**Course Assets Application 4: Employee Selection Model**

The Employee Selection Model is a classification model which aims at analysing a user throughout their course and program and finally testing to conclude if the user is employable or not.

It’ll focus on things like course\_name, course\_completed\_%, program\_name, program\_completed\_%, final\_test\_score.

The Code is elaborated in an extensive way as follows:

**Input**

The code takes input in the form of a json file. 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",

"lastName":"One",

"program\_name":"Frontend",

"program\_completed\_%":"50",

"final\_test\_score":"85",

"entrance\_test\_score":"68",

"course\_completed\_%":80,

"category":"Student",

"course\_name":"Introduction to Html"

}

The input includes name, entrance test score, program name and completion percentage, course name and completion percentage final test score and category of user.

**Code**

* Data is taken from the json input like ID, name, program name and completion percentage, course name and completion percentage final test score and category of user.
* After completion of a program the user is presented with an optional test which will test their employability.
* This employability is based upon the users past performance which includes users entrance test score, final test score program completion and course completion of every individual sub course inside the program.
* Create a dataframe which includes the programs and in its columns are the subprograms and their completion percentage included in it. So each row should be like:

Program SubProg1 Subprog1\_score SubProg2 SubProg2\_score

Frontend Core.js 87 Angular.js 92

* The CheckEmployee() function is called which will firstly sort data according to the program\_name. This data will be divided into individual subprograms. \*The Criteria for giving employability test is that the user should have above 90 score in all courses and above 85 in entrance and final test of the program.
* If a user is eligible for the test, they can give the employability test. Now they must get atleast above 80 score in this test. The difficulty of the test will be above moderate to high difficulty.
* For the test, a new dataframe will be created which will save all the QIDs of questions from the individual courses inside the program which appropriate difficulty. This dataframe is then shuffled and provided to the user as a test. Shuffling must be random so that there is no repetition of questions.
* A Final dataframe will be created which will include the initial dataframe and insert the coloumn for the employability test score. Check if all are above the conditions and print pass or fail.
* Finally the output is provided as a json file.

**Output**

The final dataframe is then converted to json format and saved into a separate file with the name of the user id.

{

"employability":”Fail”

}