**Fuzzy inference system**

**PROGRAM:**

import numpy as np

import skfuzzy as fuzz

from skfuzzy import control as ctrl

# Create fuzzy variables

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

speed = ctrl.Consequent(np.arange(0, 101, 1), 'speed')

# Define membership functions for distance

distance['near'] = fuzz.trimf(distance.universe, [0, 0, 5])

distance['medium'] = fuzz.trimf(distance.universe, [0, 5, 10])

distance['far'] = fuzz.trimf(distance.universe, [5, 10, 10])

# Define membership functions for speed

speed['slow'] = fuzz.trimf(speed.universe, [0, 0, 50])

speed['medium'] = fuzz.trimf(speed.universe, [0, 50, 100])

speed['fast'] = fuzz.trimf(speed.universe, [50, 100, 100])

# Define rules

rule1 = ctrl.Rule(distance['near'], speed['slow'])

rule2 = ctrl.Rule(distance['medium'], speed['medium'])

rule3 = ctrl.Rule(distance['far'], speed['fast'])

# Create the control system

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

car\_speed = ctrl.ControlSystemSimulation(speed\_ctrl)

# Input distance and compute speed

car\_speed.input['distance'] = 7

car\_speed.compute()

# Print the computed speed

print("Computed speed:", car\_speed.output['speed'])

OUTPUT:

