

ALP9.ipynb

File Edit View Insert Runtime Tools Help

+ Code + Text

✓ RAM 100% Disk 100%

21162101012\_CBA\_Yash\_Lakhtariya

AI Practical 9

AIM - Create a Fuzzy rules based systems

Part 1: Create a fuzzy rule-based system for Washing Machine.

Input Variables :

degree\_of\_dirt and type\_of\_dirt - (ranging between 1 to 100)

Rules :

1. If dirtiness\_of\_clothes is Large and type\_of\_dirt is Greasy then wash\_time is VeryLong

2. If dirtiness\_of\_clothes is Medium and type\_of\_dirt is Greasy then wash\_time is Long

3. If dirtiness\_of\_clothes is Small and type\_of\_dirt is Greasy then wash\_time is Long

4. If dirtiness\_of\_clothes is Large and type\_of\_dirt is Medium then wash\_time is Long

5. If dirtiness\_of\_clothes is Medium and type\_of\_dirt is Medium then wash\_time is Medium

6. If dirtiness\_of\_clothes is Small and type\_of\_dirt is Medium then wash\_time is Medium

7. If dirtiness\_of\_clothes is Large and type\_of\_dirt is NotGreasy then wash\_time is Medium

8. If dirtiness\_of\_clothes is Medium and type\_of\_dirt is NotGreasy then wash\_time is Short

9. If dirtiness\_of\_clothes is Small and type\_of\_dirt is NotGreasy then wash\_time is VeryShort

[1] !pip install scikit-fuzzy

Collecting scikit-fuzzy  
Downloading scikit-fuzzy-0.4.2.tar.gz (993 kB)  
994.0/994.0 kB 0.1 MB/s eta 0:00:00  
Preparing metadata (setup.py) ... done  
Requirement already satisfied: numpy<=1.6.0 in /usr/local/lib/python3.10/dist-packages (from scikit-fuzzy) (1.25.2)  
Requirement already satisfied: scipy<=0.9.0 in /usr/local/lib/python3.10/dist-packages (from scikit-fuzzy) (1.11.4)  
Requirement already satisfied: networkx<1.9.0 in /usr/local/lib/python3.10/dist-packages (from scikit-fuzzy) (3.3)  
Building wheels for collected packages: scikit-fuzzy  
Building wheel for scikit-fuzzy (setup.py) ... done  
Created wheel for scikit-fuzzy: filename=scikit\_fuzzy-0.4.2-py3-none-any.whl size=894878 sha256=aa4d7cdd21eea43c199b09b8bcb8466a1e0305ee39bdc1835fe3f8a87605507  
Stored in directory: /root/.cache/pip/wheels/4f/86/1b/dfe97134e2c6313e519bceba9563fede7be79440b22094bcb8  
Successfully built scikit-fuzzy  
Installing collected packages: scikit-fuzzy  
Successfully installed scikit-fuzzy-0.4.2

[2] import numpy as np  
import skfuzzy as fuzz  
from skfuzzy import control as ctrl

[3] d\_d = ctrl.Antecedent(np.arange(1, 101, 1), 'dirt degree')  
t\_d = ctrl.Antecedent(np.arange(1, 101, 1), 'dirt type')  
w\_t = ctrl.Consequent(np.arange(1, 101, 1), 'time')

[4] d\_d.automf(3)  
t\_d.automf(3)

[5] w\_t['VShort'] = fuzz.trimf(w\_t.universe, [1, 1, 25])  
w\_t['Short'] = fuzz.trimf(w\_t.universe, [1, 25, 50])  
w\_t['Medium'] = fuzz.trimf(w\_t.universe, [25, 50, 75])  
w\_t['Long'] = fuzz.trimf(w\_t.universe, [50, 75, 100])  
w\_t['VLong'] = fuzz.trimf(w\_t.universe, [75, 100, 100])

[6] r1=(ctrl.Rule(d\_d['good'] | t\_d['poor'], w\_t['VLong']))  
r2=(ctrl.Rule(d\_d['average'] | t\_d['poor'], w\_t['Long']))  
r3=(ctrl.Rule(d\_d['poor'] | t\_d['poor'], w\_t['Long']))  
r4=(ctrl.Rule(d\_d['good'] | t\_d['average'], w\_t['Long']))  
r5=(ctrl.Rule(d\_d['average'] | t\_d['average'], w\_t['Medium']))  
r6=(ctrl.Rule(d\_d['poor'] | t\_d['average'], w\_t['Medium']))  
r7=(ctrl.Rule(d\_d['good'] | t\_d['good'], w\_t['Medium']))  
r8=(ctrl.Rule(d\_d['average'] | t\_d['good'], w\_t['Short']))  
r9=(ctrl.Rule(d\_d['poor'] | t\_d['good'], w\_t['VShort']))

[7] wash\_ctrl = ctrl.ControlSystem([r1, r2, r3, r4, r5, r6, r7, r8, r9])  
washing = ctrl.ControlSystemSimulation(wash\_ctrl)

[8] d\_deg = int(input('Degree of dirt [1-100]: '))  
t\_deg = int(input('Type of dirt [1-100]: '))  
  
Degree of dirt [1-100]: 64  
Type of dirt [1-100]: 16

[9] washing.input['dirt degree'] = d\_deg  
washing.input['dirt type'] = t\_deg

[10] washing.compute()  
print("\nWash time : ", washing.output['time'].round(3), 'seconds')

Wash time : 54.09 seconds

[11] d\_d.view()

[12] t\_d.view()

[13] w\_t.view()

Part 2: Create a Fuzzy rule based system for giving tip to a restaurant worker.

Input Variables :

service and quality(ranging between 1 to 60)

Rules:

1. If services are good and quality is high then tip should be maximum.

2. If services are medium and quality is medium then tip should be average

3. If services are low and quality is low then tip should be minimum.

[14] service = ctrl.Antecedent(np.arange(1, 61, 1), 'service')  
quality = ctrl.Antecedent(np.arange(1, 61, 1), 'quality')  
tip = ctrl.Consequent(np.arange(0, 26, 1), 'tip')

[15] service.automf(3, names=['low', 'medium', 'high'])  
quality.automf(3, names=['low', 'medium', 'high'])

[16] tip['min'] = fuzz.trimf(tip.universe, [0, 0, 13])  
tip['avg'] = fuzz.trimf(tip.universe, [0, 13, 25])  
tip['max'] = fuzz.trimf(tip.universe, [13, 25, 25])

[17] r1 = ctrl.Rule(service['high'] | quality['high'], tip['max'])  
r2 = ctrl.Rule(service['medium'] | quality['medium'], tip['avg'])  
r3 = ctrl.Rule(service['low'] | quality['low'], tip['min'])

[18] tip\_ctrl = ctrl.ControlSystem([r1, r2, r3])  
tipping = ctrl.ControlSystemSimulation(tip\_ctrl)

[22] q\_in = int(input('Quality value [1-60]: '))  
s\_in = int(input('Service value [1-60]: '))  
  
Quality value [1-60]: 32  
Service value [1-60]: 16

[23] tipping.input['quality'] = q\_in  
tipping.input['service'] = s\_in  
  
tipping.compute()

[29] print('Tip :', tipping.output['tip'].round(4), 'Rupees')

Tip : 11.5098 Rupees

[25] service.view()

[26] quality.view()

[27] tip.view()

Colab paid products - Cancel contracts here

0s completed at 7:11 PM