

Слайд 2:

На цьому слайді ви можете побачити постановку задачі:

Потрібно розробити алгоритм нечіткого виводу для оцінки атестації студента.

Завдання:

1. Задати функції належності лінгвістичних змінних
2. Задати 4 правила
3. Нечітку імплікацію обчислити за правилом Мамдані
4. Для трьох відомих значень знайти четверте - невідоме

Слайд 3:

Нами була використана бібліотека `skFuzzy` для реалізації обчислень. `Scikit-Fuzzy` - це набір алгоритмів нечіткої логіки, призначених для використання в стеку `SciPy`, написаний мовою програмування `Python`.

Слайд 4-7:

Далі давайте перейдемо до опису самого алгоритму. Для початку нам треба задати лінгвістичні змінні та функції належності лінгвістичних змінних.

Вхідними лінгвістичними змінними будуть:

- Відвідуваність
- Активність на заняттях
- Домашні завдання

Вихідна змінна — це оцінка

Відвідуваність та активність може бути:

- Низькою
- Середньою
- Високою

Домашня робота може бути виконана:

- Погано
- Добре
- Відмінно

Оцінка може бути:

- Незадовільною
- Задовільною
- Доброю
- Відмінною

Всі функції належності ви можете бачити на слайдах

Слайд 8:

Далі давайте візуалізуємо результати, щоб побачити наші функції належності графічно. Будуються ці функції належності використовуючи функції `trapmf` та `trimf` з бібліотеки `skFuzzy`. Для візуалізації використали `matplotlib`

Слайд 9:

На цьому слайді ви можете побачити вхідні дані для описаного в цій презентації прикладу. Усі оцінки в нас визначаються в межах від 0 до 100.

- `attendance_score` — це оцінка відвідування
- `participation_score` — це оцінка активності на заняттях
- `homework_score` — це оцінка за виконані домашні завдання

Слайд 10:

Далі, використовуючи функцію `interp_membership` з бібліотеки `skFuzzy` ми вираховуємо ступень належності для кожного заданого параметра, тобто для відвідуваності, активності та домашніх завдань, та для кожної функції належності. Код, який це робить ви можете побачити на слайді. Результати наведено на цьому ж слайді. Тут показана ступінь належності `attendance_score` до множини високих, низьких чи середніх оцінок.

Слайд 11:

Далі іде визначення чотирьох правил, які ви бачите на слайді:

1. Якщо Відвідуваність Висока і Активність на заняттях Висока і Домашні завдання Відмінно, тоді Оцінка Відмінно.
2. Якщо Відвідуваність Середня і Активність на заняттях Середня і Домашні завдання Добре, тоді Оцінка Добре.
3. Якщо Відвідуваність Низька і Активність на заняттях Низька і Домашні завдання Погано, тоді Оцінка Незадовільно.
4. Якщо Відвідуваність Середня і Активність на заняттях Низька і Домашні завдання Добре, тоді Оцінка Задовільно.

Слайд 12:

Далі займаємось обчисленням нечіткої імплікації. Нами був обраний метод Мамдані, а не метод Ларсена чи Цукамото. Чому? Тому що він був найбільш простим в реалізації програмно + у нас ще була схема, яку ви нам кидали, тому було легше розібратись. Ця схему можете бачити на слайді.

Слайд 13:

Тут ви можете бачити код, він розбитий на ті самі етапи, що показані на схемі, тобто, спочатку `max`, далі `min`, потім знову `max` для агрегації та метод дефазифікація центроїдою, як на схемі.

В принципі, на цьому етапі ми і отримуємо бажане значення оцінки, яке буде показане далі

Слайд 14:

На цьому слайді ви можете бачити результат роботи програми, тобто оцінку, яка становить 81.16.

Слайд 15:

На цьому слайді у нас зображена функція належності вихідної змінної, вхідні параметри (позначені зірочкою, тобто:

- `attendance_score = 75`
- `participation_score = 65`
- `homework_score = 85`

та вихідний параметр (позначений кружечком) — це `grade_centroid`, що рівна 81.16 в нашому прикладі. Пунктирна лінія тут для кращої візуалізації.