Recommendation Challenge : HackerRank

The objective of the given problem was to predict a set of challenges that a HackerRank user (or hacker) is likely to solve. provided with a list of challenges along with information on where they are located on HackerRank

Recommendation must satisfy the following conditions:

* The recommended *challenge\_id* should be present in the dataset provided to you.
* The *hacker\_id* should be present in the dataset provided to you.
* The hacker should not have already successfully solved the recommended challenge.
* The challenge should be available in the contest c8ff662c97d345d2.

Aim: Recommend 10 challenge to a particular Hacker\_ID given it should be unsolved by the user.

Approach:

(1) Grouping the submissions.data by hacker\_id and challenge\_id for extract a number of stats from each group and adding a flag to determine if a hacker solved a given challenge or not using Boolean (True/False)

(2) As mentioned in problem contest\_id : c8ff662c97d345d2 should only be considered as contest\_id. So, making a variable “target\_contest” and setting Boolean values (True/flase) i.e if the contest\_id is same as mentioned In problem then it is TRUE else False

(3) Grouping the challenges.data by challenge\_id for extract a number of stats from each group and adding sum of solved\_submission count and total submission count and merging it with the “challenges dataset”

4) Removing duplicate entries and challenges variable that are not part of submission.csv

5) Sorting values in challenges dataset by “solved\_submission\_count and total\_submission\_count in descending order and naming it recommendations

(6) Replacing NAN values with text word “UNKOWN” in both challenge and submissions dataset

7) Unique values in hacker\_id are generated in variable “hackers”

8) Generating Recommendations using Loop function

The criteria is that the hacker should not have already successfully solved the recommended challenge.

* Removing the duplicate values by challenges\_id and solved,so that hacker should not have already successfully solved the recommended challenge.
* Domain\_filters : taking the unique haker\_id values which are not duplicate and of different domains from submissions data frame
* Sub\_Domain\_filters : taking the unique haker\_id values which are not duplicate and of different Sub\_domains from submission data frame
* Domain\_recomendations : getting the domain recommendations which are also in domain filters
* Sub\_Domain\_recomendations : getting the Sub\_domain recommendations which are also in domain filters
* Domain\_subdom)recomenations : concatenating sub\_domain\_recommendation and domain recommendation by challenge\_id and removing the duplicate values from it.
* Current\_recommendations: removing the duplicates values if the challenges are solved from dom\_subdom\_recs.

9) Recommendations : restricting the loop to 10 recommendation and then concatenating current recommendation and domain and subdomain variable and removing the duplicate values if any.

10) Final submission : recommendations are converted in to data frame then reverse the coloums so that hacker\_id comes first and then the recommendations.