Міністерство освіти і науки України
Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

Лабораторна робота *№2*

із дисципліни ***«Методи і технології штучного інтелекту»***

Тема: ***«Моделювання функції з двох змінних засобами нечіткої математики»***

Виконав:

Студент групи ІА-34

Ястремський Богдан

Перевірив:

старший викладач кафедри ІСТ

Польшакова Ольга Михайлівна

Київ — 2025

Тема: Моделювання функції з двох змінних засобами нечіткої математики.

Мета: Промоделювати засобами нечіткої логіки функцію з двох змінних. Провести дослідження форми функції приналежності на якість моделювання.

Примітки: Код усіх файлів буде винесено окремо в додаток А наприкінці звіту.

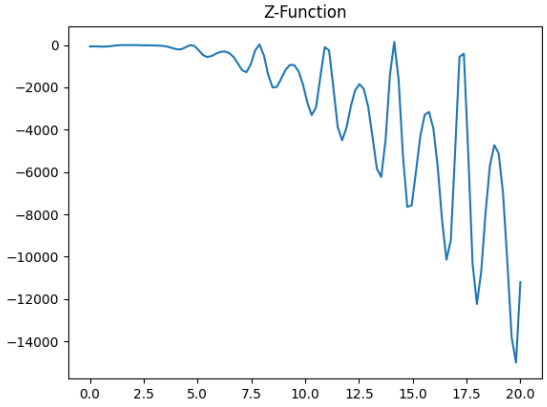
Варіант:

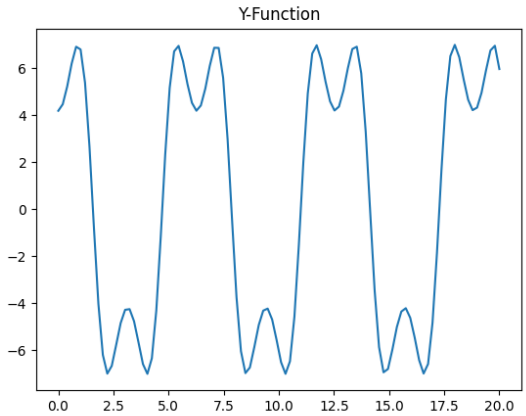
4.	$y = 7 \cdot \sin(5/2 \cdot \cos(x))$
	$z = (x-2)^2 \cdot (1-y^2)$

Хід роботи:

1. Моделювання z(x,y) за трикутною ФП:

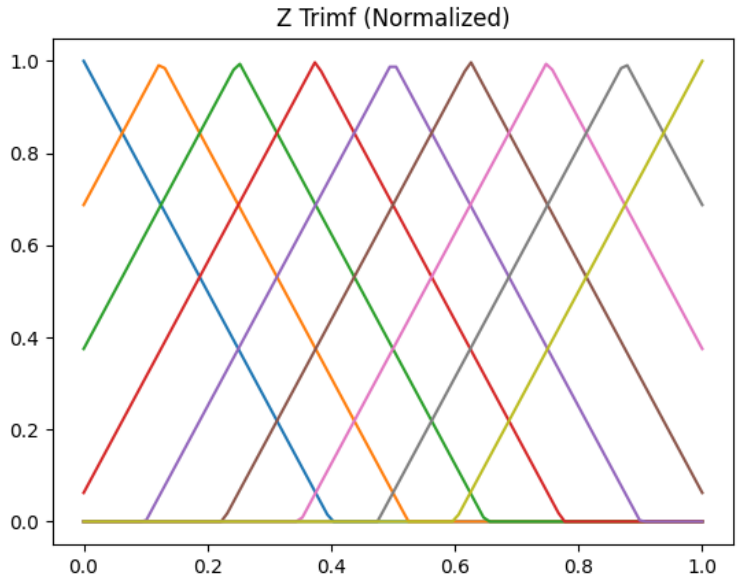
   Спочатку, виведемо ф-ції y(x), z(x,y), які вони є.

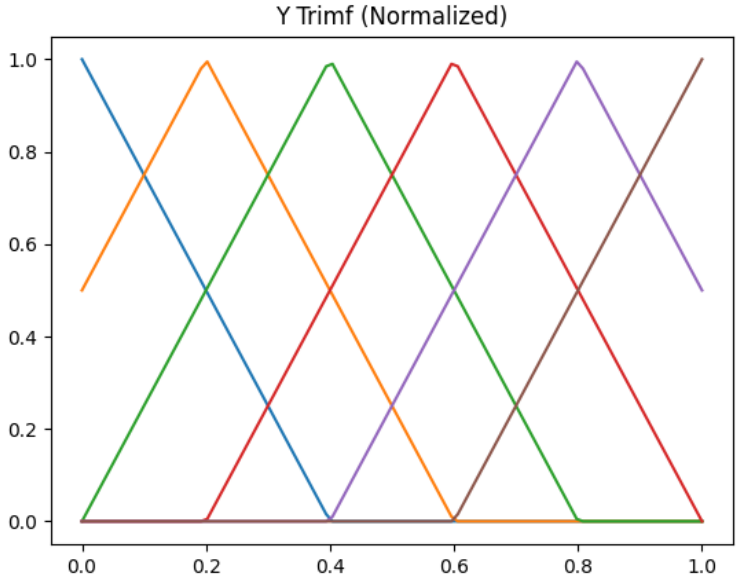


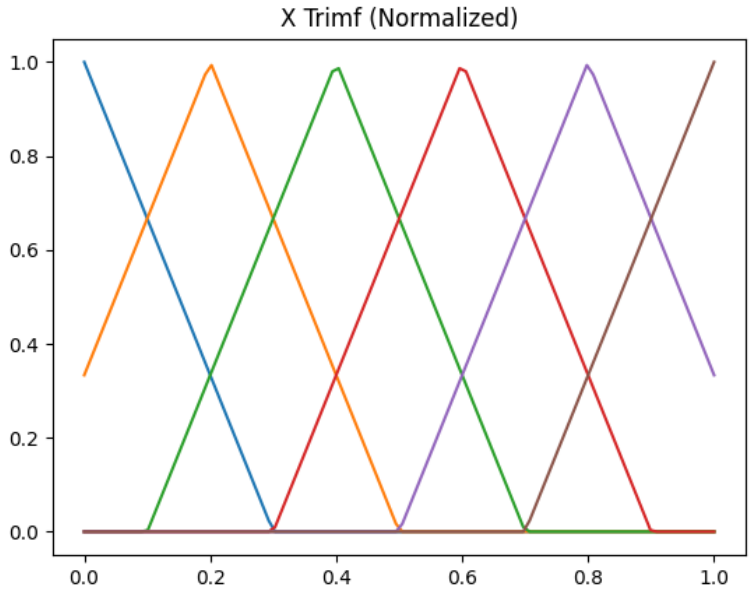


Бачимо, що значення дуже різні за масштабом. Тому необхідно нормалізувати дані за ф-лою: x_n = (x – x_min) / (x_max – x_min). Тобто звузити їх до певного діапазону, щоб моделювання було точніше. В цьому допоможе пакет sklearn.preprocessing та його клас MinMaxScaler. Fit_transform звузить дані до діапазону за замовчуванням – [0; 1]. Reshape – зробить двомірний масив, оскільки зазвичай використовується матриця. Flatten – одномірний.

На основі цих значень будуємо трикутні ФП для входів x, y та виходів z по 6, 6, 9 відповідно.







Далі, треба отримати таблиці значень та імен функцій.

Таблиця значень:

y\x	0	0.2	0.4	0.6	0.8	1
0.0	4	3.24	2.56	1.96	1.44	1
0.2	3.84	3.11	2.46	1.88	1.38	0.96
0.4	3.36	2.72	2.15	1.65	1.21	0.84
0.6	2.56	2.07	1.64	1.25	0.92	0.64
0.8	1.44	1.17	0.92	0.71	0.52	0.36
1.0	0	0	0	0	0	0

Таблиця імен функцій:

y\x	mx1	mx2	mx3	mx4	mx5	mx6
my1	mf9	mf9	mf9	mf9	mf9	mf1
my2	mf9	mf9	mf9	mf9	mf9	mf1
my3	mf9	mf9	mf9	mf9	mf8	mf1
my4	mf9	mf9	mf9	mf9	mf7	mf1
my5	mf9	mf9	mf9	mf8	mf5	mf1
my6	mf9	mf9	mf8	mf6	mf4	mf1

Правила і змодельована за ними ф-ція:

Rules:

If (x is mx1) and (y is mf1) then (f is mf9)

If (x is mx2) and (y is mf2) then (f is mf9)

If (x is mx3) and (y is mf3) then (f is mf9)

If (x is mx4) and (y is mf4) then (f is mf9)

If (x is mx5) and (y is mf5) then (f is mf9)

If (x is mx6) and (y is mf6) then (f is mf1)

If (x is mx1) and (y is mf1) then (f is mf9)

If (x is mx2) and (y is mf2) then (f is mf9)

If (x is mx3) and (y is mf3) then (f is mf9)

If (x is mx4) and (y is mf4) then (f is mf9)

If (x is mx5) and (y is mf5) then (f is mf9)

If (x is mx6) and (y is mf6) then (f is mf1)

If (x is mx1) and (y is mf1) then (f is mf9)

If (x is mx2) and (y is mf2) then (f is mf9)

If (x is mx3) and (y is mf3) then (f is mf9)

If (x is mx4) and (y is mf4) then (f is mf9)

If (x is mx5) and (y is mf5) then (f is mf8)

If (x is mx6) and (y is mf6) then (f is mf1)

If (x is mx1) and (y is mf1) then (f is mf9)

If (x is mx2) and (y is mf2) then (f is mf9)

If (x is mx3) and (y is mf3) then (f is mf9)

If (x is mx4) and (y is mf4) then (f is mf9)

If (x is mx5) and (y is mf5) then (f is mf7)

If (x is mx6) and (y is mf6) then (f is mf1)

If (x is mx1) and (y is mf1) then (f is mf9)

If (x is mx2) and (y is mf2) then (f is mf9)

If (x is mx3) and (y is mf3) then (f is mf9)

If (x is mx4) and (y is mf4) then (f is mf8)

If (x is mx5) and (y is mf5) then (f is mf5)

If (x is mx6) and (y is mf6) then (f is mf1)

If (x is mx1) and (y is mf1) then (f is mf9)

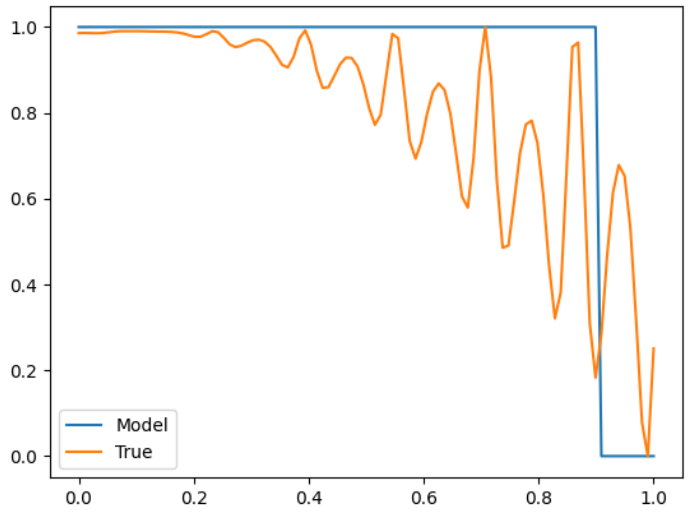
If (x is mx2) and (y is mf2) then (f is mf9)

If (x is mx3) and (y is mf3) then (f is mf8)

If (x is mx4) and (y is mf4) then (f is mf6)

If (x is mx5) and (y is mf5) then (f is mf4)

If (x is mx6) and (y is mf6) then (f is mf1)



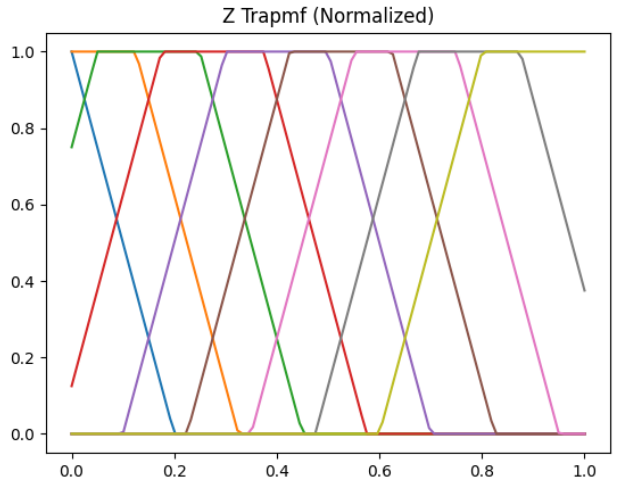
Отримані похибки:

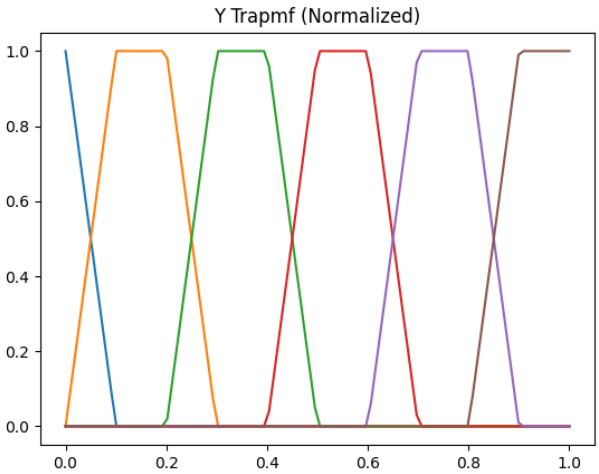
```
Mean Squared Error: 0.07007783263260067
Mean Absolute Error: 0.17417580447383407
```

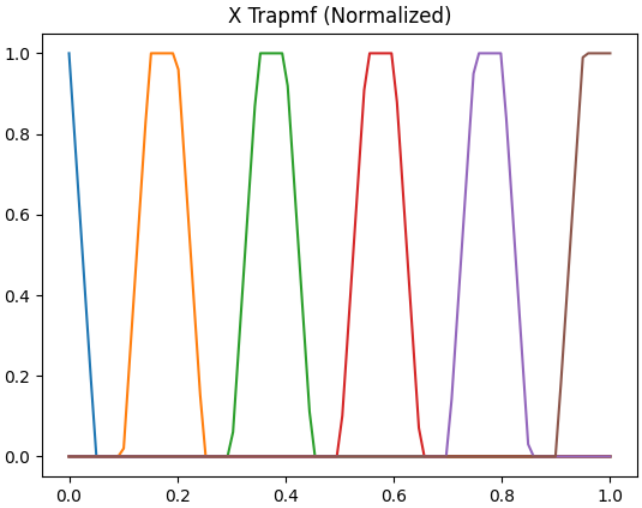
Як бачимо, нормалізація даних дуже сильно допомагає у моделюванні, зменшуючи похибки колосально.

2. Моделювання z(x,y) за трапецієвидною ФП:

   Будуємо трапецієвидні ФП для входів x, y та виходів z по 6, 6, 9 відповідно.







Далі, треба отримати таблиці значень та імен функцій.

Таблиця значень:

y\x	0	0.2	0.4	0.6	0.8	1
0.0	4	3.24	2.56	1.96	1.44	1
0.2	3.84	3.11	2.46	1.88	1.38	0.96
0.4	3.36	2.72	2.15	1.65	1.21	0.84
0.6	2.56	2.07	1.64	1.25	0.92	0.64
0.8	1.44	1.17	0.92	0.71	0.52	0.36
1.0	0	0	0	0	0	0

Таблиця імен функцій:

y\x	mx1	mx2	mx3	mx4	mx5	mx6
my1	mf9	mf9	mf9	mf9	mf9	mf1
my2	mf9	mf9	mf9	mf9	mf9	mf1
my3	mf9	mf9	mf9	mf9	mf9	mf1
my4	mf9	mf9	mf9	mf9	mf7	mf1
my5	mf9	mf9	mf9	mf9	mf6	mf1
my6	mf9	mf9	mf8	mf7	mf4	mf1

Правила і змодельована за ними ф-ція:

Rules:

If (x is mx1) and (y is mf1) then (f is mf9)

If (x is mx2) and (y is mf2) then (f is mf9)

If (x is mx3) and (y is mf3) then (f is mf9)

If (x is mx4) and (y is mf4) then (f is mf9)

If (x is mx5) and (y is mf5) then (f is mf9)

If (x is mx6) and (y is mf6) then (f is mf1)

If (x is mx1) and (y is mf1) then (f is mf9)

If (x is mx2) and (y is mf2) then (f is mf9)

If (x is mx3) and (y is mf3) then (f is mf9)

If (x is mx4) and (y is mf4) then (f is mf9)

If (x is mx5) and (y is mf5) then (f is mf9)

If (x is mx6) and (y is mf6) then (f is mf1)

If (x is mx1) and (y is mf1) then (f is mf9)

If (x is mx2) and (y is mf2) then (f is mf9)

If (x is mx3) and (y is mf3) then (f is mf9)

If (x is mx4) and (y is mf4) then (f is mf9)

If (x is mx5) and (y is mf5) then (f is mf9)

If (x is mx6) and (y is mf6) then (f is mf1)

If (x is mx1) and (y is mf1) then (f is mf9)

If (x is mx2) and (y is mf2) then (f is mf9)

If (x is mx3) and (y is mf3) then (f is mf9)

If (x is mx4) and (y is mf4) then (f is mf9)

If (x is mx5) and (y is mf5) then (f is mf7)

If (x is mx6) and (y is mf6) then (f is mf1)

If (x is mx1) and (y is mf1) then (f is mf9)

If (x is mx2) and (y is mf2) then (f is mf9)

If (x is mx3) and (y is mf3) then (f is mf9)

If (x is mx4) and (y is mf4) then (f is mf9)

If (x is mx5) and (y is mf5) then (f is mf6)

If (x is mx6) and (y is mf6) then (f is mf1)

If (x is mx1) and (y is mf1) then (f is mf9)

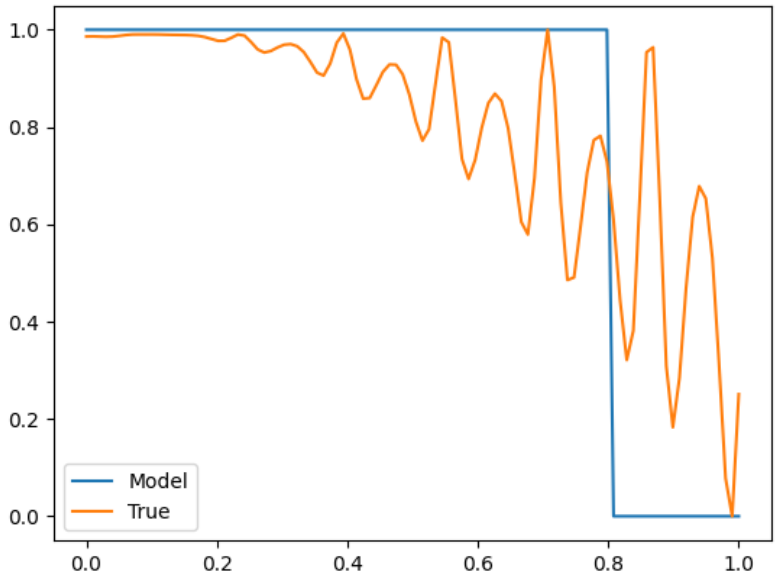
If (x is mx2) and (y is mf2) then (f is mf9)

If (x is mx3) and (y is mf3) then (f is mf8)

If (x is mx4) and (y is mf4) then (f is mf7)

If (x is mx5) and (y is mf5) then (f is mf4)

If (x is mx6) and (y is mf6) then (f is mf1)



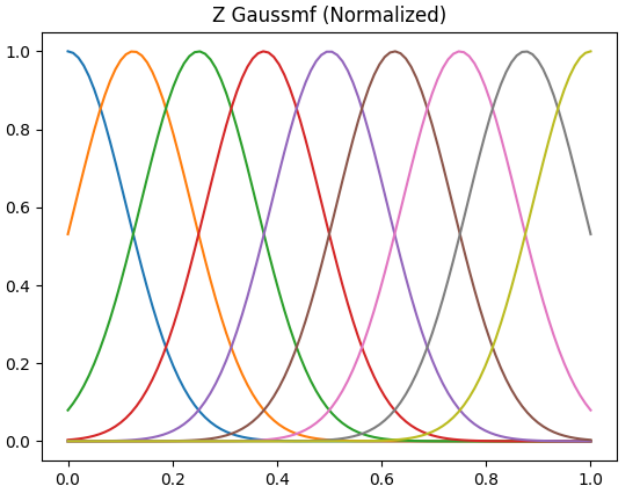
Отримані похибки:

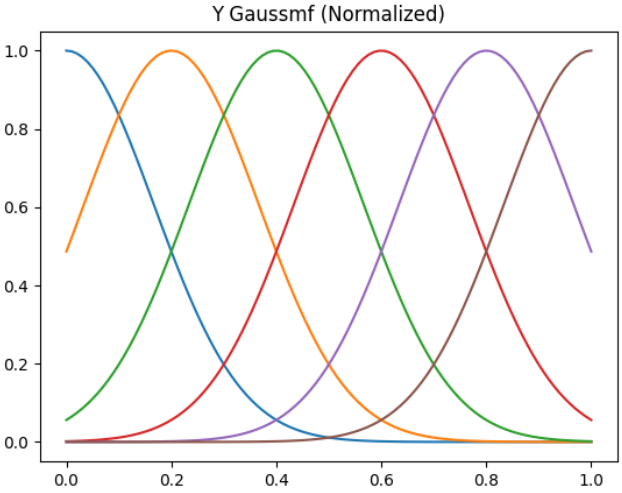
```
Mean Squared Error: 0.07971176332221841
Mean Absolute Error: 0.18380973516345187
```

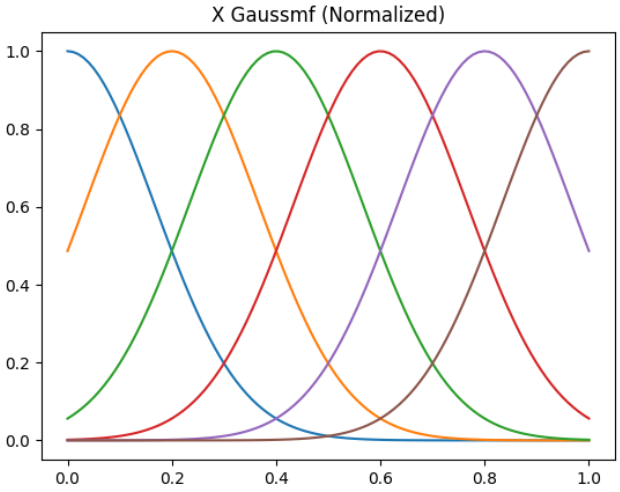
Як бачимо, отримані похибки трохи більше, ніж в трикутній, але не суттєво, тобто різниці особливої між моделюванням за трикутною і трапецієвидною немає. Єдиний момент – підбір параметрів.

3. Моделювання z(x,y) за гаусівською ФП:

   Для неї потрібен параметр сигма, для прикладу візьмемо середнє значення всіх входів і виходів.







Далі, треба отримати таблиці значень та імен функцій.

Таблиця значень:

y\x	0	0.2	0.4	0.6	0.8	1
0.0	4	3.24	2.56	1.96	1.44	1
0.2	3.84	3.11	2.46	1.88	1.38	0.96
0.4	3.36	2.72	2.15	1.65	1.21	0.84
0.6	2.56	2.07	1.64	1.25	0.92	0.64
0.8	1.44	1.17	0.92	0.71	0.52	0.36
1.0	0	0	0	0	0	0

Таблиця імен функцій:

y\x	mx1	mx2	mx3	mx4	mx5	mx6
my1	mf9	mf9	mf9	mf9	mf9	mf1
my2	mf9	mf9	mf9	mf9	mf9	mf1
my3	mf9	mf9	mf9	mf9	mf8	mf1
my4	mf9	mf9	mf9	mf9	mf7	mf1
my5	mf9	mf9	mf9	mf8	mf5	mf1
my6	mf9	mf9	mf8	mf6	mf4	mf1

Правила та змодельована за ними функція:

Rules:

If (x is mx1) and (y is mf1) then (f is mf9)

If (x is mx2) and (y is mf2) then (f is mf9)

If (x is mx3) and (y is mf3) then (f is mf9)

If (x is mx4) and (y is mf4) then (f is mf9)

If (x is mx5) and (y is mf5) then (f is mf9)

If (x is mx6) and (y is mf6) then (f is mf1)

If (x is mx1) and (y is mf1) then (f is mf9)

If (x is mx2) and (y is mf2) then (f is mf9)

If (x is mx3) and (y is mf3) then (f is mf9)

If (x is mx4) and (y is mf4) then (f is mf9)

If (x is mx5) and (y is mf5) then (f is mf9)

If (x is mx6) and (y is mf6) then (f is mf1)

If (x is mx1) and (y is mf1) then (f is mf9)

If (x is mx2) and (y is mf2) then (f is mf9)

If (x is mx3) and (y is mf3) then (f is mf9)

If (x is mx4) and (y is mf4) then (f is mf9)

If (x is mx5) and (y is mf5) then (f is mf8)

If (x is mx6) and (y is mf6) then (f is mf1)

If (x is mx1) and (y is mf1) then (f is mf9)

If (x is mx2) and (y is mf2) then (f is mf9)

If (x is mx3) and (y is mf3) then (f is mf9)

If (x is mx4) and (y is mf4) then (f is mf9)

If (x is mx5) and (y is mf5) then (f is mf7)

If (x is mx6) and (y is mf6) then (f is mf1)

If (x is mx1) and (y is mf1) then (f is mf9)

If (x is mx2) and (y is mf2) then (f is mf9)

If (x is mx3) and (y is mf3) then (f is mf9)

If (x is mx4) and (y is mf4) then (f is mf8)

If (x is mx5) and (y is mf5) then (f is mf5)

If (x is mx6) and (y is mf6) then (f is mf1)

If (x is mx1) and (y is mf1) then (f is mf9)

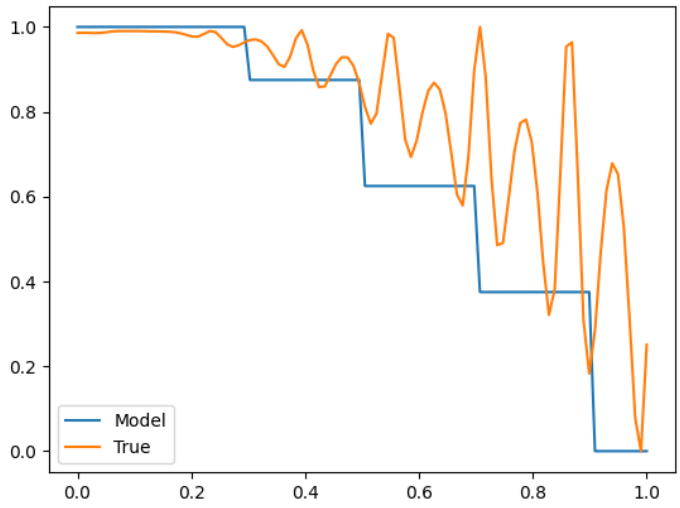
If (x is mx2) and (y is mf2) then (f is mf9)

If (x is mx3) and (y is mf3) then (f is mf8)

If (x is mx4) and (y is mf4) then (f is mf6)

If (x is mx5) and (y is mf5) then (f is mf4)

If (x is mx6) and (y is mf6) then (f is mf1)



Отримані помилки:

```
Mean Squared Error: 0.052063639951398885
Mean Absolute Error: 0.1470454157068706
```

Видно, що похибки ще менше, отже моделювання за ФП Гауса дало найкращі результати.

Висновок: на даному лабораторному занятті я познайомився з моделюванням функцій двох змінних методами нечіткої математики, а саме: трикутною ФП, трапецієвидною ФП та ФП Гауса. Також, я познайомився з поняттям нормалізації даних, та чому це дуже важливо для моделювання, і як саме воно впливає на отримувані похибки. Більше того, я довів, чому саме моделювання за ФП Гауса дає найкращі результати. Зменшення числа правил у таблиці може бути реалізовано за попереднім переглядом отриманої таблиці і виявленням закономірностей їх появи (наприклад: якщо mx1, то mf1 завжди, не важливо яка my)

Відповіді на контрольні питання:

1. Що таке нечітка змінна і чим вона відрізняється від звичайної (чіткої) змінної?

   Нечітка змінна - це змінна, значення якої не є чітким і визначеним.

   Натомість значення нечіткої змінної виражаються через ступінь належності до певної множини, зазвичай у вигляді функцій належності.

   Чітка змінна – містить точний результат виміру.

2. Які складові має нечітка змінна (базова область, терм-множина, функції приналежності)?

   Базова область – всі чіткі значення, які може приймати змінна. Наприклад, швидкість руху від 30 до 90 км/год.

   Терм-множина - це множина нечітких термінів або категорій, які описують змінну. Наприклад, ДУЖЕ ПОВІЛЬНО, ПОВІЛЬНО, ПЛАВНО, ШВИДКО, ДУЖЕ ШВИДКО.

   Функція приналежності - це функція, яка визначає ступінь належності елемента до конкретної терм-множини. Вона дає значення від 0 до 1, яке показує, наскільки сильно конкретне значення належить до множини.

3. Що таке терм-множина та яке її призначення?

Терм-множина - це множина нечітких термінів або категорій, які описують змінну. Наприклад, ДУЖЕ ПОВІЛЬНО, ПОВІЛЬНО, ПЛАВНО, ШВИДКО, ДУЖЕ ШВИДКО.

4. Наведіть приклад нечіткої змінної з повсякденного життя.
   Температура, вологість, швидкість.

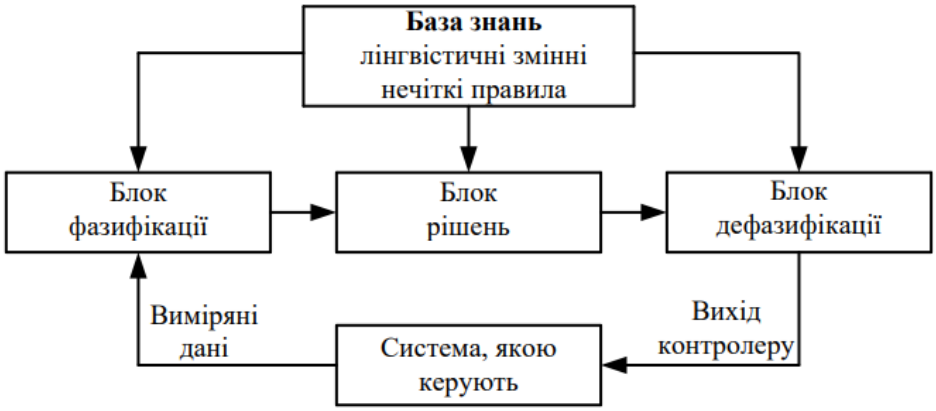
5. Яким чином описуються значення нечіткої змінної?
   Через функції приналежності.

6. Що таке лінгвістична змінна?
   Лінгвістична змінна - це змінна, яка приймає значення, що є не числовими, а мовними термінами.

7. Яка різниця між нечіткою та лінгвістичною змінною?
   Нечітка змінна виражається ФП, лінгвістична – словесно.

8. Що таке система нечіткого виводу?
   Система нечіткого виводу - це математична модель або комп'ютерна система, яка використовується для прийняття рішень або виведення результатів на основі нечітких даних.

9. Які основні етапи роботи системи нечіткого виводу?
   Блок фазіфікації перетворює чіткі величини, виміряні на виході об'єкта керування, у нечіткі величини, що описані лінгвістичними змінними в базі знань. Блок рішень використовує нечіткі умовні (if - then) правила, закладені в базі знань, для перетворення нечітких вхідних даних у необхідні керуючі впливи, що носять також нечіткий характер. Блок дефазіфікації перетворює нечіткі дані з виходу блоку рішень у чітку величину, що використовується для керування об'єктом.



10. Які етапи входять у процес фазифікації?

    Блок фазіфікації перетворює чіткі величини, виміряні на виході об'єкта керування, у нечіткі величини, що описані лінгвістичними змінними в базі знань.

11. Що таке база правил у системі нечіткого виводу і як вона формується?

    База правил у системі нечіткого виводу - це набір нечітких правил, які визначають взаємозв'язок між вхідними і вихідними змінними. Кожне правило має форму якщо-то (англ. if-then) і описує, що робити в залежності від значень вхідних змінних.

    Формуються через визначення найвищої ординати на виходах ФП, беручи за аргумент добуток входів.

12. Яка роль механізму нечіткого логічного висновку?

    Механізм нечіткого логічного висновку у системі нечіткого виводу виконує функцію обробки нечітких правил та обчислення нечітких вихідних значень на основі вхідних даних і заданих правил.

13. Що таке дефазифікація і для чого вона потрібна?

    Перетворює нечіткі дані з виходу блоку рішень у чітку величину, що використовується для керування об'єктом.

14. Які найбільш поширені методи дефазифікації (наприклад, центр ваги, максимум)?

    Центр мас, максимальне значення, середнє значення.

15. У яких практичних сферах найчастіше застосовуються системи нечіткого виводу?

    Автоматика, робототехніка, інженерія, і т.д.

ДОДАТОК А.

/triangular_model.py

```
import matplotlib.pyplot as plt
import numpy as np
import myskfuzzy as fuzz
from sklearn.metrics import mean_absolute_error, mean_squared_error
from sklearn.preprocessing import MinMaxScaler
from tabulate import tabulate

# Setting functions values
x_values = np.linspace(0, 20, 100)
y_values = 7 * np.sin(2.5 * np.cos(x_values))
z_values = (x_values - 2) ** 2 * (1 - y_values**2)

# Data normalizing for lowering the value of MSE/MAE due to function value
scaling
scaler_x = MinMaxScaler()
scaler_y = MinMaxScaler()
scaler_z = MinMaxScaler()

x_values_norm = scaler_x.fit_transform(x_values.reshape(-1, 1)).flatten()
y_values_norm = scaler_y.fit_transform(y_values.reshape(-1, 1)).flatten()
z_values_norm = scaler_z.fit_transform(z_values.reshape(-1, 1)).flatten()

# Building Y/Z-Functions
plt.plot(x_values, y_values)
plt.title("Y-Function")
plt.show()
plt.plot(x_values, z_values)
plt.title("Z-Function")
plt.show()


# Getting 6 inputs for x and y, and 9 outputs for z. Building plots
x_means = np.linspace(min(x_values_norm), max(x_values_norm), 6)
y_means = np.linspace(min(y_values_norm), max(y_values_norm), 6)
z_means = np.linspace(min(z_values_norm), max(z_values_norm), 9)

mx = [fuzz.trimf(x_values_norm,
                 [x_means[i] - 0.3, x_means[i], x_means[i] + 0.3]) for i in
range(6)]

my = [fuzz.trimf(np.linspace(min(y_values_norm), max(y_values_norm), 100),
                 [y_means[i] - 0.4, y_means[i], y_means[i] + 0.4]) for i in
range(6)]

mf = [fuzz.trimf(np.linspace(min(z_values_norm), max(z_values_norm), 100),
                 [z_means[i] - 0.4, z_means[i], z_means[i] + 0.4]) for i in
range(9)]

for i in range(6):
    plt.plot(x_values_norm, mx[i])
plt.title("X Trimf (Normalized)")
plt.show()

for i in range(6):
    plt.plot(np.linspace(min(y_values_norm), max(y_values_norm), 100), my[i])
plt.title("Y Trimf (Normalized)")
plt.show()

for i in range(9):
```

```
    plt.plot(np.linspace(min(z_values_norm), max(z_values_norm), 100), mf[i])
plt.title("Z Trimf (Normalized)")
plt.show()


# Tables of values and mfs
table = [["y\\x"] + [i for i in x_means]]
for y_mean in y_means:
    row = [round(y_mean, 2)]
    for x_mean in x_means:
        z_mean = (x_mean - 2) ** 2 * (1 - y_mean**2)
        row.append(round(z_mean, 2))
    table.append(row)

print(tabulate(table, tablefmt="grid"))

def get_biggest_ordinate(argument, argument_means, d):
    best_mf_index = -1
    best_value = -float("inf")

    for index, value in enumerate(argument_means):
        ff = fuzz.trimf([argument], [value - d, value, value + d])
        if ff > best_value:
            best_mf_index = index
            best_value = ff

    return best_mf_index


table_new = [["y\\x"] + ["mx" + str(i) for i in range(1, 7)]]
rules = {}
for i in range(6):
    row = ["my" + str(i+1)]
    for j in range(6):
        z = (x_means[i] - 2) ** 2 * (1 - y_means[j]**2)
        best_mf = get_biggest_ordinate(z, z_means, 4)
        row.append("mf" + str(best_mf + 1))
        rules[(j, i)] = best_mf
    table_new.append(row)

print(tabulate(table_new, tablefmt="grid"))


# Rules
print("\nRules: ")

for rule in rules.keys():
    print(f"If (x is mx{rule[0] + 1}) and (y is mf{rule[0] + 1}) then (f is
mf{rules[rule]+1})")

# Modeling function
model = []
for x in x_values_norm:
    best_x_func = get_biggest_ordinate(x, x_means, 1)
    best_y_func = get_biggest_ordinate(7 * np.sin(2.5 * np.cos(x)), y_means,
1)
    best_z_func = rules[(best_x_func, best_y_func)]
    model.append(z_means[best_z_func])

plt.plot(x_values_norm, model, label="Model")
plt.plot(x_values_norm, z_values_norm, label="True")
plt.legend()
plt.show()
```

```
print(f"Mean Squared Error: {mean_squared_error(z_values_norm, model)}")
print(f"Mean Absolute Error: {mean_absolute_error(z_values_norm, model)}")
```

/trapezoidal_model.py

```
import matplotlib.pyplot as plt
import numpy as np
import myskfuzzy as fuzz
from sklearn.metrics import mean_absolute_error, mean_squared_error
from sklearn.preprocessing import MinMaxScaler
from tabulate import tabulate

# Setting functions values
x_values = np.linspace(0, 20, 100)
y_values = 7 * np.sin(2.5 * np.cos(x_values))
z_values = (x_values - 2) ** 2 * (1 - y_values**2)

# Data normalizing for lowering the value of MSE/MAE due to function value
scaling
scaler_x = MinMaxScaler()
scaler_y = MinMaxScaler()
scaler_z = MinMaxScaler()

x_values_norm = scaler_x.fit_transform(x_values.reshape(-1, 1)).flatten()
y_values_norm = scaler_y.fit_transform(y_values.reshape(-1, 1)).flatten()
z_values_norm = scaler_z.fit_transform(z_values.reshape(-1, 1)).flatten()

x_means = np.linspace(min(x_values_norm), max(x_values_norm), 6)
y_means = np.linspace(min(y_values_norm), max(y_values_norm), 6)
z_means = np.linspace(min(z_values_norm), max(z_values_norm), 9)


# Getting 6 inputs for x and y, and 9 outputs for z. Building plots
mx = [fuzz.trapmf(x_values_norm,
                  [x_means[i] - 0.1, x_means[i] - 0.05, x_means[i],
x_means[i] + 0.05]) for i in range(6)]

my = [fuzz.trapmf(np.linspace(min(y_values_norm), max(y_values_norm), 100),
                  [y_means[i] - 0.2, y_means[i] - 0.1, y_means[i], y_means[i]
+ 0.1]) for i in range(6)]

mf = [fuzz.trapmf(np.linspace(min(z_values_norm), max(z_values_norm), 100),
                  [z_means[i] - 0.4, z_means[i] - 0.2, z_means[i], z_means[i]
+ 0.2]) for i in range(9)]

for i in range(6):
    plt.plot(x_values_norm, mx[i])
plt.title("X Trapmf (Normalized)")
plt.show()

for i in range(6):
    plt.plot(np.linspace(min(y_values_norm), max(y_values_norm), 100), my[i])
plt.title("Y Trapmf (Normalized)")
plt.show()

for i in range(9):
    plt.plot(np.linspace(min(z_values_norm), max(z_values_norm), 100), mf[i])
plt.title("Z Trapmf (Normalized)")
plt.show()


# Tables of values and mfs
```

```
table = [["y\\x"] + [round(i, 2) for i in x_means]]
for y_mean in y_means:
    row = [round(y_mean, 2)]
    for x_mean in x_means:
        z_mean = (x_mean - 2) ** 2 * (1 - y_mean**2)
        row.append(round(z_mean, 2))
    table.append(row)

print(tabulate(table, tablefmt="grid"))

def get_biggest_ordinate(argument, argument_means, d):
    best_mf_index = -1
    best_value = -float("inf")

    for index, value in enumerate(argument_means):
        ff = fuzz.trapmf([argument], [value - 2 * d, value - d, value, value
+ d])
        if ff > best_value:
            best_mf_index = index
            best_value = ff

    return best_mf_index


table_new = [["y\\x"] + ["mx" + str(i) for i in range(1, 7)]]
rules = {}
for i in range(6):
    row = ["my" + str(i+1)]
    for j in range(6):
        z = (x_means[i] - 2) ** 2 * (1 - y_means[j]**2)
        best_mf = get_biggest_ordinate(z, z_means, 10)
        row.append("mf" + str(best_mf + 1))
        rules[(j, i)] = best_mf
    table_new.append(row)

print(tabulate(table_new, tablefmt="grid"))


# Rules
print("\nRules: ")
for rule in rules.keys():
    print(f"If (x is mx{rule[0] + 1}) and (y is mf{rule[0] + 1}) then (f is
mf{rules[rule]+1})")


# Modeling function
model = []
for x in x_values_norm:
    best_x_func = get_biggest_ordinate(x, x_means, 1)
    best_y_func = get_biggest_ordinate(7 * np.sin(2.5 * np.cos(x)), y_means,
1)
    best_z_func = rules[(best_x_func, best_y_func)]
    model.append(z_means[best_z_func])

plt.plot(x_values_norm, model, label="Model")
plt.plot(x_values_norm, z_values_norm, label="True")
plt.legend()
plt.show()

print(f"Mean Squared Error: {mean_squared_error(z_values_norm, model)}")
print(f"Mean Absolute Error: {mean_absolute_error(z_values_norm, model)}")
```

/gauss_model.py

```
import matplotlib.pyplot as plt
import numpy as np
import myskfuzzy as fuzz
from sklearn.metrics import mean_absolute_error, mean_squared_error
from sklearn.preprocessing import MinMaxScaler
from tabulate import tabulate

# Setting functions values
x_values = np.linspace(0, 20, 100)
y_values = 7 * np.sin(2.5 * np.cos(x_values))
z_values = (x_values - 2) ** 2 * (1 - y_values**2)

# Data normalizing for lowering the value of MSE/MAE due to function value
scaling
scaler_x = MinMaxScaler()
scaler_y = MinMaxScaler()
scaler_z = MinMaxScaler()

x_values_norm = scaler_x.fit_transform(x_values.reshape(-1, 1)).flatten()
y_values_norm = scaler_y.fit_transform(y_values.reshape(-1, 1)).flatten()
z_values_norm = scaler_z.fit_transform(z_values.reshape(-1, 1)).flatten()

x_means = np.linspace(min(x_values_norm), max(x_values_norm), 6)
y_means = np.linspace(min(y_values_norm), max(y_values_norm), 6)
z_means = np.linspace(min(z_values_norm), max(z_values_norm), 9)

# Getting sigmas 6 inputs for x and y, and 9 outputs for z. Building plots.
x_sigma = (max(x_values_norm) - min(x_values_norm)) / 6
y_sigma = (max(y_values_norm) - min(y_values_norm)) / 6
z_sigma = (max(z_values_norm) - min(z_values_norm)) / 9

mx = [fuzz.gaussmf(x_values_norm, x_means[i], x_sigma) for i in range(6)]
my = [fuzz.gaussmf(np.linspace(min(y_values_norm), max(y_values_norm), 100),
y_means[i], y_sigma) for i in range(6)]
mf = [fuzz.gaussmf(np.linspace(min(z_values_norm), max(z_values_norm), 100),
z_means[i], z_sigma) for i in range(9)]

for i in range(6):
    plt.plot(x_values_norm, mx[i])
plt.title("X Gaussmf (Normalized)")
plt.show()

for i in range(6):
    plt.plot(np.linspace(min(y_values_norm), max(y_values_norm), 100), my[i])
plt.title("Y Gaussmf (Normalized)")
plt.show()

for i in range(9):
    plt.plot(np.linspace(min(z_values_norm), max(z_values_norm), 100), mf[i])
plt.title("Z Gaussmf (Normalized)")
plt.show()


# Tables of values and mfs
table = [["y\\x"] + [round(i, 2) for i in x_means]]
for y_mean in y_means:
    row = [round(y_mean, 2)]
    for x_mean in x_means:
        z_mean = (x_mean - 2) ** 2 * (1 - y_mean**2)
        row.append(round(z_mean, 2))
    table.append(row)
```

```
print(tabulate(table, tablefmt="grid"))

def get_biggest_ordinate(argument, argument_means, sigma):
    best_mf_index = -1
    best_value = -float("inf")

    for index, value in enumerate(argument_means):
        ff = fuzz.gaussmf([argument], value, sigma)
        if ff > best_value:
            best_mf_index = index
            best_value = ff

    return best_mf_index

table_new = [["y\\x"] + ["mx" + str(i) for i in range(1, 7)]]
rules = {}
for i in range(6):
    row = ["my" + str(i+1)]
    for j in range(6):
        z = (x_means[i] - 2) ** 2 * (1 - y_means[j]**2)
        best_mf = get_biggest_ordinate(z, z_means, z_sigma)
        row.append("mf" + str(best_mf + 1))
        rules[(j, i)] = best_mf
    table_new.append(row)

print(tabulate(table_new, tablefmt="grid"))


# Rules
print("\nRules: ")
for rule in rules.keys():
    print(f"If (x is mx{rule[0] + 1}) and (y is mf{rule[0] + 1}) then (f is
mf{rules[rule]+1})")

# Modeling function
model = []
for x in x_values_norm:
    best_x_func = get_biggest_ordinate(x, x_means, x_sigma)
    best_y_func = get_biggest_ordinate(7 * np.sin(2.5 * np.cos(x)), y_means,
y_sigma)
    best_z_func = rules[(best_x_func, best_y_func)]
    model.append(z_means[best_z_func])

plt.plot(x_values_norm, model, label="Model")
plt.plot(x_values_norm, z_values_norm, label="True")
plt.legend()
plt.show()

print(f"Mean Squared Error: {mean_squared_error(z_values_norm, model)}")
print(f"Mean Absolute Error: {mean_absolute_error(z_values_norm, model)}")
```