**User Search Based Course Recommendation Model**

The Recommender engine offers generalized recommendations to every user based on course popularity and specific skill/technology. The basic idea behind this recommender is that courses that are more popular and more critically acclaimed will have a higher probability of being liked by the average users.

The Code is elaborated in an extensive way as follows:

**Input**

The code takes input in the form of a json file(input\_usb.json). This file must contain every feature the user wants to put in, in a detailed format:-

{

"\_id":{"$oid":"5ecea1b690cfa44a44f83139"},

"user\_id":"Caramel\_1",

"search": "python"

}

Since all the user data has been already gathered in previous iterations and at the start of the courses they completed, there is no need to get access to more data. This is done to reduce redundancy.

The simple search phrase is taken into account as the keyword.

This program was initially made in IPython compiler and for converting to a .py file a lot of dependencies have to be installed.

Since this is an autonomous Process and no user input is required. The data from this model will be showed as suggestions after the user has completed a course

**Code**

* Input is taken into the program from the json file.
* The dataset is loaded on to a dataframe. Unnecessary columns are removes and the dataset is refined. In practical use this has to be done only once.
* Features like “users\_reviewed” and “’ratings” are taken into account and their mean is taken out for better accuracy.
* Now based on the above two features a dataframe is created which includes all the qualified courses for the search output.
* A weighted\_rating(x): function is created which takes in a course as an input and gives out the releation between their no of users reviewd and the ratings of that course.
* This weighted rating function is applied on the dataframe which includes the qualified courses. The results are sorted. The finalised courses are saved into a list named course\_list.
* Finally, this list is converted into json format for output.

**Output**

The final dataframe is then converted to json format and saved into a separate file with the name as 1usb.json

Example output

[

96, 97, 98, 99, 100, 101, 102, 103, 104, 105,

106, 107, 108, 109, 110, 111, 112, 113, 114, 115,

116, 117, 118, 119, 120, 121, 122, 123, 124, 125,

126, 127, 128, 129, 130, 131, 132, 133, 134, 135,

136, 137, 138, 139, 140, 141, 142, 143, 144, 145

]