

main.py

Run

Share

Clear

```
1 import numpy as np
2 import skfuzzy as fuzz
3 from skfuzzy import control as ctrl
4
5 experience = ctrl.Antecedent(np.arange(0, 21, 1), 'experience')
6 success_rate = ctrl.Antecedent(np.arange(0, 101, 1), 'success_rate')
7 performance = ctrl.Consequent(np.arange(0, 101, 1), 'performance')
8
9 experience['low'] = fuzz.trimf(experience.universe, [0, 0, 10])
10 experience['medium'] = fuzz.trimf(experience.universe, [5, 10, 15])
11 experience['high'] = fuzz.trimf(experience.universe, [10, 20, 20])
12
13 success_rate['low'] = fuzz.trimf(success_rate.universe, [0, 0, 50])
14 success_rate['medium'] = fuzz.trimf(success_rate.universe, [25, 50, 75])
15 success_rate['high'] = fuzz.trimf(success_rate.universe, [50, 100, 100])
16
17 performance['poor'] = fuzz.trimf(performance.universe, [0, 0, 50])
18 performance['average'] = fuzz.trimf(performance.universe, [25, 50, 75])
19 performance['excellent'] = fuzz.trimf(performance.universe, [50, 100, 100])
20
21 rule1 = ctrl.Rule(experience['low'] & success_rate['low'], performance['poor'])
22 rule2 = ctrl.Rule(experience['medium'] | success_rate['medium'], performance['average'])
23 rule3 = ctrl.Rule(experience['high'] & success_rate['high'], performance['excellent'])
24
25 performance_ctrl = ctrl.ControlSystem([rule1, rule2, rule3])
26 performance_sim = ctrl.ControlSystemSimulation(performance_ctrl)
27
28 performance_sim.input['experience'] = 13
29 performance_sim.input['success_rate'] = 68
30
31 performance_sim.compute()
32
33 print(f"Predicted performance score: {performance_sim.output['performance']:1.2f}")
34
```


Output

Predicted Performance Score: 67.85

Programiz

Premium Courses by Programiz

Learn More



Breaking news

Search

ENG IN

17:57

04-06-2025