1)First of all we will take 3 columns from sub\_allMerged.csv file which are:-

-business\_id(id's of restraunts)

-user\_id(id's of users)

-stars.x(ratings given by the users to restraunts)

2)Now we will make a matrix in which rownames will show the unique id's of users and columnnames will show the unique id's of restaurants and values in every row will show the ratings given a particular users to all restaurants id's .some of them may be blank because a user did'nt give the ratings to all restaurants .

-This table can be achieved by using above 3 columns

-let's say this table is X

|  |  |  |
| --- | --- | --- |
|  | Res id1 | Res id2 |
| User id1 |  |  |
| User id2 |  |  |
| User id3 |  |  |

* The below algo will work within a cluster

3)Now we will find the similarity matrix which will be a square matrix of (nxn) order where(n is the number of unique users)

-This similarity matrix will show how a user is similar to other.

4)This we will calculate by pearson correlation function in which we will pass two vectors of ratings by two users to all restaurants.so first we will put the avg rating of users when no rating is given by the user to any restaurants

-now for making this matrix:- we will use two for loops:-

for(i in length(total no of unique users)) # for each row of similarity matrix

{

for(j in length(total no of unique users)) # for each column of similarity matrix

{

pearson\_correlation(vec1,vec2)# vec1 represents the ith row in table X & vec2 represents the jth row in table X

# this pearson correlation gives the value of ith row and jth column of nxn similarity matrix

}

}

In this way we will get the whole similarity matrix .Let's say this matrix is M

|  |  |  |  |
| --- | --- | --- | --- |
|  | User id1 | User id2 | User id3 |
| User id1 |  |  |  |
| User id2 |  |  |  |
| User id3 |  |  |  |

5)Now we have Table X(nxm) and Similarity Matrix M(nxn)# m is total no of unique restaurants.

now First we take the Average ratings of values in each row of table X and subtract the average ratings from the row values

6) multiply Matrix M and Table X and divide it by sum of similarity of users that have corated the restaurants and then add the average ratings of user for which weight has to be calculated so we will get a prediction score matrix of (nxm) order which shows prediction score of every restaurant for every user. let's say

Predicted score Matrix(W) = average ratings of user + [Matrix(M) \* Table(X)]/summation(sim of users that co-rated the restaurant for which prediction score has to find)

|  |  |  |
| --- | --- | --- |
|  | Weight of Res id1 | Weight of Res id2 |
| User id1 |  |  |
| User id2 |  |  |
| User id3 |  |  |

* Now combine all the prediction score for each cluster in a table

7) so now we will recommend the restaurants in decreasing order of prediction score to user represented by row.

8) this all thing will be done on clusters .first we will make clusters of users from all the users that are given in our dataset using some clustering algorithms. then the pseudo code that I have described above will be apply on the clusters rather than individual users.

9>So first for a new user we will find, to which cluster that users belongs to .So by the algorithm we will find what restaurants should be recommend to that user by finding similarity between only the users that belongs to that cluster then that restaurants we will recommend to that new user.