**STUDENT NAME**

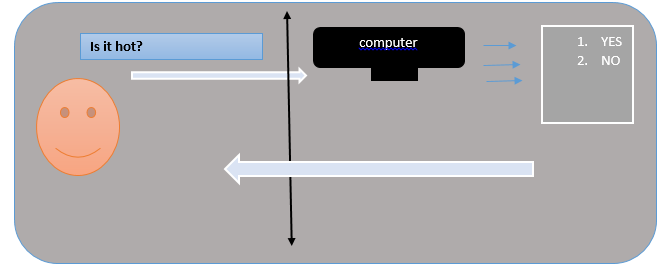
**COURSE NAME**

**STUDENT NUMBER**

**TOPIC: FUZZY LOGIC IMPLEMENTATION WITH PYTHON PROGRAMMING**

**Question One:**

Rather than spew out random 0s and 1s, computers can also be trained to make judgements to a certain degree of truth. These have led to programmable machines that almost represent human beings. They are called artificial intelligent agents or smart agents. These robots have been programmed to almost make truthful judgements based on certain inputs with human interactions. These applications have advanced and have been incorporated in fields such as medicine, agriculture, motor vehicle industry and even in the hotel industry. Traditional computers would select Boolean responses as output with no further declarations. Look at the figurative below:



From the above illustration, the computer will only give an output of yes or No , based on its binary representation of digits as 0s and 1s. However, with fuzzy logic, this is taken a step further and implemented in smart agents, so instead of just giving a single handed answer, the computer also goes ahead to give full details of temperature and recommended clothes to put on.

Moreover, we can also use fuzzies, to simulate an interaction between students and the university, this can automate for instance a student college recommendation processes and from there be able to help students make decisions on what to do next after the recommendation has been made. In such a case, the student seeking admission will enter a given input based on the prompts that appear in the screen , and based on the inputs given, the computer will be able to provide instant feedback to the user as to whether they would accept the institution or not.

Such a program can be developed in Python and executed within any Python interpreter such as the Python idle, windows CMD or an online interpreter such as <https://www.onlinegdb.com/online_python_interpreter> . For this activity , the code given below generates a fuzzy logic that will pick the selection entered, prompt an input and then give an interactive answer to the student as to whether they should pick the institution or not.

#define an new function to execute code

def fuzzy\_check():

    n = int(input("Enter Number of Institutes: ")) #promt user for the nunber of institutes

def check\_float(f1):

    try:#in case we have bugs, use the try syntax to handle

        float(f1)

        return True

    except:

        return False

    while n:# start our loop condition, validate the inputs and check for return statements

        faith = input("Enter Faith Value of Institute: ")

        if not check\_float(faith):

            print("\tInvalid Value...")

        else:

            faith = float(faith)

            if 0<=faith<0.3:

                print("\t--> Don't take admission")

            elif 0.3<=faith<0.5:

                print("\t--> Not recommended. But you may consider.")

            elif 0.5<=faith<0.8:

                print("\t--> Good institute to take admission.")

            elif 0.8<=faith<=1:

                print("\t--> Excellent institute. Highly recommended.")

            else:

                print("\tFuzzy Value must be between 0 and 1")

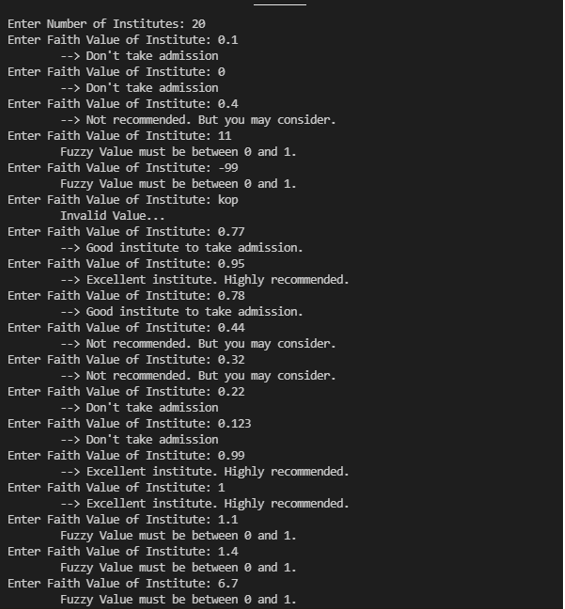
                n-=1

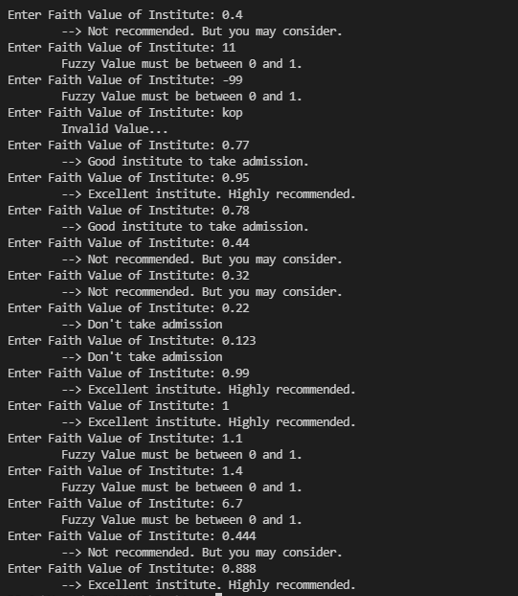
            return fuzzy\_check()

Once the code runs and executes smoothly on a python environment, the variable “faith” is taken and stored by the program then put a side in hold, meanwhile a loop statement starts and verifies classifications of inputs as 0-03, 0.3-0.5, 0.5-0.8 and finally 0.8 -1. These inputs shall hold true as long as the condition is still true, if a value out of 0 and 1, then an error is thrown and user has to enter a valid memory stack value between 0-1.

What this fuzzy logic has done in human Natural Language processing is to build a faith and confidence value in particular institutions by ranking them, where therefore proceed to recommend these institutes to potential students who would like to join them.

Based on the above, successful output of the code is as shown below:





We have run the test cases against a population of say 37 students with each giving a possibly different input and the corresponding output shown on the screens as long as the conditional statements hold true. If the student enters a value out of the scope of 0-1 then an “invalid value” error is thrown and user has to do it again.

**REFERENCES**

Grosan, C. & Abraham, A. (2011). Intelligent systems: A modern approach (Vol. 17). Berlin: Springer. doi:10.1007/978-3-642-21004-4.

* Chapter 8: “Managing Uncertainty in Rule Based Expert Systems” (pp. 187–217).

Alcala-Fdez, J., & Alonso, J. M. (2016). A survey of fuzzy systems software: Taxonomy, current research trends, and prospects. IEEE Transactions on Fuzzy Systems, 24(1), 40–56. doi:10.1109/tfuzz.2015.2426212.

Belohlavek, R., Kruse, R., & Moewes, C. (2011). Fuzzy logic for computer science. Retrieved from