

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО”
ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

Кафедра інформатики та програмної інженерії

Звіт до лабораторної роботи №1

з курсу

«Методи ІІІ»

студента 2 курсу
групи ІТ-02
Макарова Іллі Сергійовича

Викладач:

Київ – 2022

ЛАБОРАТОРНА РОБОТА №1

Дослідження способів формування нечітких множин і операцій над ними

Мета роботи: Побудувати нечіткі множини з використанням різних типів функцій приналежності. Виконати найбільш поширені логічні операції над нечіткими множинами.

ЗАВДАННЯ

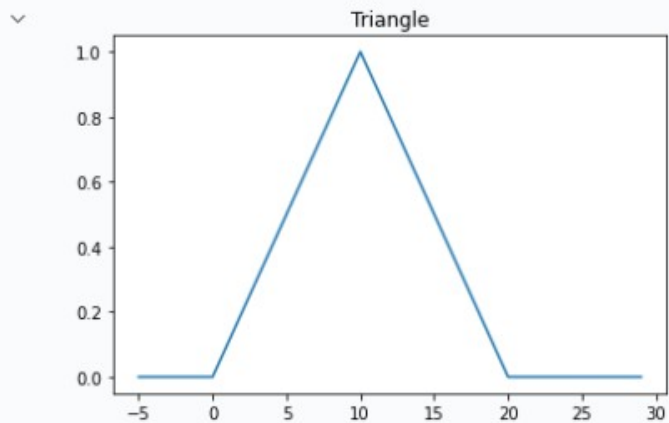
За допомогою пакетів моделювання або мови програмування високого рівня:

1. Побудувати трикутну і трапецієподібну функцію приналежності.
2. Побудувати просту і двосторонню функцію приналежності Гаусса, утворену за допомогою різних функцій розподілу.
3. Побудувати функцію приналежності "узагальнений дзвін", яка дозволяє представляти нечіткі суб'єктивні переваги.
4. Побудувати набір сігмоїдних функцій: основну односторонню, яка відкрита зліва чи справа; додаткову двосторонню; додаткову несиметричну.
5. Побудувати набір поліноміальних функцій приналежності (Z-, PI- і S-функцій).
6. Побудувати мінімаксну інтерпретацію логічних операторів з використанням операцій пошуку мінімуму і максимуму.
7. Побудувати вірогідну інтерпретацію кон'юнктивну і диз'юнктивних операторів.
8. Побудувати доповнення нечіткої множини, яке описує деяке розмите судження і представляє собою математичний опис вербального вираження, який заперечує це нечітка множина.
При виконанні пунктів 1 - 8 індивідуального завдання, значення змінних a , b , c , d і т.д. необхідно вибирати довільним чином.
9. Оформіть звіт по лабораторній роботі.

ВИКОНАННЯ

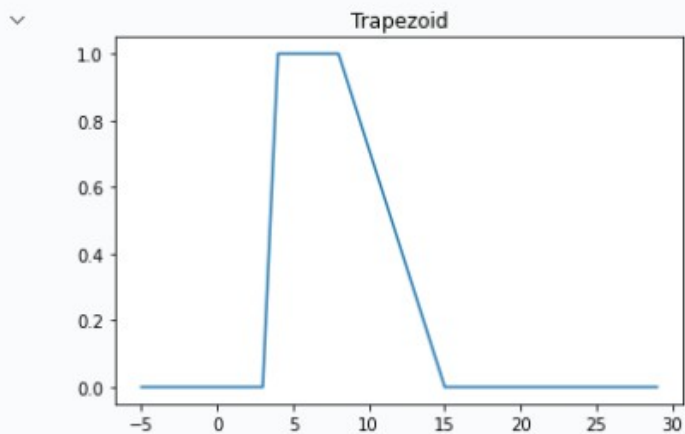
Трикутна

```
64 1 x = np.arange(-5, 30)
    2 plt.plot(x, skfuzzy.trimf(x, [0, 10, 20]))
    3 plt.title("Triangle")
    4 plt.show()
```



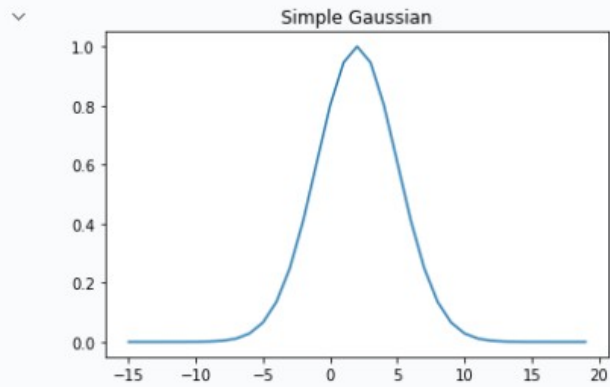
Трацеподібна

```
1 19 1 x = np.arange(-5, 30)
    2 plt.plot(x, skfuzzy.trapmf(x, [3, 4, 8, 15]))
    3 plt.title("Trapezoid")
    4 plt.show()
```



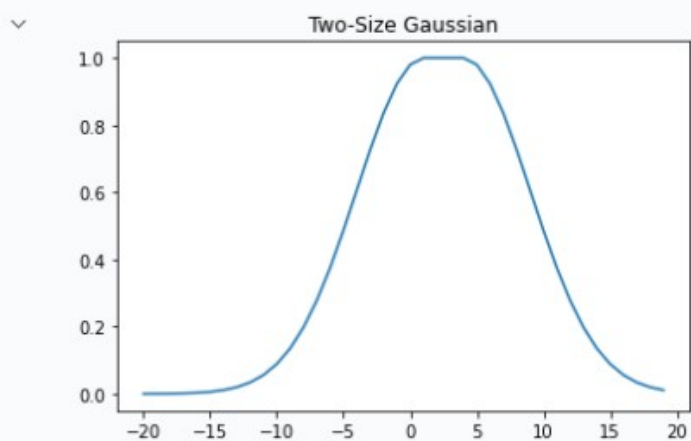
Проста Гаусса

```
In 25 1 x = np.arange(-15, 20)
      2 plt.plot(x, skfuzzy.gaussmf(x, 2, 3))
      3 plt.title("Simple Gaussian")
      4 plt.show()
```



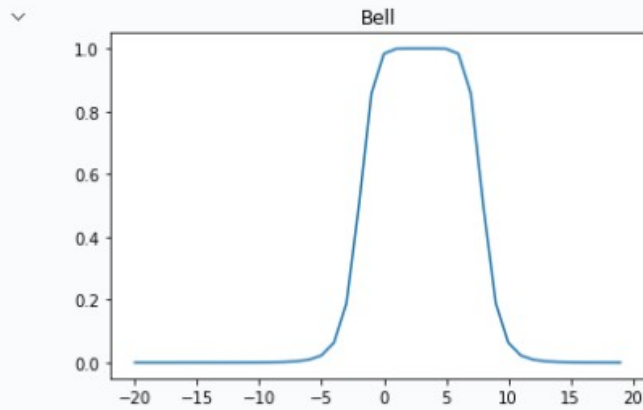
Двостороння Гаусса

```
n 24 1 x = np.arange(-20, 20)
      2 plt.plot(x, skfuzzy.gauss2mf(x, 1, 5, 4, 5))
      3 plt.title("Two-Size Gaussian")
      4 plt.show()
```



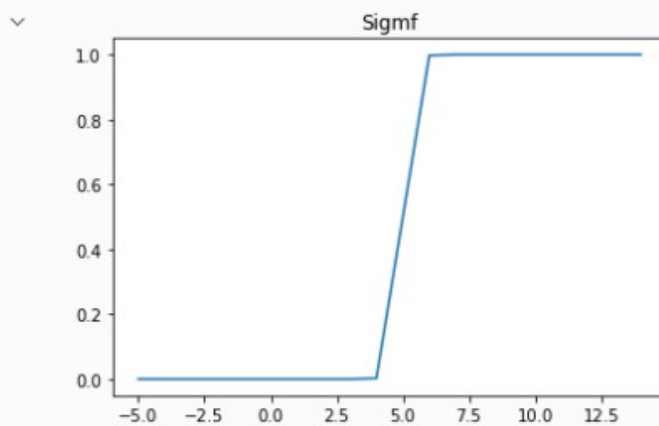
Узагальнений дзвін

```
In 65 1 x = np.arange(-20, 20)
      2 plt.plot(x, skfuzzy.gbellmf(x, 5, 4, 3))
      3 plt.title("Bell")
      4 plt.show()
```

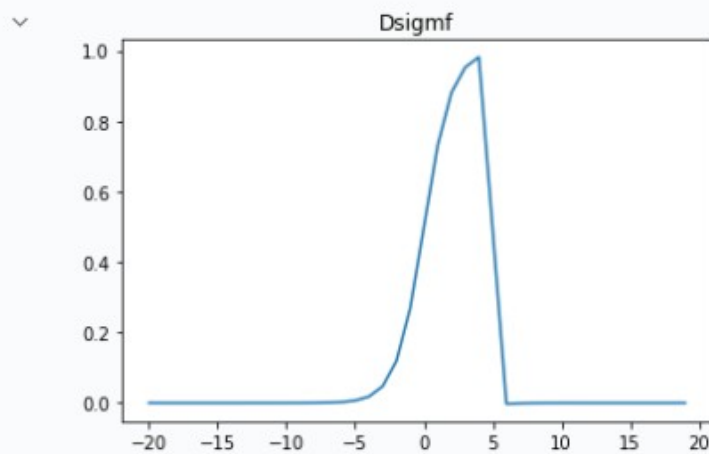


Набір сігмоїдних функцій

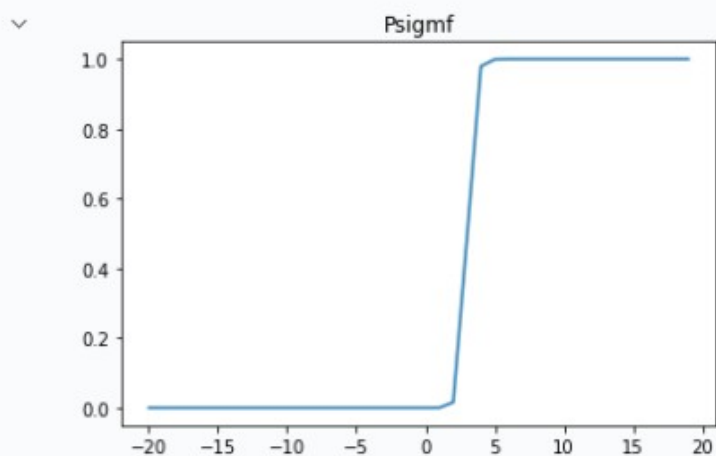
```
46 1 x = np.arange(-5, 15)
    2 plt.plot(x, skfuzzy.sigmf(x, 5, 6))
    3 plt.title("Sigmf")
    4 plt.show()
```



```
In 69 1 x = np.arange(-20, 20)
      2 plt.plot(x, skfuzzy.dsigmf(x, 0, 1, 5, 10))
      3 plt.title("Dsigmf")
      4 plt.show()
```

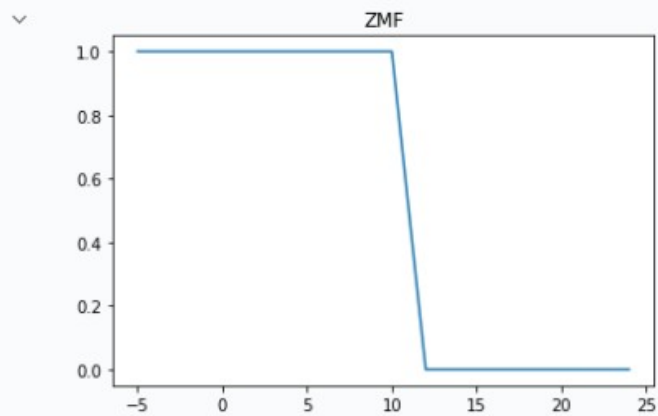


```
In 48 1 x = np.arange(-20, 20)
      2 plt.plot(x, skfuzzy.psigmf(x, 1, 2, 3, 4))
      3 plt.title("Psigmf")
      4 plt.show()
```



Набір поліноміальних приналежностей

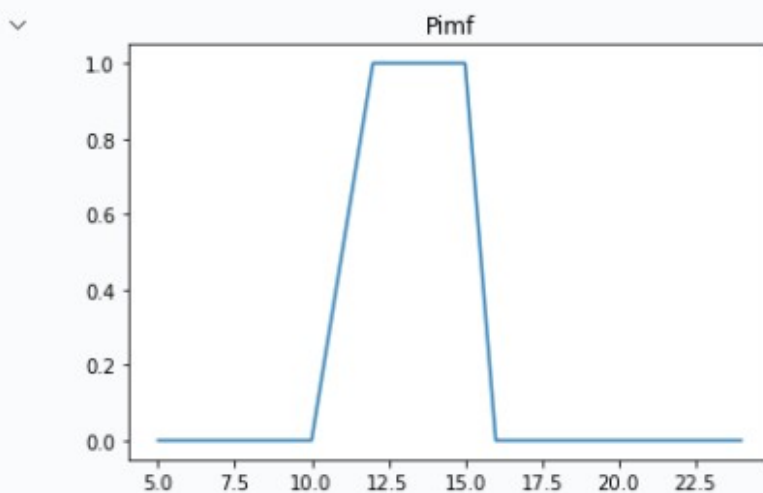
```
In 63 1 x = np.arange(-5, 25)
      2 plt.plot(x, skfuzzy.zmf(x, 10, 12))
      3 plt.title("ZMF")
      4 plt.show()
```



```
n 58 1 x = np.arange(5, 25)
      2 plt.plot(x, skfuzzy.pimf(x, 10, 12, 15, 16))
      3 plt.title("Pimf")
      4 plt.plot()
```

t 58

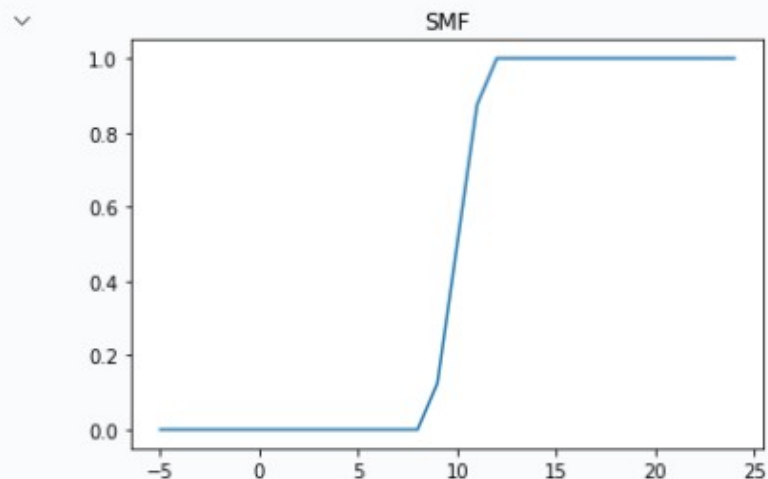
[]



```

62 1 x = np.arange(-5, 25)
    2 plt.plot(x, skfuzzy.smf(x, 8, 12))
    3 plt.title("SMF")
    4 plt.show()

```

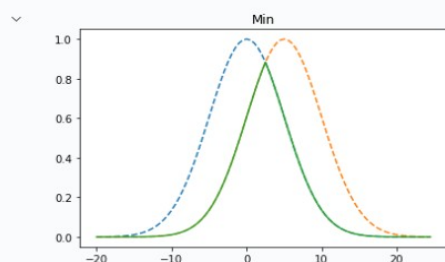


Мінімаксна інтерпретація логічних операторів з використанням операцій пошуку мінімуму і максимуму

```

in 97 1 def min_f(x, f1, f2):
    2     x_min, y_min = skfuzzy.fuzzy_and(x, f1, x, f2)
    3     plt.plot(x, f1, linestyle='--')
    4     plt.plot(x, f2, linestyle='--')
    5     plt.plot(x_min, y_min)
    6     plt.title(f"Min")
    7     plt.show()
    8
    9
   10 x = np.arange(-20, 25, 0.5)
   11 min_f(x, skfuzzy.gaussmf(x, 0, 5), skfuzzy.gaussmf(x, 5, 5))

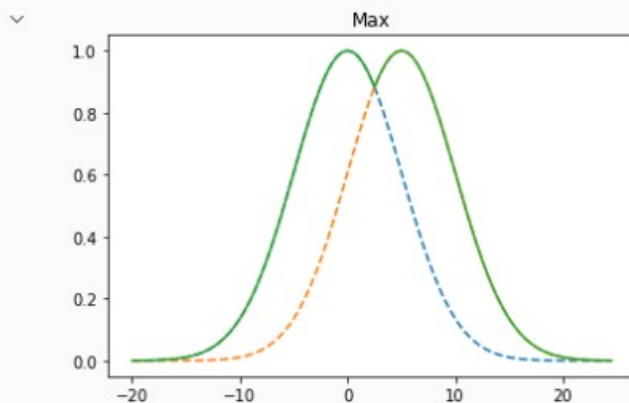
```




```

In 99 1 def max_f(x, f1, f2):
      2     x_max, y_max = skfuzzy.fuzzy_or(x, f1, x, f2)
      3     plt.plot(x, f1, linestyle='--')
      4     plt.plot(x, f2, linestyle='--')
      5     plt.plot(x_max, y_max)
      6     plt.title('Max')
      7     plt.show()
      8
      9
     10 x = np.arange(-20, 25, 0.5)
     11 max_f(x, skfuzzy.gaussmf(x, 0, 5), skfuzzy.gaussmf(x, 5, 5))

```

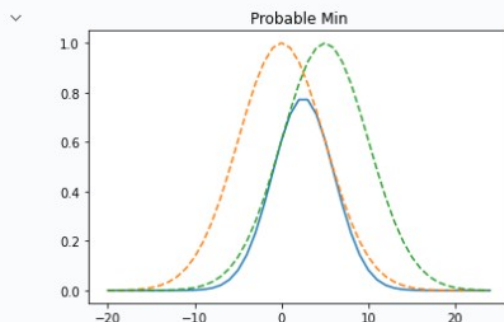


Вірогідна інтерпретація кон'юктивну і диз'юктивних операторів

```

n 72 1 def prob_mix_f(x, f1, f2):
      2     y = f1 * f2
      3     plt.plot(x, y)
      4     plt.plot(x, f1, linestyle='--')
      5     plt.plot(x, f2, linestyle='--')
      6     plt.title(f"Probable Min")
      7     plt.show()
      8
      9
     10 x = np.arange(-20, 25)
     11 prob_mix_f(x, skfuzzy.gaussmf(x, 0, 5), skfuzzy.gaussmf(x, 5, 5))

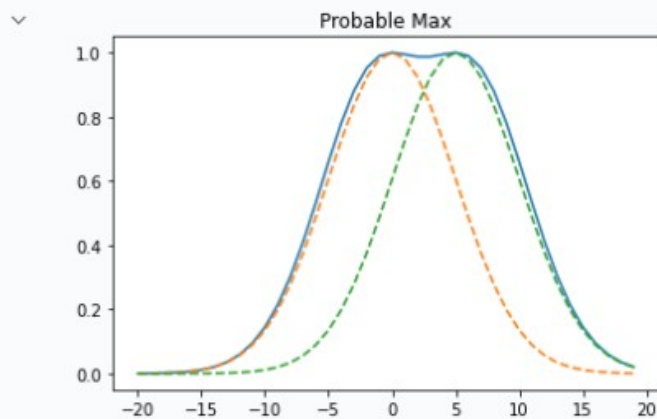
```



```

In 73 1 def prob_max_f(x, f1, f2):
      2     y = f1 + f2 - (f1 * f2)
      3     plt.plot(x, y)
      4     plt.plot(x, f1, linestyle='--')
      5     plt.plot(x, f2, linestyle='--')
      6     plt.title(f"Probable Max")
      7     plt.show()
      8
      9
     10 x = np.arange(-20, 20)
     11 prob_max_f(x, skfuzzy.gaussmf(x, 0, 5), skfuzzy.gaussmf(x, 5, 5))

```



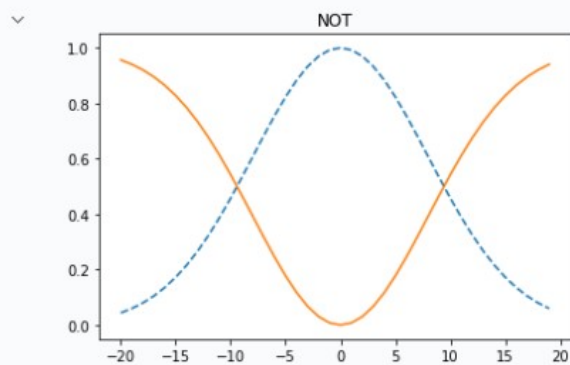
Add Code

Заперечення

```

91 1 def not_f(x, func):
      2     not_func = skfuzzy.fuzzy_not(func)
      3     plt.plot(x, func, linestyle='--')
      4     plt.plot(x, not_func)
      5     plt.title("NOT")
      6     plt.show()
      7
      8
      9 x = np.arange(-20, 20)
     10 not_f(x, skfuzzy.gaussmf(x, 0, 8))

```



Add Code Cell Add Notebook

Висновок: в даній лабораторній роботі ми використали декілька функцій приналежності для побудови нечітких множин. А також виконали найбільш поширені логічні операції над нечіткими множинами.