Міністерство освіти і науки України

Дніпровський національний університет імені Олеся Гончара

Факультет прикладної математики

Кафедра комп'ютерних технологій

Лабораторна робота №1

|  |  |
| --- | --- |
| Виконавець: | студент групи ПК-21м-1  Панасенко Єгор  Сергійович |

# Постановка задачі

Тема: «Створення нечіткої продукційної моделі»

Мета роботи:

1. Розробити нечітку продукційну модель для обраної самостійно предметної області.
2. Створити і налагодити алгоритм, який реалізує розроблену модель.

Порядок виконання:

1. Обрати предметну область і створити базу правил.
2. Ввести лінгвістичні змінні для створеної бази правил (шкали, функції приналежності, терми).
3. Побудувати нечітку модель виведення на основі бази правил.
4. Провести дефазифікацію вихідної змінної.
5. Створити алгоритмічну реалізацію моделі нечіткого логічного виводу.

# Хід роботи

Розглянемо модель роботи холодильника у залежності від температури навколишнього середовища, та терміну придатності продукта.

Нехай x1 — температура навколишнього середовища у градусах Цельсія, x2 - термін придатності продукта у днях, x3 — потужність роботи холодильника в діапазоні [0, 10].

Значення нечітких змінних:

* – низька (Н) або короткий (К)
* – середня (С)
* – висока (В) або довгий (Д)

Правила:

* Якщо =Н та =К, то =С
* Якщо =Н та =С, то =Н
* Якщо =Н та =Д, то =Н
* Якщо =С та =К, то =В
* Якщо =С та =С, то =С
* Якщо =С та =Д, то =Н
* Якщо =В та =К, то =В
* Якщо =В та =С, то =В
* Якщо =В та =Д, то =С

Нечітка модель:

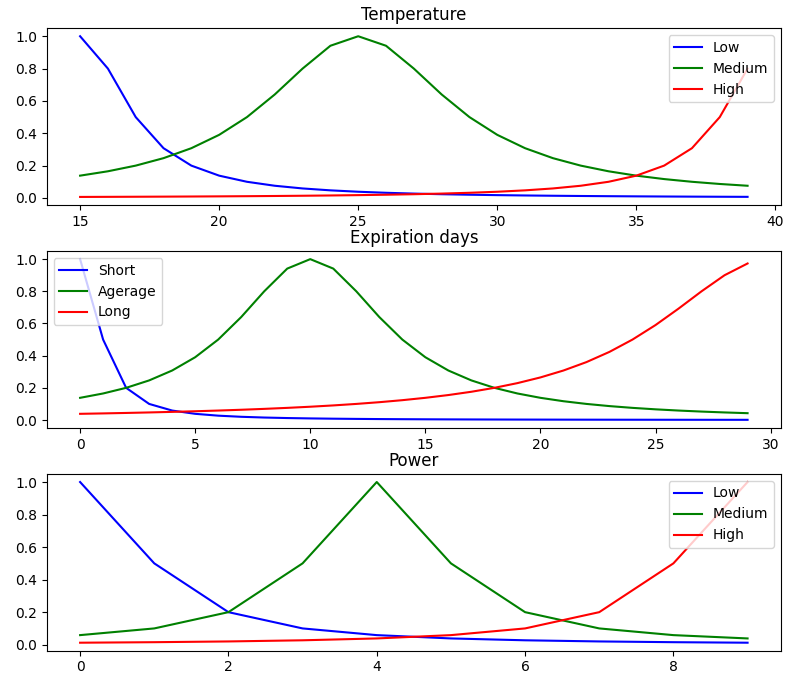
Шкала для змінної виводу

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| низька | | середня | | висока | |
|  |  |  |  |  |  |

Дефазифікація

# Алгоритмічна реалізація моделі

Наведемо функції належності термів:

Наведемо декілька прикладів роботи алгоритму:

15 and 0 -> 4.108

25 and 10 -> 3.779

25 and 10 -> 3.779

25 and 20 -> 1.688

40 and 30 -> 3.934

15 and 30 -> 0.202

Код алгоритму:

import matplotlib.pyplot as plt

import numpy as np

# Inspiration: https://pythonhosted.org/scikit-fuzzy/auto\_examples/plot\_tipping\_problem.html

def mf(x, b, c):

return 1/(1+((x-b)/c)\*\*2)

x3\_ = 0

x31 = 2

x3m = 4

x32 = 7

x33 = 9

x\_temp = np.arange(15, 40, 1)

x\_expr = np.arange(0, 30, 1)

x\_powr = np.arange(0, 10, 1)

temp\_lo\_f = lambda x: mf(x, 15, 2)

temp\_md\_f = lambda x: mf(x, 25, 4)

temp\_hi\_f = lambda x: mf(x, 40, 2)

expr\_sh\_f = lambda x: mf(x, 0, 1)

expr\_md\_f = lambda x: mf(x, 10, 4)

expr\_ln\_f = lambda x: mf(x, 30, 6)

powr\_lo\_f = lambda x: mf(x, 0, 1)

powr\_md\_f = lambda x: mf(x, 4, 1)

powr\_hi\_f = lambda x: mf(x, 9, 1)

temp\_lo = temp\_lo\_f(x\_temp)

temp\_md = temp\_md\_f(x\_temp)

temp\_hi = temp\_hi\_f(x\_temp)

expr\_sh = expr\_sh\_f(x\_expr)

expr\_md = expr\_md\_f(x\_expr)

expr\_ln = expr\_ln\_f(x\_expr)

powr\_lo = powr\_lo\_f(x\_powr)

powr\_md = powr\_md\_f(x\_powr)

powr\_hi = powr\_hi\_f(x\_powr)

fig, (ax0, ax1, ax2) = plt.subplots(nrows=3, figsize=(8, 9))

ax0.plot(x\_temp, temp\_lo, 'b', linewidth=1.5, label='Low')

ax0.plot(x\_temp, temp\_md, 'g', linewidth=1.5, label='Medium')

ax0.plot(x\_temp, temp\_hi, 'r', linewidth=1.5, label='High')

ax0.set\_title('Temperature')

ax0.legend()

ax1.plot(x\_expr, expr\_sh, 'b', linewidth=1.5, label='Short')

ax1.plot(x\_expr, expr\_md, 'g', linewidth=1.5, label='Agerage')

ax1.plot(x\_expr, expr\_ln, 'r', linewidth=1.5, label='Long')

ax1.set\_title('Expiration days')

ax1.legend()

ax2.plot(x\_powr, powr\_lo, 'b', linewidth=1.5, label='Low')

ax2.plot(x\_powr, powr\_md, 'g', linewidth=1.5, label='Medium')

ax2.plot(x\_powr, powr\_hi, 'r', linewidth=1.5, label='High')

ax2.set\_title('Power')

ax2.legend()

plt.tight\_layout()

def mu\_lo(x):

return max([

min(temp\_lo\_f(x[0]), expr\_md\_f(x[1])),

min(temp\_lo\_f(x[0]), expr\_ln\_f(x[1])),

min(temp\_md\_f(x[0]), expr\_ln\_f(x[1])),

])

def mu\_md(x):

return max([

min(temp\_lo\_f(x[0]), expr\_sh\_f(x[1])),

min(temp\_md\_f(x[0]), expr\_md\_f(x[1])),

min(temp\_hi\_f(x[0]), expr\_ln\_f(x[1])),

])

def mu\_hi(x):

return max([

min(temp\_md\_f(x[0]), expr\_sh\_f(x[1])),

min(temp\_hi\_f(x[0]), expr\_sh\_f(x[1])),

min(temp\_hi\_f(x[0]), expr\_md\_f(x[1])),

])

def defuzz(x):

print(mu\_lo(x), mu\_md(x), mu\_hi(x))

return (x3\_\*mu\_lo(x)+x3m\*mu\_md(x)+x33\*mu\_hi(x))/(mu\_lo(x)+mu\_md(x)+mu\_hi(x))

def solve(x):

print("%2.f and %2.f -> %0.3f" % (x[0], x[1], defuzz(x)))

solve((15, 0))

solve((25, 10))

solve((25, 10))

solve((25, 20))

solve((40, 30))

solve((15, 30))

plt.show()