1.Código usado:

import numpy as np

import skfuzzy as fuzz

from skfuzzy import control as ctrl

import matplotlib.pyplot as plt

calorias = ctrl.Antecedent(np.arange(0, 3000, 1), 'calorias')

peso = ctrl.Consequent(np.arange(0, 200, 1), 'peso')

#Triangular

calorias['baixa'] = fuzz.trimf(calorias.universe, [0, 0, 1500])

calorias['media'] = fuzz.trimf(calorias.universe, [0, 1500, 3000])

calorias['alta'] = fuzz.trimf(calorias.universe, [1500, 3000, 3000])

calorias.view()

plt.show()

#Gauss

calorias['baixa'] = fuzz.gaussmf(calorias.universe, 500, 400)

calorias['media'] = fuzz.gaussmf(calorias.universe, 1500, 600)

calorias['alta'] = fuzz.gaussmf(calorias.universe, 2500, 400)

calorias.view()

plt.show()

#Trapezoidal

calorias['baixa'] = fuzz.trapmf(calorias.universe, [0, 0, 1000, 1500])

calorias['media'] = fuzz.trapmf(calorias.universe, [1000, 1500, 2000, 2500])

calorias['alta'] = fuzz.trapmf(calorias.universe, [2000, 2500, 3000, 3000])

calorias.view()

plt.show()

peso['baixo'] = fuzz.trimf(peso.universe, [0, 0, 100])

peso['medio'] = fuzz.trimf(peso.universe, [0, 100, 200])

peso['alto'] = fuzz.trimf(peso.universe, [100, 200, 200])

calorias.automf(names=['baixa', 'media', 'alta'])

peso.automf(names=['magro', 'media', 'obesidade'])

regra1= ctrl.Rule(calorias['baixa'], peso['magro'])

regra2= ctrl.Rule(calorias['media'], peso['media'])

regra3= ctrl.Rule(calorias['alta'], peso['obesidade'])

controlador = ctrl.ControlSystem([regra1, regra2, regra3])

simulaPeso = ctrl.ControlSystemSimulation(controlador)

simulaPeso.input['calorias'] = 500

simulaPeso.compute()

print(simulaPeso.output['peso'])

calorias.view(sim=simulaPeso)

peso.view(sim=simulaPeso)

plt.show()

2.Comparação das saídas

Trinagular:

Gráfico, Gráfico de linhas

Descrição gerada automaticamente

Gaussiana:

Diagrama

Descrição gerada automaticamente

Trapeizodal:

Gráfico, Gráfico de linhas

Descrição gerada automaticamente

3.Saidas

Calorias = 500

Uma imagem contendo Gráfico de linhas

Descrição gerada automaticamente

Calorias = 1500

Uma imagem contendo Gráfico

Descrição gerada automaticamente  
Calorias 2500

Gráfico

Descrição gerada automaticamente

4.Implementação de atividade física:  
  
import numpy as np

import skfuzzy as fuzz

from skfuzzy import control as ctrl

import matplotlib.pyplot as plt

calorias = ctrl.Antecedent(np.arange(0, 3000, 1), 'calorias')

atividadeFisica = ctrl.Antecedent(np.arange(0, 10, 1), 'atividadeFisica')

peso = ctrl.Consequent(np.arange(0, 200, 1), 'peso')

calorias['baixa'] = fuzz.trimf(calorias.universe, [0, 0, 1500])

calorias['media'] = fuzz.trimf(calorias.universe, [0, 1500, 3000])

calorias['alta'] = fuzz.trimf(calorias.universe, [1500, 3000, 3000])

atividadeFisica['baixa'] = fuzz.trimf(atividadeFisica.universe, [0, 0, 5])

atividadeFisica['media'] = fuzz.trimf(atividadeFisica.universe, [0, 5, 10])

atividadeFisica['alta'] = fuzz.trimf(atividadeFisica.universe, [5, 10, 10])

peso['baixo'] = fuzz.trimf(peso.universe, [0, 0, 100])

peso['medio'] = fuzz.trimf(peso.universe, [0, 100, 200])

peso['alto'] = fuzz.trimf(peso.universe, [100, 200, 200])

calorias.automf(names=['baixa', 'media', 'alta'])

atividadeFisica.automf(names=['baixa', 'media', 'alta'])

peso.automf(names=['magro', 'media', 'obesidade'])

regra1= ctrl.Rule(calorias['baixa'] & atividadeFisica['baixa'], peso['magro'])

regra2= ctrl.Rule(calorias['media'] & atividadeFisica['media'], peso['media'])

regra3= ctrl.Rule(calorias['media'] & atividadeFisica['alta'], peso['magro'])

regra4= ctrl.Rule(calorias['alta'] & atividadeFisica['baixa'], peso['obesidade'])

regra5= ctrl.Rule(calorias['alta'] & atividadeFisica['media'], peso['obesidade'])

regra6= ctrl.Rule(calorias['alta'] & atividadeFisica['alta'], peso['media'])

controlador = ctrl.ControlSystem([regra1, regra2, regra3,regra4, regra5, regra6])

simulaPeso = ctrl.ControlSystemSimulation(controlador)

simulaPeso.input['calorias'] = 500

simulaPeso.input['atividadeFisica'] = 5

simulaPeso.compute()

print(simulaPeso.output['peso'])

calorias.view(sim=simulaPeso)

peso.view(sim=simulaPeso)

atividadeFisica.view(sim=simulaPeso)

plt.show()

5.Achamos muito intuitivo resolver esse problema usando o fuzzy, pois além de ser fácil de entender o que cada linha de código está fazendo, também te disponibiliza muitas possibilidades para expandir o quão especifico que seja a amplitude dos dados.

Por exemplo, foi implementado apenas três medidas, magro, médio e obeso, mas poderíamos ter aumentado essas variáveis em n números, deixando a analise de dados bem especifica.

Outro exemplo que pensamos seria por exemplo para probabilidade de acidentes em transito.

Poderiamos ter como variáveis de antecedente como “Velocidade” e “Quantidade de curvas” e com isso chegar na variável consequente “Probabilidade de acidente”.