the group. Then from all the individual pagerank score lists , I would find top-k elements using Fagins algorithm. For ex, for group {1701:1,1702:4} , after getting pagerank vectors for both these individuals, those were multiplied by 1 and 4 respectively and then the combined list was found out thru fagins by selecting top-k elements. Following were the results obtained :

[210,22,181,496,174,82,50,148,15,588,204,274,323,151,243,633,143,934,8,202,498,423,378,211...]

The drawback of this method is computational time since this method runs pagerank for every user in the group and also takes time to implement fagins. So I showed only method-1 in the python file.