**Practical 9B**

**# Solve Tipping Problem using fuzzy logic**

import numpy as np

import skfuzzy as fuzz

from skfuzzy import control as ctrl

quality = ctrl.Antecedent(np.arange(0, 11, 1), 'quality')

service = ctrl.Antecedent(np.arange(0, 11, 1), 'service')

tip = ctrl.Consequent(np.arange(0, 26, 1), 'tip')

**# Auto-membership function population is possible with .automf(3, 5, or 7)**

quality.automf(3)

service.automf(3)

tip['low'] = fuzz.trimf(tip.universe, [0, 0, 13])

tip['medium'] = fuzz.trimf(tip.universe, [0, 13, 25])

tip['high'] = fuzz.trimf(tip.universe, [13, 25, 25])

quality['average'].view()

service.view()

tip.view()

rule1 = ctrl.Rule(quality['poor'] | service['poor'], tip['low'])

rule2 = ctrl.Rule(service['average'], tip['medium'])

rule3 = ctrl.Rule(service['good'] | quality['good'], tip['high'])

rule1.view()

tipping\_ctrl = ctrl.ControlSystem([rule1, rule2, rule3])

tipping = ctrl.ControlSystemSimulation(tipping\_ctrl)

**# Pass inputs to the ControlSystem using Antecedent labels with Pythonic API**

**# Note: if you like passing many inputs all at once, use .inputs(dict\_of\_data)**

tipping.input['quality'] = 6.5

tipping.input['service'] = 9.8

**# Crunch the numbers**

tipping.compute()

print (tipping.output['tip'])

tip.view(sim=tipping)

#The resulting suggested tip is \*\*20.24%\*\*.

**Output :**

19.847607361963192  
image:: PLOT2RST.current\_figure





