Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Кафедра автоматизації проектування

 енергетичних процесів і систем

**Звіт**

**з циклу лабораторних робіт**

**з дисципліни**

**„Нечітке моделювання та управління”**

Лабораторна  робота № 2

**Розробка нечіткої моделі**

Виконала: **Юрченко Богдана**

Студентка групи:   **ТВ-01 мп**

Перевірив викладач: **проф. Шушура О. М.**

Київ - 2021

1. Завдання відповідно до варіанту.

Для визначеного відповідно до варіанту процесу побудувати нечітку модель.

Необхідно

1. розробити структуру системи управління,
2. визначити вхідні і вихідні змінні,
3. сформувати базу нечітких правил,
4. розробити алгоритм фазифікації,
5. процедуру активізації та агрегування,
6. реалізувати алгоритм дефазифікації.

Концептуальний опис виконувати на основі представленої в лекції методики. На основі використання стандартних бібліотек та підсистем (наприклад, MATLAB), розробити програмне забезпечення для реалізації створених алгоритмів моделювання або управління обраним об'єктом.

1. Номер старої залікової книжки - tv-6149, варіант 10 - класифікація смартфонів.

Виконання роботи.

# Визначити вхідні і вихідні змінні

Вхідними даними системи є числові характеристики смартфона

1. ціна
2. діагональ екрану
3. кількість мегапікселей в камері
4. рік випуску
5. кількість GB оперативної памяті
6. кількість ядер процесора
7. ціна - невелика, коли менше 200 долларів, середня, коли до 600 долларів , велика коли більше 600 долларів
8. діагональ екрану маленька, коли менше 5 дюймів , середня коли до 6 дюймів, велика, коли більше 6 дюймів
9. кількість пікселей в камері - мало, коли менше 15, нормально коли до 30, багато коли до 60, більше - дуже багато
10. рік випуску - дуже добре, коли 2021, нормально коли 2020, не добре коли менше 2020, дуже погано, коли менше 2019
11. кількість оперативної памяті в GB - дуже погано, коли менше 2 гб, погано, коли до 4, нормально коли до 10, добре коли до 16, дуже добре коли більше
12. кількість ядер процесора - дуже погано, коли менше 2, поагно коли до 4, нормально коли до 8, добре коли більше

# Вихідним значенням системи і клас телефону

1. дешевий телефон - якщо характеристики не дуже хороші і ціна маленька
2. хороший телефон за свою ціну - якщо характеристики хороші, а ціна невелика
3. мід-рендж - коли характеристики хороші і ціна нормальна
4. для фотографій - коли хороша камера і хороший екран і хороший процесор
5. для геймерів - коли хороший екран, процесор і багато памяті
6. флагман - коли характеристики чудові і ціна висока
7. поганий телефон - коли ціна велика, а характеристики погані

## Розробити алгоритм фазифікації

Для виконання роботи була використана бібліотека Python skfuzzy .

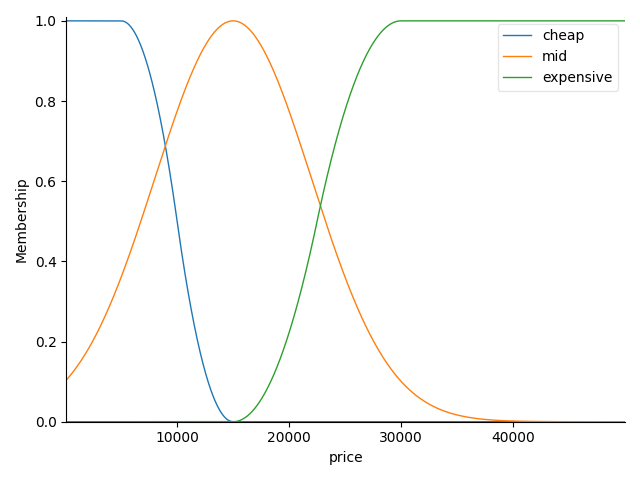
Бібліотека створена для роботи із нечиткими множинами, числами, правилами та системами.

Документація доступна за посиланням

<https://pythonhosted.org/scikit-fuzzy/>

В бібліотеці також були використані функції для генерації функцій належності.

Визначені функції належності для кожної характеристики:

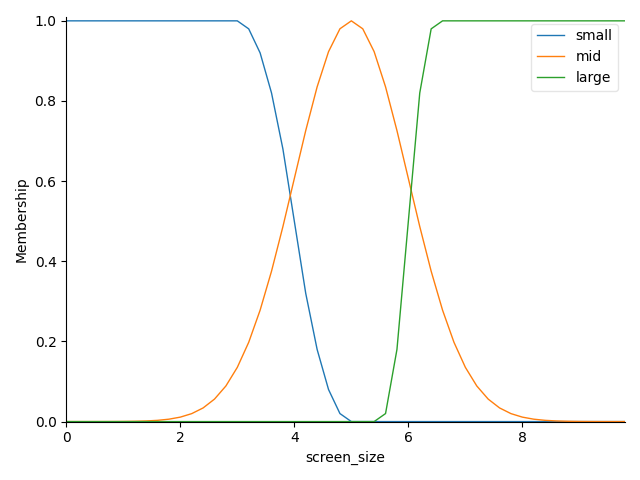


price = ctrl.Antecedent(np.arange(100., 50000., 1), 'price')

price['cheap'] = fuzz.zmf(price.universe, 5000, 15000)

price['mid'] = fuzz.gaussmf(price.universe, 15000, 7000)

price['expensive'] = fuzz.smf(price.universe, 15000, 30000)

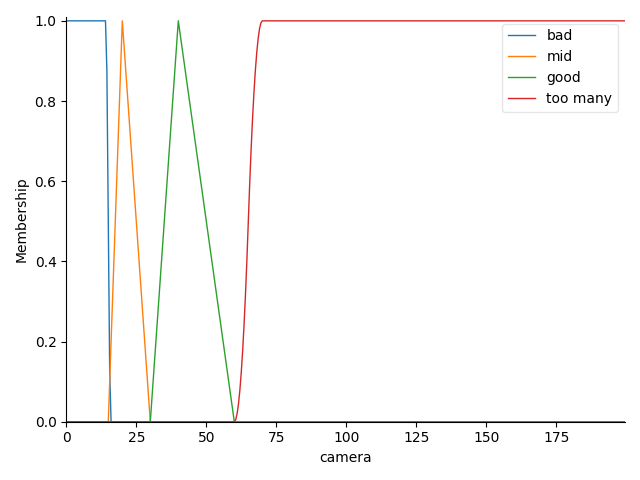


screen\_size = ctrl.Antecedent(np.arange(0., 10., 0.2), 'screen\_size')

screen\_size['small'] = fuzz.zmf(screen\_size.universe, 3, 5)

screen\_size['mid'] = fuzz.gaussmf(screen\_size.universe, 5, 1)

screen\_size['large'] = fuzz.smf(screen\_size.universe, 5.5, 6.5)



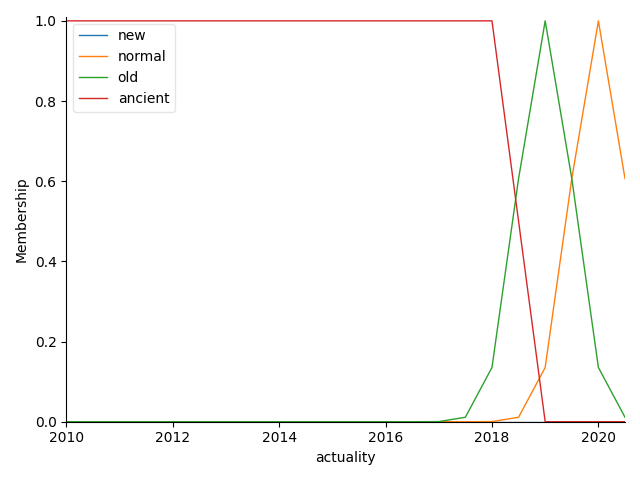
camera = ctrl.Antecedent(np.arange(0., 200., 0.5), 'camera')

camera['bad'] = fuzz.zmf(camera.universe, 14, 16)

camera['mid'] = fuzz.trimf(camera.universe, [15, 20, 30])

camera['good'] = fuzz.trimf(camera.universe, [30, 40, 60])

camera['too many'] = fuzz.smf(camera.universe, 60, 70)



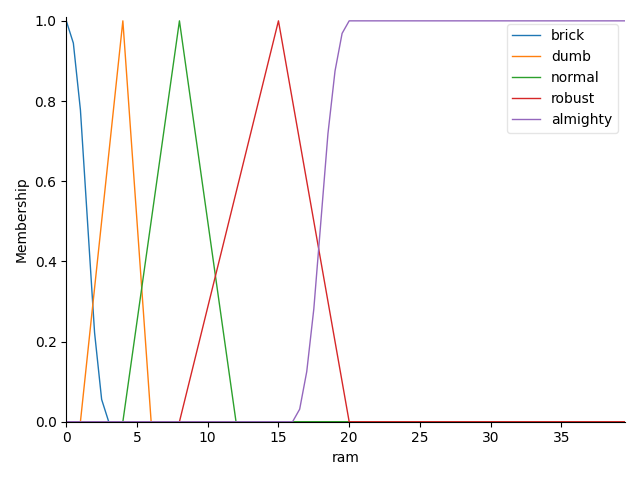
actuality = ctrl.Antecedent(np.arange(2010., 2021., 0.5), 'actuality')

actuality['new'] = fuzz.smf(actuality.universe, 2020.5, 2021)

actuality['normal'] = fuzz.gaussmf(actuality.universe, 2020, 0.5)

actuality['old'] = fuzz.gaussmf(actuality.universe, 2019, 0.5)

actuality['ancient'] = fuzz.zmf(actuality.universe, 2018, 2019)



ram = ctrl.Antecedent(np.arange(0., 40., 0.5), 'ram')

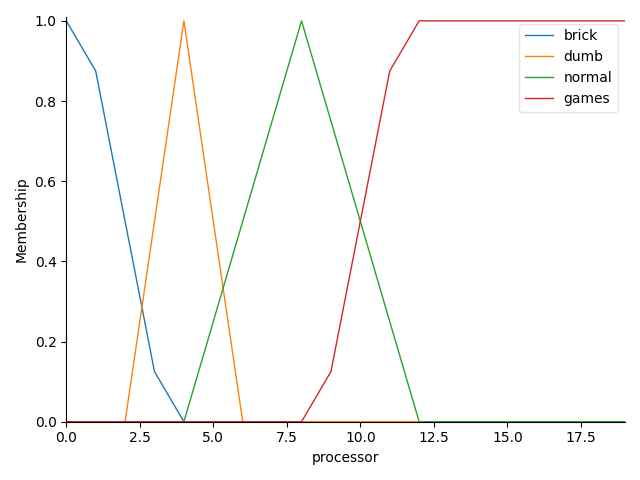
ram['brick'] = fuzz.zmf(ram.universe, 0, 3)

ram['dumb'] = fuzz.trimf(ram.universe, [1, 4, 6])

ram['normal'] = fuzz.trimf(ram.universe, [4, 8, 12])

ram['robust'] = fuzz.trimf(ram.universe, [8, 15, 20])

ram['almighty'] = fuzz.smf(ram.universe, 16, 20)



cores = ctrl.Antecedent(np.arange(0., 20., 1), 'processor')

cores['brick'] = fuzz.zmf(cores.universe, 0, 4)

cores['dumb'] = fuzz.trimf(cores.universe, [2, 4, 6])

cores['normal'] = fuzz.trimf(cores.universe, [4, 8, 12])

cores['games'] = fuzz.smf(cores.universe, 8, 12)

# Визначення правил

cheap\_phone = ctrl.Rule(price['cheap'] & camera['bad'] & actuality["normal"] & ram["dumb"] & cores["dumb"],

recommendation["cheap\_phone"])

good\_deal = ctrl.Rule(price['cheap'] & camera['good'] & actuality["normal"] & ram["normal"] & cores["normal"],

recommendation["good\_deal"])

mid\_phone = ctrl.Rule(price['cheap'] & camera['good'] & actuality["normal"] & ram["normal"] & cores["normal"],

recommendation["mid\_phone"])

for\_photos = ctrl.Rule(camera["too many"] & ram["robust"] & cores["normal"], recommendation["for\_photos"])

for\_gamers = ctrl.Rule(screen\_size["large"] & actuality["normal"] & ram["almighty"] & cores["games"],

recommendation["for\_gamers"])

flagman = ctrl.Rule(price['expensive'] & camera['good'] & actuality["new"] & ram["almighty"] & cores["games"],

recommendation["flagman"])

awful = ctrl.Rule(price['expensive'] & (camera['mid'] | camera['bad']) &

ram["dumb"] & cores["dumb"], recommendation["awful"])

# Було протестовано на 3 прикладах :

