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:

- ϕ_1 – низька (Н) або короткий (К)

- ϕ_2 – середня (С)
- ϕ_3 – висока (В) або довгий (Д)

Правила:

- Якщо x_1 =Н та x_2=К, то x_3=С
- Якщо x_1 =Н та x_2=С, то x_3=Н
- Якщо x_1 =Н та x_2=Д, то x_3=Н
- Якщо x_1 =С та x_2=К, то x_3=В
- Якщо x_1 =С та x_2=С, то x_3=С
- Якщо x_1 =С та x_2=Д, то x_3=Н
- Якщо x_1 =В та x_2=К, то x_3=В
- Якщо x_1 =В та x_2=С, то x_3=В
- Якщо x_1 =В та x_2=Д, то x_3=С

Нечітка модель:

- $\mu_{\text{Н}}(x_3)=max\begin{pmatrix} min(\mu_{\text{Н}}(x_1),\mu_{\text{С}}(x_2)) \\ min(\mu_{\text{Н}}(x_1),\mu_{\text{Д}}(x_2)) \\ min(\mu_{\text{С}}(x_1),\mu_{\text{Д}}(x_2)) \end{pmatrix}$
- $\mu_{\text{С}}(x_3)=max\begin{pmatrix} min(\mu_{\text{Н}}(x_1),\mu_{\text{К}}(x_2)) \\ min(\mu_{\text{С}}(x_1),\mu_{\text{С}}(x_2)) \\ min(\mu_{\text{В}}(x_1),\mu_{\text{Д}}(x_2)) \end{pmatrix}$
- $\mu_{\text{В}}(x_3)=max\begin{pmatrix} min(\mu_{\text{С}}(x_1),\mu_{\text{К}}(x_2)) \\ min(\mu_{\text{В}}(x_1),\mu_{\text{К}}(x_2)) \\ min(\mu_{\text{В}}(x_1),\mu_{\text{С}}(x_2)) \end{pmatrix}$

Шкала для змінної виводу

низька		середня	висока
$\underline{x_3}$	x_3^1	x_3^2	$\overline{x_3}$

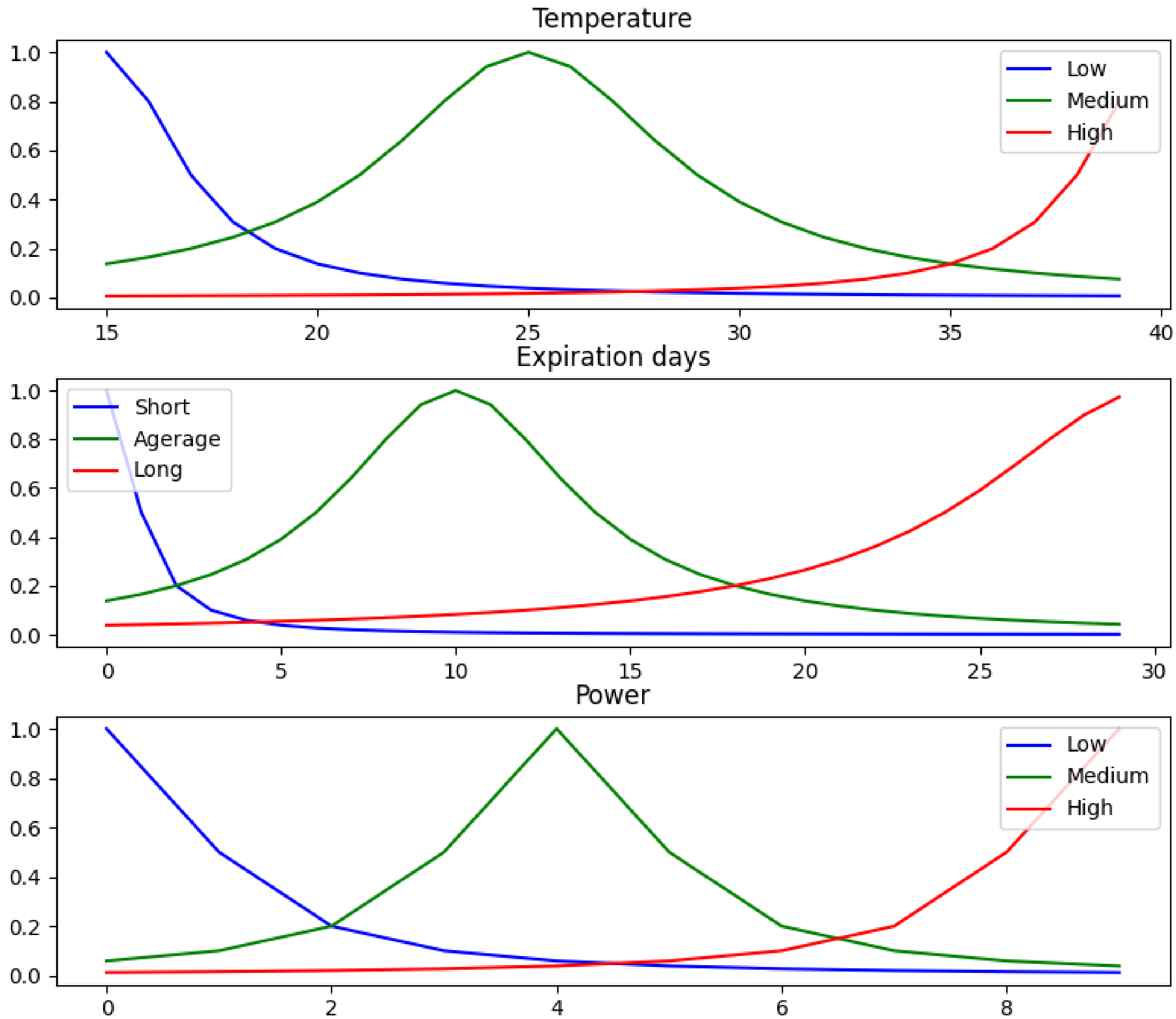
$x_3=\{\underline{x_3},x_3^1,x_3^2,\overline{x_3}\}$

Дефазифікація

$$x_3=\frac{\underline{x_3}\,\mu_{\text{Н}}(x_3)+x_3^2\mu_{\text{С}}(x_3)+\overline{x_3}\,\mu_{\text{В}}(x_3)}{\mu_{\text{Н}}(x_3)+\mu_{\text{С}}(x_3)+\mu_{\text{В}}(x_3)}$$

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()
```