ASSIGNMENT ONE: FUZZY LOGIC

To execute this assignment you need to refer closely to the slide set named “Algorithm for Implementation of a Fuzzy System”. You may also refer, if you wish, to the set of slides “Fuzzy Systems in BDS”, but this would be more for interest than necessity.

Here, you have to write a program in Python that reads in two crisp inputs x1 and x2, and generates a crisp output y. In the real world of the Breakout Detection System, the two inputs correspond to the temperature slope, and the temperature rise, at any given instant of the process. The output is the value of Breakoutability.

The program will execute the following steps:

1. Normalization of the two inputs
2. Fuzzification of the two inputs
3. Evaluation of firing intensities of different rules in the FAM
4. Defuzzification
5. Denormalization to obtain the final crisp output.

You will need to perform the first two steps precisely according to the set distributions and degrees of belonging specified in slides 12 and 13 of the ppt file “Algorithm for ….”. You shall perform steps 4 and 5 exactly according to distributions and degrees specified in slides 14 to 16. You shall be using the FAM specified in slide #16 (neglect all superscripts). In slide #14, take the Centroid values of sets (mentioned as “subsets” in the tables) numbered 1 and 9 as -0.75, and +0.95, respectively. The Centroids of sets 2 to 8 should be obvious from the figure.

All formulae are provided in slides 2, 3 and 5.

You can refer to the Algorithm given in slides 7 and 8.

You are to generate crisp output values of Breakoutability for the following sets of inputs. Obviously, your program should read in the input values and print out the output and ideally avoid any direct hard-coding of numbers into the body of the code.

**Cases** (given to you in the file input.csv)**:**

|  |  |  |
| --- | --- | --- |
| Cases | x1 (°C / sec) | x2 (°C) |
| 1 | 0 | 0 |
| 2 | 9 | 25 |
| 3 | -3 | -10 |
| 4 | 0 | 29 |
| 5 | 9 | 0 |
| 6 | 7 | 23 |
| 7 | 5 | 29 |

When you get the final outputs, check if they make sense. Take this as a complete software development exercise – starting from formulation and specifications – and completion with verification and validation.

**Submission instructions:**

**Please follow the instructions Strictly as the checking will be mostly automated. Failure to follow the instructions could lead your submission to be declared void.**

* Each of you are to submit a folder named <AS1-Full Roll No>. So if your Roll No. is 18XJ1A0518, then your folder name will be <AS1-18XJ1A0518>. Exact submission logistics will be informed a little before submission date. Note that any deviation here will lead to loss (non-consideration) of your assignment.
* Within this folder you will be submitting a python script named <Roll no.>.py

Example : 18XJ1A0518.py. Please ensure that you submit only .py files and not .ipynb files. If you submit the latter, it will be considered as null.

* When we execute your code, it should read input.csv (as provided in the assignment package, and this should be included in your assignment folder) and save your predictions in the same folder as <output>.csv
* The actual output that you have generated should also be provided in the same folder and named as <Roll no.>.csv. For example, 18XJ1A0518.csv. This should have the same rows as the input file but one more column. The header of this column should be “Breakoutability” and the contents of this column will be the final crisp, denormalized values of Breakoutability for each of the given cases.
* **To summarize**: you will submit a folder with three files – a python program, the (original) input.csv and (your output file) named <Roll no.>.csv. It does not need to contain the file named <output>.csv; that we shall generate. However, the format and contents of this output file should match exactly with the <Roll No.>.csv file you have submitted.

**Deadline : February 28th.**