**Score Based Other Recommended Courses Model**

The Score Based Other Recommended Courses Model aims at generating a set of suggestions for the user based on the previous courses applied/completed by the user.

The Code is elaborated in an extensive way as follows:

**Input**

I’ve added an input to the code which was not previously there. The code takes input in the form of a json file(CRinput.json). This file must contain every feature the user wants to put in, in a detailed format:-

{

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

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

"firstName":"Student",

"suggested\_course":"Node.ls"

}

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.

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**

* Every course is firstly indexed inside the dataset. In practical use this has to be done only once.
* Data is taken from the json input of CourseName.
* All features are saved as tags into a list and empty data slots are filled and handled. This tags list is then added to the initial dataframe for better referencing.
* A CountVectorizer() object is created. This converts a collection of text documents to a matrix of token counts. Where in the tokens are the “tags”. The count matrix created by this operation is feeded into cosine similarity function. Cosine similarity is a measure of similarity between two non-zero vectors of an inner product space. So it will measure the similarity between the vectors found by the tags in the matrix.
* Two functions are created to call the course names and to find their indices - get\_Course\_Name\_from\_index(index) and get\_index\_from\_Course\_Name(Course\_Name)
* Finally, a list is enumerated with the help of the above two functions. This list is sorted. From this list, elements are called in a loop and saved onto a dataframe. And this dataframe is converted into json format.

**Output**

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

Example output

{"Suggested Course":

{ "0":"Learn and Understand NodeJS",

"1":"Angular & NodeJS - The MEAN Stack Guide [2020 Edition]",

"2":"Node with React: Fullstack Web Development",

"3":"Node.js: The Complete Guide to Build RESTful APIs (2018)",

"4":"Node.js',' Express',' MongoDB & More: The Complete Bootcamp 2020",

"5":"Node.js API Masterclass With Express & MongoDB"

}

}

\*Previously no json input or output was provided into the model.