

Divyadharshini K

241501053

Experiment 10

IMPLEMENTATION OF A FUZZY INFERENCE SYSTEM

Aim:

To implement Fuzzy Inference System.

Scenario:

A company wants to automate employee performance evaluation based on two factors:

1. Work Experience (Years)
2. Project Success Rate (%)

Using Fuzzy Logic, we classify employee performance as Poor, Average, or Excellent, which helps determine bonuses or promotions.

The system follows these rules:

- If experience is low AND success rate is low → Performance is Poor.
- If experience is medium OR success rate is medium → Performance is Average.
- If experience is high AND success rate is high → Performance is Excellent.

Procedure:

1. Define Input Variables:

- Experience (0 to 20 years)
- Success Rate (0 to 100%)

2. Define Output Variable:

- Performance Score (0 to 100%)

3. Create Fuzzy Membership Functions for Experience, Success Rate, and Performance:

- Low, Medium, High (for input variables)
- Poor, Average, Excellent (for output variable)

4. Define Fuzzy Rules:

- IF experience is low AND success rate is low → THEN performance is poor.
- IF experience is medium OR success rate is medium → THEN performance is average.
- IF experience is high AND success rate is high → THEN performance is excellent.

5. Build the Fuzzy Inference System (FIS) using

control rules. 6. Provide Input Values:

- Example: Experience = 12 years, Success Rate =

70% 7. Perform Fuzzy Computation to determine the final

performance score. 8. Output the Performance Score

based on fuzzy logic inference.

Program:

```
import numpy as np
import skfuzzy as fuzz
from skfuzzy import control as ctrl

# Define fuzzy variables
experience = ctrl.Antecedent(np.arange(0, 21, 1),
                             'experience')
success_rate = ctrl.Antecedent(np.arange(0, 101, 1), 'success_rate')
performance = ctrl.Consequent(np.arange(0, 101, 1), 'performance')

# Define fuzzy membership functions
experience['low'] = fuzz.trimf(experience.universe,
                               [0, 0, 10])
experience['medium'] = fuzz.trimf(experience.universe, [5, 10, 15])
```

```
experience['high'] = fuzz.trimf(experience.universe,  
[10, 20, 20])
```

```
success_rate['low'] = fuzz.trimf(success_rate.universe,  
[0, 0, 50]) success_rate['medium'] =  
fuzz.trimf(success_rate.universe, [25, 50, 75])  
success_rate['high'] = fuzz.trimf(success_rate.universe,  
[50, 100, 100])
```

```
performance['poor'] = fuzz.trimf(performance.universe, [0,  
0, 50]) performance['average'] =  
fuzz.trimf(performance.universe, [25, 50, 75])  
performance['excellent'] = fuzz.trimf(performance.universe,  
[50, 100, 100])
```

```
# Define fuzzy rules
```

```
rule1 = ctrl.Rule(experience['low'] & success_rate['low'],  
performance['poor']) rule2 = ctrl.Rule(experience['medium'] |  
success_rate['medium'], performance['average']) rule3 =  
ctrl.Rule(experience['high'] & success_rate['high'],  
performance['excellent'])
```

```
# Create FIS control system
```

```
performance_ctrl = ctrl.ControlSystem([rule1, rule2, rule3])  
performance_sim = ctrl.ControlSystemSimulation(performance_ctrl)
```

```
# Provide input values
```

```
performance_sim.input['experience'] = 12 # Example: 12  
years of experience performance_sim.input['success_rate'] =  
70 # Example: 70% success rate
```

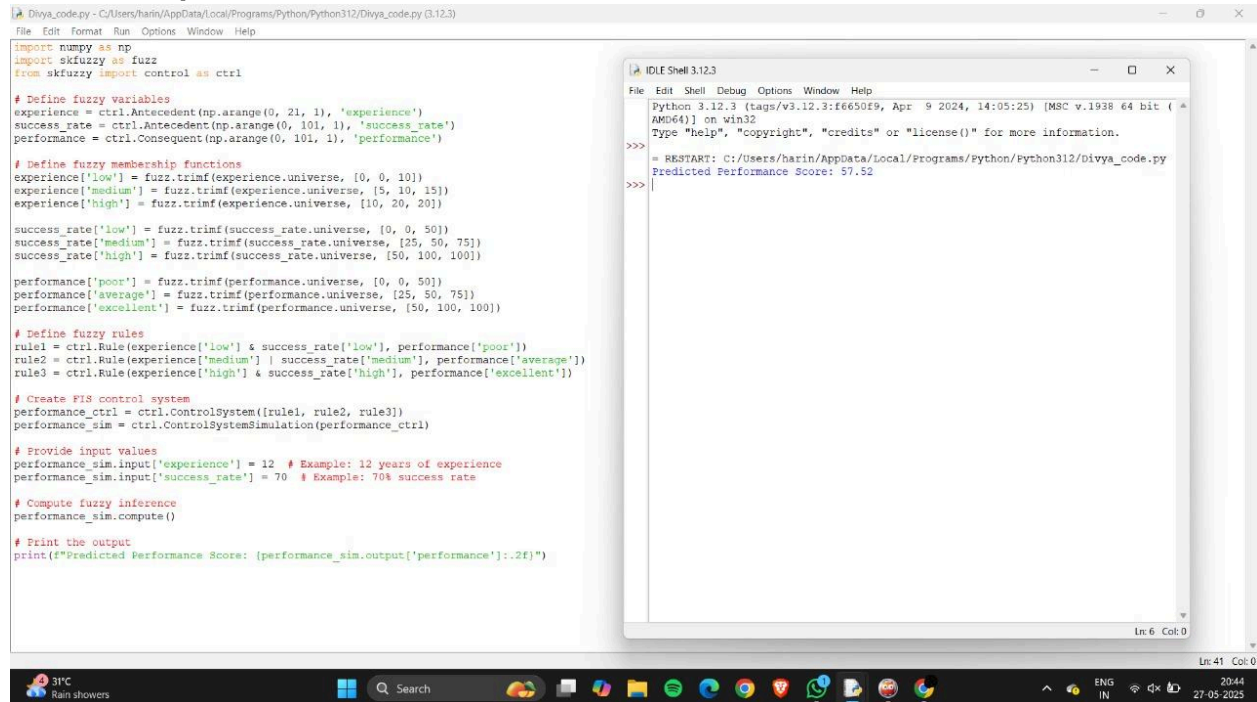
```
# Compute fuzzy inference
```

```
performance_sim.compute()
```

Print the output

```
print(f"Predicted Performance Score:  
{performance_sim.output['performance']:.2f}")
```

Output:



The screenshot displays a Python IDE with two windows. The main window shows a script for a fuzzy inference system. The script defines fuzzy variables for 'experience', 'success_rate', and 'performance', sets up membership functions, defines fuzzy rules, and performs a simulation with input values of 12 for experience and 70 for success rate. The output is a predicted performance score of 57.52. A second window, titled 'IDLE Shell 3.12.3', shows the execution output, confirming the predicted performance score of 57.52.

```
Diya_code.py - C:/Users/harin/AppData/Local/Programs/Python/Python312/Diyya_code.py (3.12.3)
File Edit Format Run Options Window Help

import numpy as np
import skfuzzy as fuzz
from skfuzzy import control as ctrl

# Define fuzzy variables
experience = ctrl.Antecedent(np.arange(0, 21, 1), 'experience')
success_rate = ctrl.Antecedent(np.arange(0, 101, 1), 'success_rate')
performance = ctrl.Consequent(np.arange(0, 101, 1), 'performance')

# Define fuzzy membership functions
experience['low'] = fuzz.trimf(experience.universe, [0, 0, 10])
experience['medium'] = fuzz.trimf(experience.universe, [5, 10, 15])
experience['high'] = fuzz.trimf(experience.universe, [10, 20, 20])

success_rate['low'] = fuzz.trimf(success_rate.universe, [0, 0, 50])
success_rate['medium'] = fuzz.trimf(success_rate.universe, [25, 50, 75])
success_rate['high'] = fuzz.trimf(success_rate.universe, [50, 100, 100])

performance['poor'] = fuzz.trimf(performance.universe, [0, 0, 50])
performance['average'] = fuzz.trimf(performance.universe, [25, 50, 75])
performance['excellent'] = fuzz.trimf(performance.universe, [50, 100, 100])

# Define fuzzy rules
rule1 = ctrl.Rule(experience['low'] & success_rate['low'], performance['poor'])
rule2 = ctrl.Rule(experience['medium'] | success_rate['medium'], performance['average'])
rule3 = ctrl.Rule(experience['high'] & success_rate['high'], performance['excellent'])

# Create FIS control system
performance_ctrl = ctrl.ControlSystem([rule1, rule2, rule3])
performance_sim = ctrl.ControlSystemSimulation(performance_ctrl)

# Provide input values
performance_sim.input['experience'] = 12 # Example: 12 years of experience
performance_sim.input['success_rate'] = 70 # Example: 70% success rate

# Compute fuzzy inference
performance_sim.compute()

# Print the output
print(f"Predicted Performance Score: {performance_sim.output['performance']:.2f}")

IDLE Shell 3.12.3
File Edit Shell Debug Options Window Help
Python 3.12.3 (tags/v3.12.3:f6650f9, Apr 9 2024, 14:05:25) [MSC v.1938 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/harin/AppData/Local/Programs/Python/Python312/Diyya_code.py
Predicted Performance Score: 57.52
>>>
```