

به نام خدا



گزارش تحلیلی پروژه دوم منطق فازی

موضوع :

پیاده سازی کنترل کننده منطق فازی برای سیستم آبیاری خودکار گیاه

درس: هوش محاسباتی

استاد: دکتر حسین کارشناس

دستیاران آموزش :

رضا برزگر - علی شاه زمانی - آرمان خلیلی

اعضای گروه :

یونس ایوبی راد

پویا اسفندانی

# Antecedents & Consequents

```
# Antecedents (Inputs)
fitness_level = ctrl.Antecedent(np.arange(0, 101, 1), 'fitness_level')
energy_level = ctrl.Antecedent(np.arange(0, 101, 1), 'energy_level')
goal = ctrl.Antecedent(np.arange(0, 101, 1), 'goal')
age = ctrl.Antecedent(np.arange(0, 101, 1), 'age')
weight_status = ctrl.Antecedent(np.arange(0, 101, 1), 'weight_status')

# Consequents (Consequent)
intensity = ctrl.Consequent(np.arange(0, 101, 1), 'intensity')
duration = ctrl.Consequent(np.arange(0, 121, 1), 'duration')
```

## Membership functions

```
# Membership functions for Fitness Level
fitness_level['beginner'] = fuzz.trimf(fitness_level.universe, [0, 0, 50])
fitness_level['intermediate'] = fuzz.trimf(fitness_level.universe, [20, 50, 80])
fitness_level['advanced'] = fuzz.trimf(fitness_level.universe, [50, 100, 100])

# Membership functions for Energy Level
energy_level['low'] = fuzz.trimf(energy_level.universe, [0, 0, 50])
energy_level['medium'] = fuzz.trimf(energy_level.universe, [20, 50, 80])
energy_level['high'] = fuzz.trimf(energy_level.universe, [50, 100, 100])

# Membership functions for Goal
goal['weight_loss'] = fuzz.trimf(goal.universe, [0, 0, 40])
goal['general'] = fuzz.trimf(goal.universe, [25, 50, 75])
goal['muscle_gain'] = fuzz.trimf(goal.universe, [60, 100, 100])

# Membership functions for Age
age['young'] = fuzz.trapmf(age.universe, [0, 0, 18, 35])
age['middle_aged'] = fuzz.trimf(age.universe, [30, 50, 65])
age['old'] = fuzz.trapmf(age.universe, [60, 80, 100, 100])

# Membership functions for Weight Status
weight_status['underweight'] = fuzz.trimf(weight_status.universe, [0, 0, 30])
weight_status['normal'] = fuzz.trimf(weight_status.universe, [20, 50, 80])
weight_status['overweight'] = fuzz.trimf(weight_status.universe, [70, 100, 100])

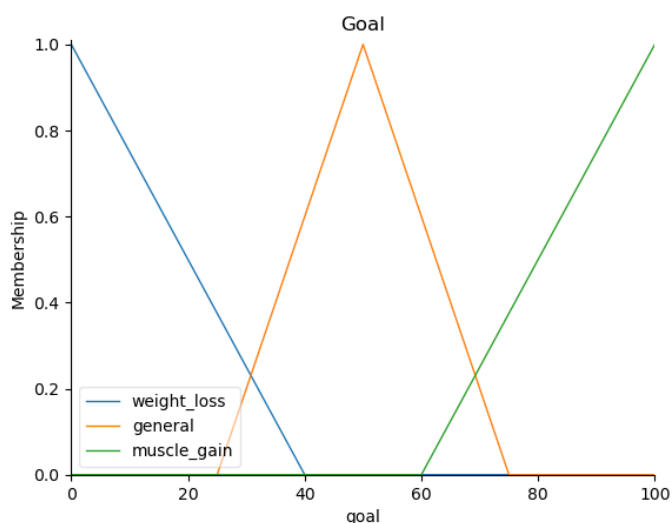
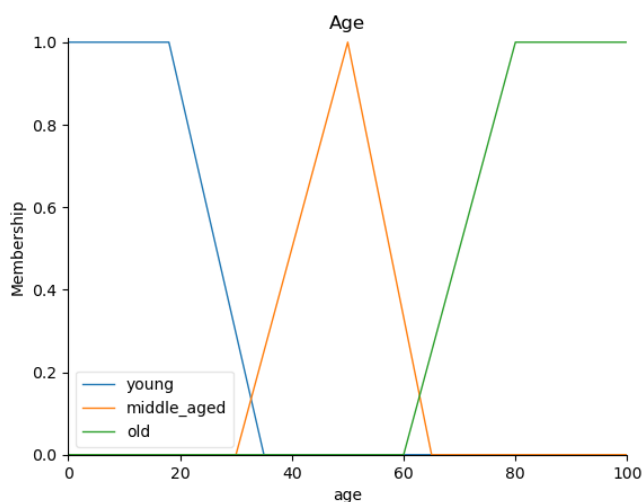
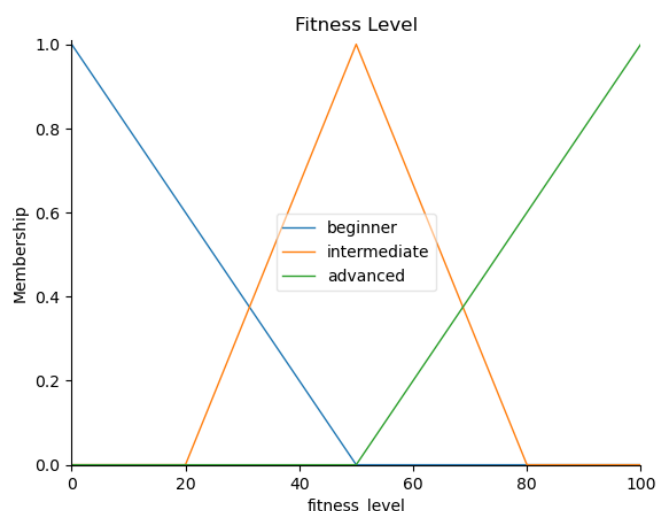
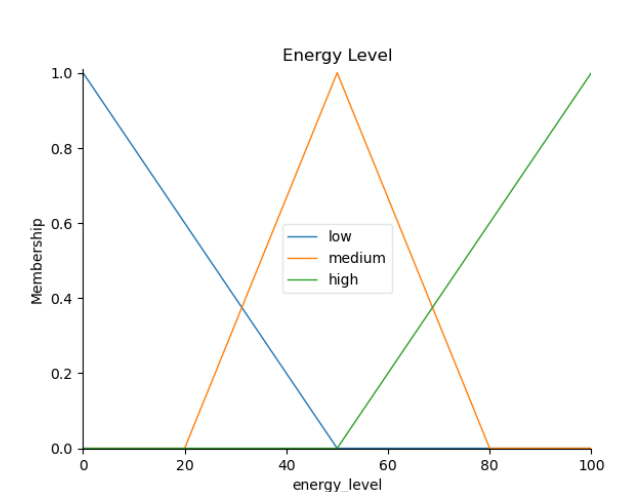
# Membership functions for Intensity
intensity['low'] = fuzz.trimf(intensity.universe, [0, 0, 40])
intensity['medium'] = fuzz.trimf(intensity.universe, [30, 50, 70])
intensity['high'] = fuzz.trimf(intensity.universe, [60, 100, 100])

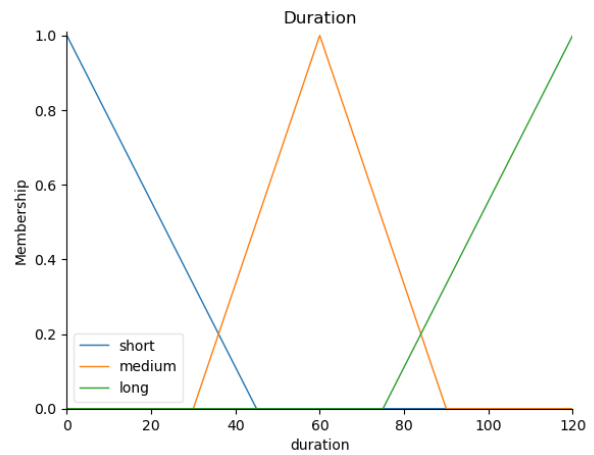
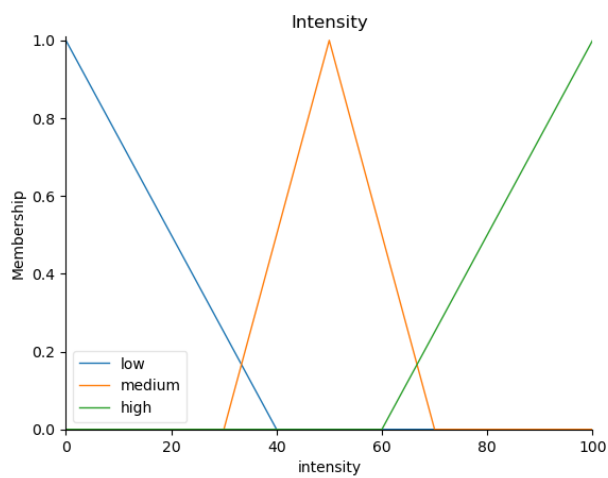
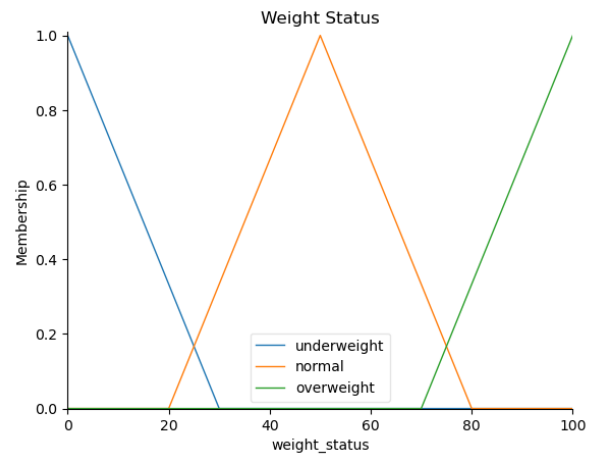
# Membership functions for Duration
duration['short'] = fuzz.trimf(duration.universe, [0, 0, 45])
duration['medium'] = fuzz.trimf(duration.universe, [30, 60, 90])
duration['long'] = fuzz.trimf(duration.universe, [75, 120, 120])
```

در این پروژه، یک سیستم استنتاج فازی برای پیشنهاد شدت و مدت زمان تمرین ورزشی طراحی شده است. ورودی‌های (Antecedents) این سیستم شامل سطح آمادگی جسمانی (fitness\_level)، سطح انرژی (energy\_level)، هدف ورزشی (goal)، سن (age) و وضعیت وزن (weight\_status) می‌باشند که هر یک در بازه‌ای مشخص تعریف شده‌اند. خروجی‌های (Consequents) سیستم نیز شامل شدت تمرین (intensity) و مدت زمان تمرین (duration) هستند که بازه مقادیر پیشنهادی برای آن‌ها نیز تعیین شده است.

## ۲. توابع عضویت فازی

برای هر یک از ورودی‌ها و خروجی‌های تعریف شده، توابع عضویت فازی (Membership Functions) مشخص شده‌اند. این توابع به سیستم اجازه می‌دهند تا مقادیر ورودی و خروجی را به درجات عضویت در مجموعه‌های فازی مانند مبتدی، متوسط و پیشرفته برای سطح آمادگی جسمانی یا کم، متوسط و زیاد برای شدت تمرین تبدیل کند. در این پیاده‌سازی از توابع عضویت مثلثی (trimf) و ذوزنقه‌ای (trapmf) استفاده شده است.





## Fuzzy Rules

- The rules are defined in general terms to be able to handle different situations.
- The rules are structured based on common sense scenarios and medical and sports physiology theories.

Rules based primarily on Fitness and Energy

Rules considering Goal

Rules considering Age

## Rules considering Weight Status

### Rules for specific edge cases or combinations

#### ۳. تعریف قوانین فازی

بخش حیاتی و تصمیم‌ساز سیستم فازی، مجموعه قوانین "اگر-آنگاه" (If-Then) است که چگونگی استنتاج خروجی‌ها از ورودی‌ها را مشخص می‌کند. این قوانین منطق فازی پروژه را پیاده‌سازی کرده و ارتباط بین حالت‌های مختلف ورودی‌ها (که با توابع عضویت فازی بیان شدند) و مقادیر پیشنهادی برای خروجی‌ها را تعریف می‌کنند.

هر قانون از یک بخش "اگر" (Antecedent) و یک بخش "آنگاه" (Consequent) تشکیل شده است. بخش "اگر" ترکیبی از شرایط فازی بر روی یک یا چند ورودی (مانند "سطح آمادگی مبتدی" و "سطح انرژی پایین") است و بخش "آنگاه" مقادیر فازی متناظر برای خروجی‌ها (مانند "شدت پایین" و "مدت زمان کوتاه") را مشخص می‌کند.

مجموعه قوانین طراحی شده در این پروژه، طیف وسیعی از سناریوها را پوشش می‌دهند. این قوانین بر اساس ترکیبات مختلفی از ورودی‌ها، از جمله سطح آمادگی و انرژی، هدف ورزشی، سن و وضعیت وزن، شدت و مدت زمان تمرین را پیشنهاد می‌دهند. برای مثال:

- قوانینی برای افراد با سطوح مختلف آمادگی و انرژی (مثلاً فرد مبتدی با انرژی بالا یا فرد پیشرفته با انرژی پایین).
  - قوانینی که هدف ورزشی را در نظر می‌گیرند (کاهش وزن، افزایش عضله یا آمادگی عمومی) و آن را با سطح آمادگی و انرژی ترکیب می‌کنند.
  - قوانینی که تأثیر سن (به‌ویژه افراد مسن‌تر) و وضعیت وزن (کم‌وزن، نرمال، اضافه‌وزن) را در توصیه‌های تمرینی لحاظ می‌کنند.
  - و قوانینی خاص برای پوشش دادن شرایط یا ترکیبات کمتر رایج ورودی‌ها.
- این مجموعه قوانین، دانش اصلی سیستم را تشکیل داده و امکان استنتاج پیشنهادات تمرینی منعطف و منطبق با شرایط فردی را فراهم می‌آورند.

# Control System & Defuzzification

```
# Combine rules into a Control System
exercise_ctrl = ctrl.ControlSystem([
    rule1, rule2, rule3, rule4, rule5, rule6, rule7, rule8, rule9, rule10,
    rule11, rule12, rule13, rule14, rule15, rule16, rule17, rule18, rule19, rule20,
    rule21, rule22, rule23, rule24, rule25, rule26, rule27, rule28, rule29, rule30,
    rule31, rule32, rule33, rule34, rule35, rule36, rule37, rule38, rule39, rule40,
    rule41, rule42
])

# Create a Control System Simulation
exercise_simulation = ctrl.ControlSystemSimulation(exercise_ctrl)
```

## Fuzzy Inference

- create a Simulation
- define TestCases
- Test & Validation

## Simulation

```
# Create a Control System Simulation
exercise_simulation = ctrl.ControlSystemSimulation(exercise_ctrl)
```

### ۴. سیستم کنترل فازی و استنتاج (Inference)

در این مرحله، تمامی قوانین فازی تعریف شده ترکیب شده و یک "سیستم کنترل فازی" کامل را تشکیل می‌دهند. سپس، یک شبیه‌ساز (Simulation) از این سیستم کنترل ایجاد می‌شود. این شبیه‌ساز مسئول اجرای فرآیند استنتاج فازی است که شامل مراحل زیر می‌شود:

- فازی‌سازی (Fuzzification): تبدیل مقادیر عددی و دقیق ورودی‌ها (مانند سن ۲۲ یا سطح آمادگی ۹۰) به درجات عضویت در مجموعه‌های فازی مربوطه با استفاده از توابع عضویت.
- ارزیابی قوانین (Rule Evaluation): تعیین میزان صدق کردن بخش "اگر" هر قانون بر اساس درجات عضویت فازی شده ورودی‌ها.
- تجميع (Aggregation): ترکیب نتایج قوانین مختلف برای هر یک از خروجی‌ها (شدت و مدت زمان).
- غیرفازی‌سازی (Defuzzification): تبدیل نتایج فازی تجميع شده برای هر خروجی، به یک مقدار عددی و دقیق نهایی (مثلاً شدت ۷۵٪ یا مدت زمان ۶۰ دقیقه). در این پروژه، از روش Centroid برای غیرفازی‌سازی استفاده شده است.

## Test & Evaluation & Visualize

```
print("Multiple Test Cases for Validation")

# Loop through each test case
for i, case in enumerate(test_cases):
    print(f"\n===== Test Case {i + 1}: {case['name']} =====")
    print("Inputs:")
    inputs = case['inputs']
    for key, value in inputs.items():
        print(f"  {key}: {value}")

    # Feed inputs to the simulation
    try:
        for key, value in inputs.items():
            exercise_simulation.input[key] = value

        # Compute the results
        exercise_simulation.compute()

        # Get output values
        recommended_intensity = exercise_simulation.output['intensity']
        recommended_duration = exercise_simulation.output['duration']

        print("\nOutputs:")
        print(f"  Recommended Intensity: {recommended_intensity:.2f} / 100%")
        print(f"  Recommended Duration: {recommended_duration:.2f} / 120 minutes")

        # -- Visualize --

        # following code will show two plots for each test case.
        intensity.view(sim=exercise_simulation)
        plt.title(f"Test Case {i + 1} - Intensity")
        plt.show()

        duration.view(sim=exercise_simulation)
        plt.title(f"Test Case {i + 1} - Duration")
        plt.show()

    except Exception as e:
        print(f"Error processing Test Case {i + 1}")
        print(f"Error message: {e}")

print("--- All Test Cases Completed ---")
```

### ۵. آزمون و اعتبارسنجی

برای اطمینان از عملکرد صحیح سیستم، چندین سناریوی ورودی متفاوت به عنوان موارد آزمون (Test Cases) تعریف شده‌اند. هر مورد آزمون شامل مجموعه‌ای از مقادیر مشخص برای تمام ورودی‌ها (سطح آمادگی، انرژی، هدف، سن، وضعیت وزن) است که شرایط مختلف کاربران را شبیه‌سازی می‌کند.

شبیه‌ساز سیستم کنترل فازی، ورودی‌های هر مورد آزمون را دریافت کرده، فرآیند استنتاج فازی را اجرا می‌کند و در نهایت مقادیر عددی پیشنهادی برای شدت و مدت زمان تمرین را محاسبه و ارائه می‌دهد. نتایج حاصل از این شبیه‌سازی‌ها برای هر مورد آزمون نمایش داده شده و در صورت لزوم می‌توانند به صورت بصری نیز مورد بررسی قرار گیرند تا ارزیابی شود که

پیشنهادات سیستم تا چه حد با منطق و دانش موجود مطابقت دارند. این مرحله برای اعتبارسنجی مدل فازی و اطمینان از معقول بودن خروجی‌ها در شرایط مختلف ضروری است.

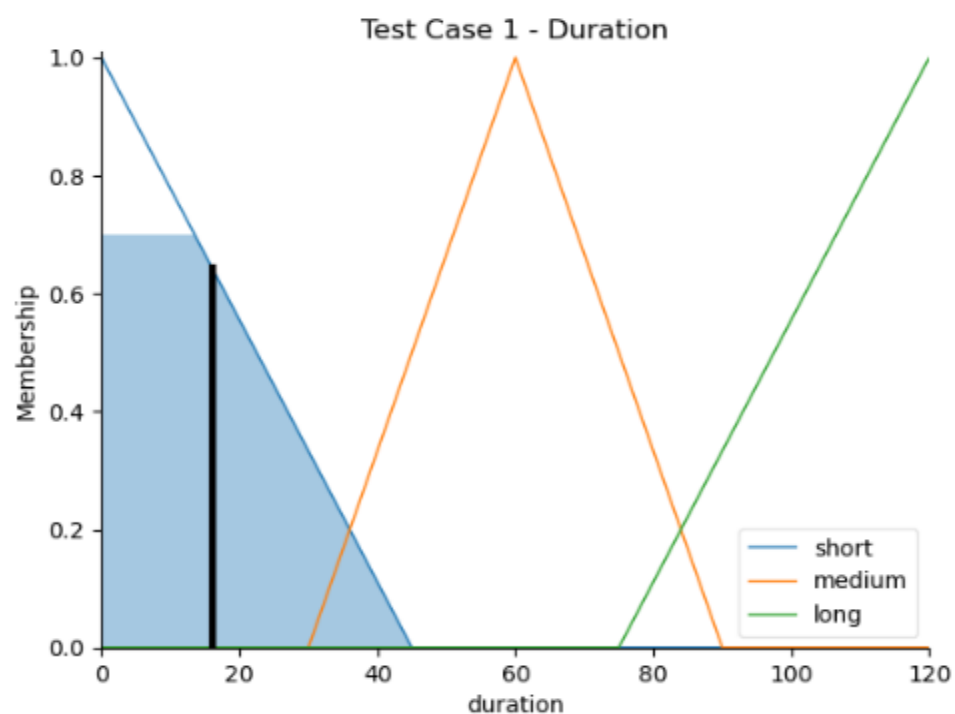
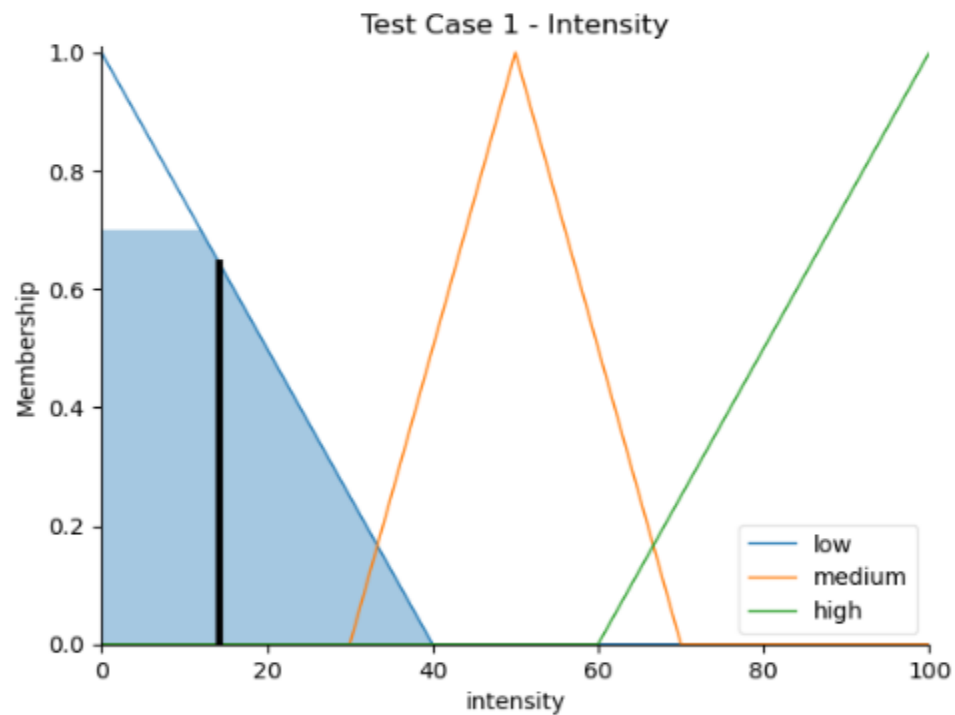
===== Test Case 1: Case 1: Beginner, Low Energy, General, Young, Normal =====

Inputs:

fitness\_level: 15  
energy\_level: 20  
goal: 50  
age: 22  
weight\_status: 50

Outputs:

Recommended Intensity: 14.26 / 100%  
Recommended Duration: 16.04 / 120 minutes





مورد آزمون ۱: مبتدی، انرژی پایین، هدف عمومی، جوان، وزن نرمال

- ورودی‌ها: سطح آمادگی: ۱۵ (مبتدی)، سطح انرژی: ۲۰ (پایین)، هدف: ۵۰ (عمومی)، سن: ۲۲ (جوان)، وضعیت وزن: ۵۰ (نرمال).
- خروجی‌های سیستم: شدت پیشنهادی: ۱۴.۲۶ از ۱۰۰، مدت زمان پیشنهادی: ۱۶.۰۴ از ۱۲۰ دقیقه.
- تحلیل: با توجه به اینکه ورودی‌ها نشان‌دهنده فردی "مبتدی" با "انرژی پایین" هستند، پیشنهاد سیستم برای "شدت پایین" و "مدت زمان کوتاه" کاملاً منطقی و محافظه‌کارانه است. این نتیجه منعکس‌کننده نیاز به شروع آرام برای جلوگیری از فشار بیش از حد یا آسیب دیدگی در یک فرد تازه‌کار و کم‌انرژی است.

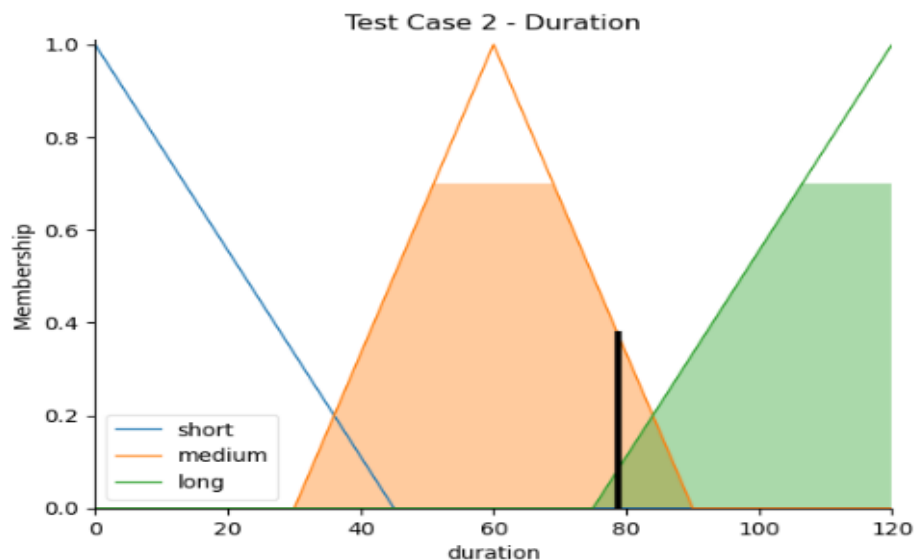
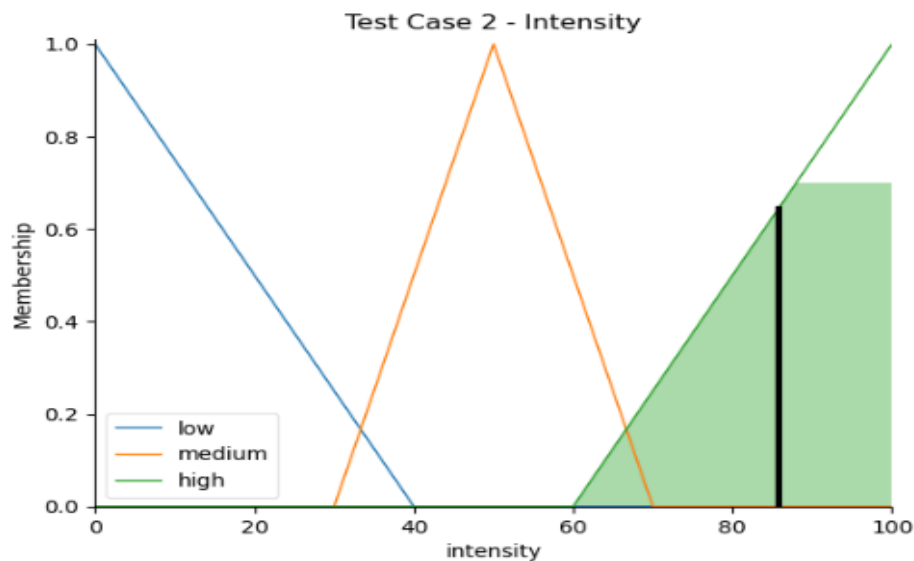
===== Test Case 2: Case 2: Advanced, High Energy, Muscle Gain, Young, Normal =====

Inputs:

fitness\_level: 90  
energy\_level: 85  
goal: 95  
age: 28  
weight\_status: 60

Outputs:

Recommended Intensity: 85.74 / 100%  
Recommended Duration: 78.71 / 120 minutes



مورد آزمون ۲: پیشرفته، انرژی بالا، هدف افزایش عضله، جوان، وزن نرمال

- ورودی‌ها: سطح آمادگی: ۹۰ (پیشرفته)، سطح انرژی: ۸۵ (بالا)، هدف: ۹۵ (افزایش عضله)، سن: ۲۸ (جوان)، وضعیت وزن: ۶۰ (نرمال).
  - خروجی‌های سیستم: شدت پیشنهادی: ۸۵.۷۴ از ۱۰۰، مدت زمان پیشنهادی: ۷۸.۷۱ از ۱۲۰ دقیقه.
  - تحلیل: در مقابل مورد اول، این ورودی‌ها مربوط به فردی "پیشرفته" با "انرژی بالا" و هدف "افزایش عضله" هستند. سیستم با توجه به این شرایط، "شدت بالا" و "مدت زمان نسبتاً طولانی" را پیشنهاد داده است. این خروجی نیز کاملاً با منطق ورزشی برای فردی با این مشخصات که به دنبال نتایج قابل توجه است، همخوانی دارد و نشان‌دهنده توانایی سیستم در ارائه توصیه‌های مناسب برای سطوح بالای آمادگی و انرژی است.
- این موارد آزمون اولیه نشان می‌دهند که سیستم فازی قادر است بر اساس ترکیب پیچیده‌ای از ویژگی‌های ورودی، پیشنهادات معقول و متناسب با شرایط کاربر ارائه دهد. تحلیل سایر موارد آزمون نیز برای پوشش دادن طیف وسیع‌تری از سناریوها و اعتبارسنجی بیشتر سیستم ضروری است.

باقی تست‌ها در فایل Jupyter