Fuzzy Sets: Employee Productivity Evaluation

Module Code	**:**	Fuzzy Logic (112023-VAZ)
Intake Code	**:**	APDMF2310AI
Lecturer Name	**:**	Dr. Vazeerudeen Hameed
Hand in Date	**:**	4th of March, 2024
Students Name	**:**	Nurlan Amanov (TP077526), Gorken Mezli (TP074468)

Table of Contents

ABSTRACT

Fuzzy Logic and Fuzzy Sets introduced by Lotfi Zadeh in 1965. This paper explores the origin of Fuzzy Logic and Zadeh's motivation to produce Fuzzy Logic. The focus is to develop Fuzzy Logic system for employee productivity using created scenario. The scenario is based on task competition time, work environment satisfaction and caffeine intake. The paper describes the motivation behind choosing those three factors. There are seven rules to create Fuzzy Logic system and identify employee productivity level.

1. INTRODUCTION

Fuzzy Logic and Fuzzy Sets are generalizations of classical set theory and classical formal logic. For the first time, the concepts were proposed by American scientist Lotfi Zadeh in 1965. The main reason for the appearance of Fuzzy Logic was the presence of approximate reasoning in the description of various processes, systems, or objects. It took not a little time before Fuzzy Logic was recognized worldwide. How did Zadeh come to create Fuzzy Logic? He believed that humans do not need input like machines - a simple "yes" or "no" or "1" and "0". So, to think "our way", an algorithm must act like a human. Such a thing can be achieved by rules. Example, with an air conditioner

1. If the room is cold and you put the air conditioner on the heating, the room will warm up.

2. If the room is hot and the air conditioner is set to cool, the room will decrease in temperature.

3. If the room is warm and the air conditioner is set to keep warm, the room will not change temperature.

Examples of using Fuzzy Logic in programming are spell checkers, search engines, and precomputation. The use of Fuzzy Sets will be shown in this paper.

2. PROBLEM STATEMENT

While on the job, it is important for every boss to make sure that the environment in which his or her workers are located is the most productive. There are a vast number of factors that can affect this environment, so in an article from Charffold (n.d.) spoke about eight reasons that can affect the deterioration of productivity, where she said:

1. Lack of big-picture view
2. Poor supervision
3. Poor communication
4. Lack of delegation
5. Inconsistency
6. Weak company culture
7. Inadequate technologies
8. Lack of acknowledgment

By analyzing the reasons presented, it can be said that the environment in which workers are located is important for productivity. A friendly and positive environment can have a positive impact on all active work.

The time spent on this very work can affect the productivity of the work. Not everyone wants to overwork, but not working at all is not work either. It is important to find the right balance and complying with all health standards, so as not to kill your body at the computer. Work Golden (2012) talks about the relationship between work and time spent on work. Thus, highlighting the types of work schedules and how they affect productivity and describing cases and studies that showed how reduced time spent on work affected positive productivity gains.

However, it is not just internal moral factors that can affect your performance. You are also the cause of your productivity. Your lifestyle, your actions, anything you commit in any way can affect this, but when imagining adult life, we can usually picture an adult with a briefcase in hand, a watch in the other hand and coffee in hand. In fact, according to statistics from the Sleep Foundation, over 94% of Americans consume caffeine, of which 64% consume it daily. McLellan et al (2016) talks about caffeine and how it affects your productivity throughout the day.

So, considering all the factors we noted above, we produced our scenario to show Fuzzy

Sets on it. It looks as follows:

Three factors have been identified for employee productivity. First is Task Completion Time on a scale from 0 to 60 minutes (short: 0-15 minutes, moderate: 10-40 minutes, long: 30-60 minutes), Work Environment Satisfaction on a scale from 0 to 100 (unpleasant: 0-30, neutral 20-70, pleasant: 60-100) and caffeine intake on a scale from 0 to 5 cups (low: 0-1, moderate: 0.5-3, high: 2-5).

3. SYSTEM DEVELOPMENT

3.1. Design

For this scenario, seven rules have been created that will make up the system:

1. IF Task Completion Time is Long AND Work Environment Satisfaction is Unpleasant THEN Productivity Level is Low.
2. IF Caffeine Intake is High THEN Productivity Level is Moderate.
3. IF Task Completion Time is Short AND Work Environment Satisfaction is Pleasant THEN Productivity Level is High.
4. IF Work Environment Satisfaction is Neutral AND Caffeine Intake is Moderate THEN Productivity Level is Moderate.
5. IF Caffeine Intake is Low THEN Productivity Level is Low.
6. IF Task Completion Time is Moderate AND Work Environment Satisfaction is Pleasant THEN Productivity Level is High.
7. IF Task Completion Time is Short THEN Productivity Level is High.

For this paper sigmoid and gaussian memberships will be used. Sigmoid is used to represent gradual and sigmoidal changes, gaussian has a symmetrical shape that controls the spread of the curve. Together, both memberships are excellent for this work. It is also worth mentioning that the Python libraries 'numpy' and 'matplotlib' will be used. Figure 1 shows the fuzzification.

```
# Fuzzification
tct = np.linspace(0, 60, 200)
short = np.array([mem.sigmoid(x, -0.5, 15) for x in tct])
moderate = np.array([mem.gauss(x, 25, 4) for x in tct])
long = np.array([mem.sigmoid(x, 0.6, 30) for x in tct])

wes = np.linspace(0, 100, 200)
unpleasant = np.array([mem.sigmoid(x, -0.5, 30) for x in wes])
neutral = np.array([mem.gauss(x, 45, 4) for x in wes])
pleasant = np.array([mem.sigmoid(x, 0.6, 60) for x in wes])

coffee = np.linspace(0, 5, 200)
low = np.array([mem.sigmoid(x, -5, 1) for x in coffee])
middle = np.array([mem.gauss(x, 1.75, 0.4) for x in coffee])
high = np.array([mem.sigmoid(x, 6, 2) for x in coffee])

productivity = np.linspace(0, 100, 200)
low_pr = np.array([mem.sigmoid(x, -0.5, 30) for x in productivity])
middle_pr = np.array([mem.gauss(x, 45, 4) for x in productivity])
high_pr = np.array([mem.sigmoid(x, 0.6, 60) for x in productivity])
```

Figure 1. Fuzzification

Task Completion Time (TCT):

Sigmoid (short, long): smooth transition from low to high.

Gaussian (moderate): more centralized representation.

Work Environment Satisfaction (WES):

Sigmoid (unpleasant, pleasant): depicts a negative/positive perception of the surroundings with a seamless transition.

Gaussian (neutral): neutral satisfaction level.

Caffeine Intake (CF):

Sigmoid (low, high): low/high intake with smooth transition.

Gaussian (middle): represents moderate.

Figure 2 shows the input model created in Python.

```
# Modeling Input
inp_tct_sh = mem.sigmoid(inp_tct, -0.5, 15)
inp_tct_m = mem.gauss(inp_tct, 25, 4)
inp_tct_l = mem.sigmoid(inp_tct, 0.6, 30)

inp_wes_unpl = mem.sigmoid(inp_wes, -0.5, 30)
inp_wes_neut = mem.gauss(inp_wes, 45, 4)
inp_wes_pl = mem.sigmoid(inp_wes, 0.6, 60)

inp_cf_l = mem.sigmoid(inp_cf, -5, 1)
inp_cf_m = mem.gauss(inp_cf, 1.75, 0.4)
inp_cf_h = mem.sigmoid(inp_cf, 6, 2)
```

Figure 2. Input Model

The flowchart of the system is presented in Figure 3.

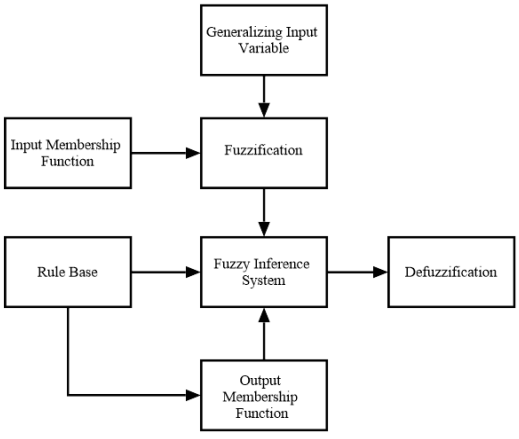

Figure 3. Flowchart

Our main output value is productivity presented in Figure 4.

```
# Aggregate all the rules
aggregated = np.maximum(r1, np.maximum(r2, np.maximum(r3, np.maximum(r4, np.maximum(r5, np.maximum(r6, r7))))))

# Defuzzification using the Centroid method
output_productivity = np.trapz(aggregated * productivity, productivity) / np.trapz(aggregated, productivity)

print("Employee Productivity Level =", output_productivity)
```

Figure 4. Productivity Output

3.2. Implementation

After presenting input variables, let us look at implementation of each rule that we designed in Python on Figure 5.

```
# RULE 1
# IF Task Completion Time is Long AND Work Environment Satisfaction is Unpleasant THEN Productivity Level is Low.
ante = np.min([inp_tct_l, inp_wes_unpl])
r1 = np.fmin(ante, low_pr)

# RULE 2
# IF Caffeine Intake is High THEN Productivity Level is Moderate.
ante = np.min([inp_cf_h])
r2 = np.fmin(ante, middle_pr)

# RULE 3
# IF Task Completion Time is Short AND Work Environment Satisfaction is Pleasant THEN Productivity Level is High.
ante = np.min([inp_tct_sh, inp_wes_pl])
r3 = np.fmin(ante, high_pr)

# RULE 4
# IF Work Environment Satisfaction is Neutral AND Caffeine Intake is Moderate THEN Productivity Level is Moderate.
ante = np.min([inp_wes_neut, inp_cf_m])
r4 = np.fmin(ante, middle_pr)

# RULE 5
# IF Caffeine Intake is Low THEN Productivity Level is Low.
ante = np.min([inp_cf_l])
r5 = np.fmin(ante, low_pr)

# RULE 6
# IF Task Completion Time is Moderate AND Work Environment Satisfaction is Pleasant THEN Productivity Level is High.
ante = np.min([inp_tct_m, inp_wes_pl])
r6 = np.fmin(ante, high_pr)

# RULE 7
# IF Task Completion Time is Short THEN Productivity Level is High.
ante = np.min([inp_tct_sh])
r7 = np.fmin(ante, high_pr)
```

Figure 5. Rules Implementation

To provide visual implementation of the code we created four plots for each of the characteristics that have been used in this study on Figure 6.

```
plt.figure(figsize=(12, 8))

# Task Completion Time
plt.subplot(3, 2, 1)
plt.plot(tct, short, label='Short')
plt.plot(tct, moderate, label='Moderate')
plt.plot(tct, long, label='Long')
plt.scatter(inp_tct, 0)
plt.title('Task Completion Time')
plt.legend()

# Work Environment Satisfaction
plt.subplot(3, 2, 2)
plt.plot(wes, unpleasant, label='Unpleasant')
plt.plot(wes, neutral, label='Neutral')
plt.plot(wes, pleasant, label='Pleasant')
plt.scatter(inp_wes, 0)
plt.title('Work Environment Satisfaction')
plt.legend()
```

```
# Caffeine Intake
plt.subplot(3, 2, 3)
plt.plot(coffee, low, label='Low')
plt.plot(coffee, middle, label='Middle')
plt.plot(coffee, high, label='High')
plt.scatter(inp_cf, 0)
plt.title('Caffeine Intake')
plt.legend()

# Productivity
plt.subplot(3, 2, 4)
plt.plot(productivity, low_pr, label='Low')
plt.plot(productivity, middle_pr, label='Middle')
plt.plot(productivity, high_pr, label='High')
plt.scatter(output_productivity, 0)
plt.title('Productivity')
plt.legend()

# Show the plots
plt.tight_layout()
plt.show()
```

Figure 6. Plots Implementation

The look of the plots presented in Figure 7.

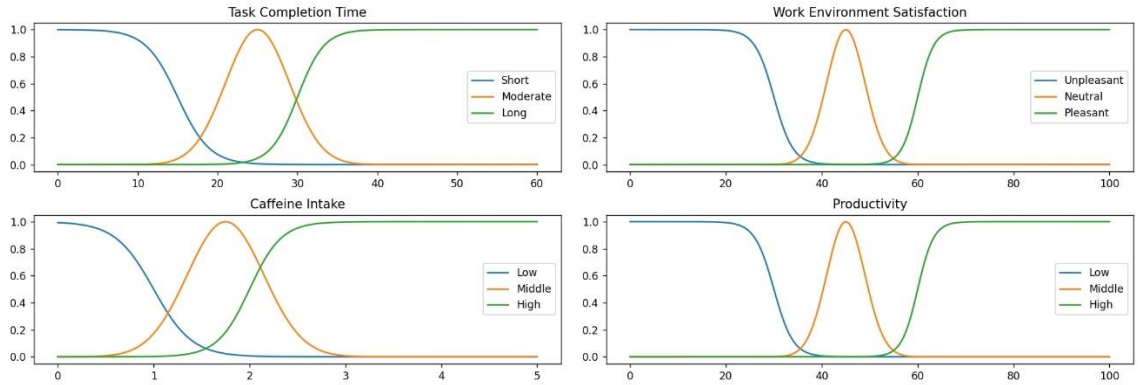

Figure 7. Plots Visualization

3.3. Testing

In this paper, seven tests will be conducted. Each test will aim to validate each of the rules by inputting three values for Task Completion Time (TCT), Work Environment Satisfaction (WES), Caffeine Intake (CI) to calculate Employee Productivity (Figure 8). The values that were set can also be seen in the graphs.

```
inp_tct = -
inp_wes = -
inp_cf =
```

Figure 8. Input Variables

The results of each of the tests and their explanation will be presented below:

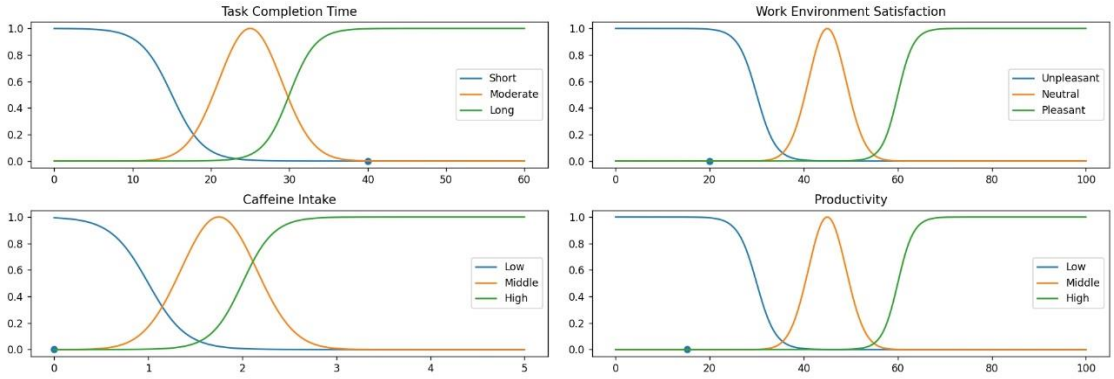

Figure 9. Plot for Rule 1

```
Employee Productivity Level = 15.2436322980167
```

Figure 10. Productivity Level for Rule 1

The first rule says that TCT is long, and WES is unpleasant then productivity is low. Our input values are TCT = 40, WES = 20, CI = 0. Productivity equals 15.24, which is in low level.

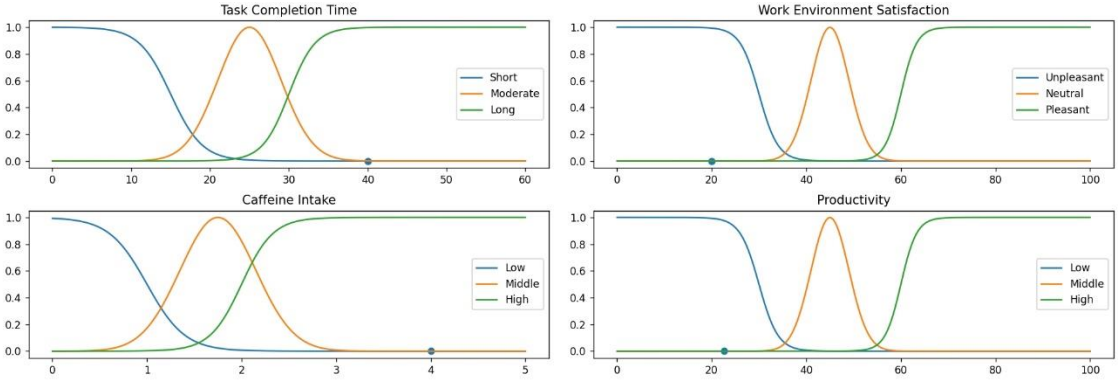

Figure 11. Plot for Rule 2

```
Employee Productivity Level = 22.648450465354166
```

Figure 12. Productivity Level for Rule 2

The second rule says that CI is high then productivity is moderate. The previous input is the same but high CI: TCT = 40, WES = 20, CI = 4. Productivity equals 22.65, which satisfies rule and rule two.

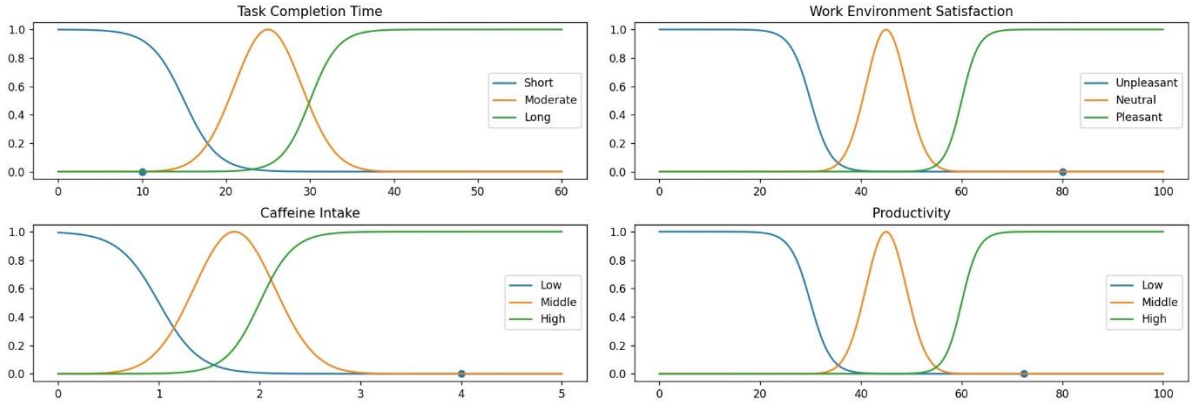

Figure 13. Plot for Rule 3

```
Employee Productivity Level = 72.40180383627447
```

Figure 14. Productivity Level for Rule 3

Rule three says that if TCT is short and WES is pleasant then productivity level is high. Our inputs are TCT = 10, WES = 80 with the same CI = 4. Productivity level equals 72.40, which satisfies rule three.

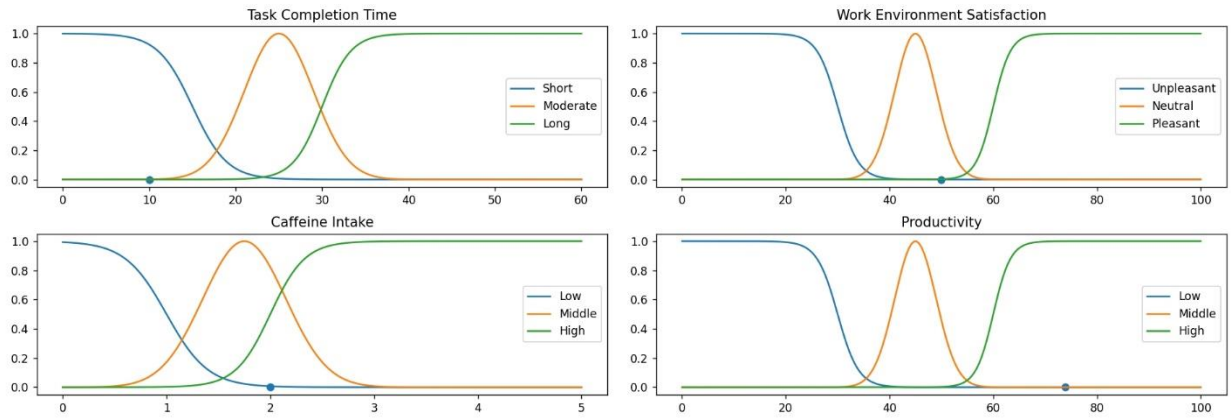

Figure 15. Plot for Rule 4

```
Employee Productivity Level = 73.92947062879358
```

Figure 16. Productivity Level for Rule 4

Rule four states that if WES is neutral and CI is moderate then productivity is moderate. Our inputs are TCT = 10, WES = 50, CI = 2. The productivity level is 73.93, which is not moderate, but due to other rules like rule seven, our productivity is high. The moderate's range is 20-70, so the input makes sense.

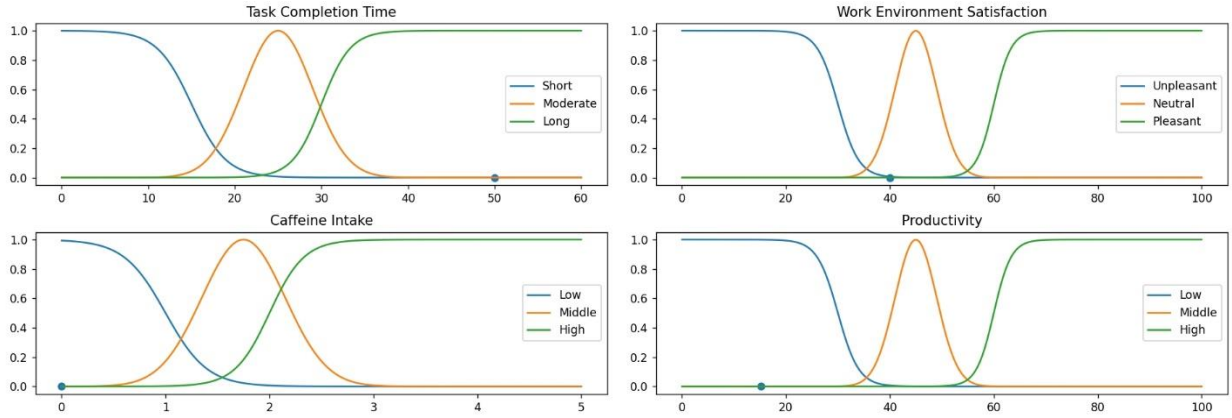

Figure 17. Plot for Rule 5

```
Employee Productivity Level = 15.244453222015359
```

Figure 18. Productivity Level for Rule 5

Rule five states that if CI is low, then productivity is low. We changed inputs from the previous rule, to show this rule. Our inputs are TCT = 50, WES = 40, CI = 0. The productivity level is equal to 15.24, which is low.

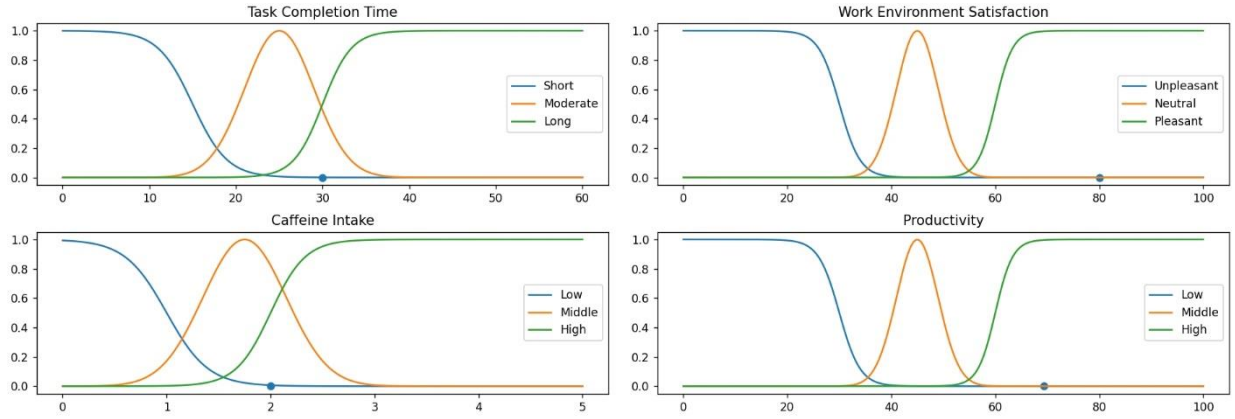

Figure 19. Plot for Rule 6

```
Employee Productivity Level = 69.34309769884149
```

Figure 20. Productivity Level for Rule 6

Rule six states that if TCT is moderate and WES is pleasant, then productivity is high. Our inputs are TCT = 30, WES = 80, CI = 2. The productivity level is 69.34, which is high.

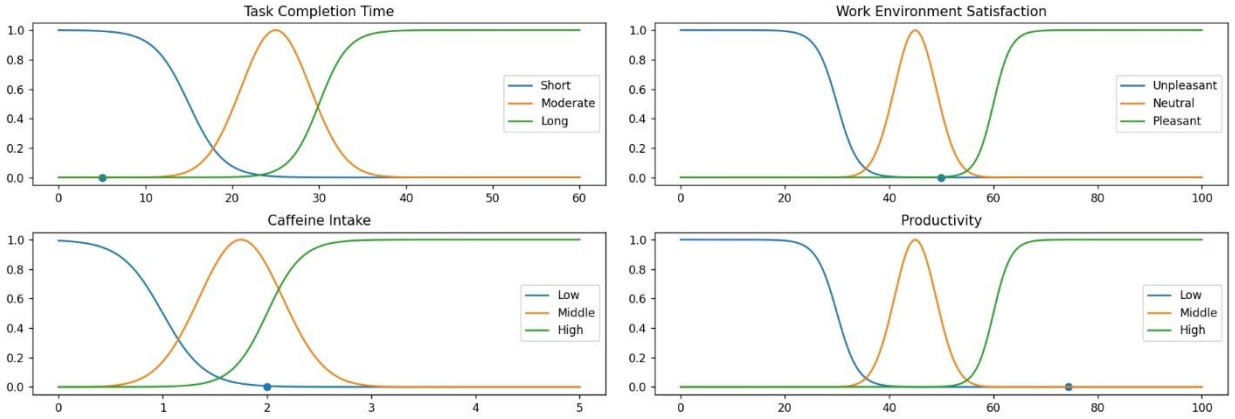

Figure 21. Plot for Rule 7

Figure 22. Productivity Level for Rule 7

Rule seven states that if TCT is short then the productivity level is high. Our inputs are TCT = 5, WES = 50, CI = 2. The productivity level is equal to 74.38, which is high.

4. CONCLUSION

In conclusion, this study shows the success of application of Fuzzy Logic in evaluating employee productivity with three factors task completion time, work environment satisfaction and caffeine intake. The usage of sigmoid and gaussian membership function has proven to be effective in scenario given in the study. The rules show the flexibility and adaptability of Fuzzy Sets in handling decision-making processes.

The effective creation and testing of fuzzy logic have demonstrated its value as a useful tool for addressing the complex interactions between variables affecting worker productivity, opening the door to more sensible and practical approaches.

REFERENCES

Golden, L. (2012). The effects of working time on productivity and firm performance, research synthesis paper. *International Labor Organization (ILO) Conditions of Work and Employment Series*, (33).

Insperity. (n.d.). "8 Reasons Why Employee Productivity May Suffer." Retrieved 4 March 2024 from https://www.insperity.com/blog/8-reasons-why-employee-productivity-may-suffer/

McLellan, T. M., Caldwell, J. A., & Lieberman, H. R. (2016). A review of caffeine's effects on cognitive, physical and occupational performance. *Neuroscience & Biobehavioral Reviews*, *71*, 294-312.

Sleep Foundation. (n.d.). 94 Percent of Us Drink Caffeinated Beverages. Sleep Foundation. https://www.sleepfoundation.org/sleep-news/94-percent-of-us-drink-caffeinated beverages

APPENDIX A

Memberships

```
import numpy as np
import matplotlib.pyplot as plt


def inc(x, a, b):  # increasing membership function
    if x <= a:
        r = 0
    elif x >= b:
        r = 1
    else:
        r = (x - a)/(b - a)
    return r


def dec(x, a, b):  # decreasing membership function (the same as increasing,
but flipped)
    r = 1 - inc(x, a, b)
    return r


def gauss(x, mu, sig):  # gaussian membership function
    r = np.exp(-(x - mu)**2/(2 * sig**2))
    return r


def sigmoid(x, a, b):  # sigmoid membership function
    r = 1/(1 + np.exp(-a * (x - b)))
    return r


def trapmf(x, params):
    a, b, c, d = params
    if x <= a or x >= d:
        r = 0
    elif a < x <= b:
        r = (x - a) / (b - a)
    elif b < x <= c:
        r = 1
    elif c < x <= d:
        r = (d - x) / (d - c)
    else:
        r = 0
    return r


x = np.linspace(0, 10, 100)
y = np.zeros_like(x)  # create zero array with same dimension as x

for i in range(len(x)):
    y[i] = sigmoid(x[i], -4, 5)
```

APPENDIX B

Main Code

```
import numpy as np
import matplotlib.pyplot as plt
import membership as mem

# Fuzzification
tct = np.linspace(0, 60, 200)
short = np.array([mem.sigmoid(x, -0.5, 15) for x in tct])
moderate = np.array([mem.gauss(x, 25, 4) for x in tct])
long = np.array([mem.sigmoid(x, 0.6, 30) for x in tct])

wes = np.linspace(0, 100, 200)
unpleasant = np.array([mem.sigmoid(x, -0.5, 30) for x in wes])
neutral = np.array([mem.gauss(x, 45, 4) for x in wes])
pleasant = np.array([mem.sigmoid(x, 0.6, 60) for x in wes])

coffee = np.linspace(0, 5, 200)
low = np.array([mem.sigmoid(x, -5, 1) for x in coffee])
middle = np.array([mem.gauss(x, 1.75, 0.4) for x in coffee])
high = np.array([mem.sigmoid(x, 6, 2) for x in coffee])

productivity = np.linspace(0, 100, 200)
low_pr = np.array([mem.sigmoid(x, -0.5, 30) for x in productivity])
middle_pr = np.array([mem.gauss(x, 45, 4) for x in productivity])
high_pr = np.array([mem.sigmoid(x, 0.6, 60) for x in productivity])

inp_tct = 5
inp_wes = 50
inp_cf = 2

# Modeling Input
inp_tct_sh = mem.sigmoid(inp_tct, -0.5, 15)
inp_tct_m = mem.gauss(inp_tct, 25, 4)
inp_tct_l = mem.sigmoid(inp_tct, 0.6, 30)

inp_wes_unpl = mem.sigmoid(inp_wes, -0.5, 30)
inp_wes_neut = mem.gauss(inp_wes, 45, 4)
inp_wes_pl = mem.sigmoid(inp_wes, 0.6, 60)

inp_cf_l = mem.sigmoid(inp_cf, -5, 1)
inp_cf_m = mem.gauss(inp_cf, 1.75, 0.4)
inp_cf_h = mem.sigmoid(inp_cf, 6, 2)

# RULE 1
# IF Task Completion Time is Long AND Work Environment Satisfaction is
Unpleasant THEN Productivity Level is Low.
ante = np.min([inp_tct_l, inp_wes_unpl])
r1 = np.fmin(ante, low_pr)

# RULE 2
# IF Caffeine Intake is High THEN Productivity Level is Moderate.
ante = np.min([inp_cf_h])
```

```
r2 = np.fmin(ante, middle_pr)

# RULE 3
# IF Task Completion Time is Short AND Work Environment Satisfaction is
Pleasant THEN Productivity Level is High.
ante = np.min([inp_tct_sh, inp_wes_pl])
r3 = np.fmin(ante, high_pr)

# RULE 4
# IF Work Environment Satisfaction is Neutral AND Caffeine Intake is Moderate
THEN Productivity Level is Moderate.
ante = np.min([inp_wes_neut, inp_cf_m])
r4 = np.fmin(ante, middle_pr)

# RULE 5
# IF Caffeine Intake is Low THEN Productivity Level is Low.
ante = np.min([inp_cf_l])
r5 = np.fmin(ante, low_pr)

# RULE 6
# IF Task Completion Time is Moderate AND Work Environment Satisfaction is
Pleasant THEN Productivity Level is High.
ante = np.min([inp_tct_m, inp_wes_pl])
r6 = np.fmin(ante, high_pr)

# RULE 7
# IF Task Completion Time is Short THEN Productivity Level is High.
ante = np.min([inp_tct_sh])
r7 = np.fmin(ante, high_pr)

# Aggregate all the rules
aggregated = np.maximum(r1, np.maximum(r2, np.maximum(r3, np.maximum(r4,
np.maximum(r5, np.maximum(r6, r7))))))

# Defuzzification using the Centroid method
output_productivity = np.trapz(aggregated * productivity, productivity) /
np.trapz(aggregated, productivity)

print("Employee Productivity Level =", output_productivity)

plt.figure(figsize=(12, 8))

# Task Completion Time
plt.subplot(3, 2, 1)
plt.plot(tct, short, label='Short')
plt.plot(tct, moderate, label='Moderate')
plt.plot(tct, long, label='Long')
plt.scatter(inp_tct, 0)
plt.title('Task Completion Time')
plt.legend()

# Work Environment Satisfaction
plt.subplot(3, 2, 2)
plt.plot(wes, unpleasant, label='Unpleasant')
plt.plot(wes, neutral, label='Neutral')
plt.plot(wes, pleasant, label='Pleasant')
plt.scatter(inp_wes, 0)
```

```
plt.title('Work Environment Satisfaction')
plt.legend()

# Caffeine Intake
plt.subplot(3, 2, 3)
plt.plot(coffee, low, label='Low')
plt.plot(coffee, middle, label='Middle')
plt.plot(coffee, high, label='High')
plt.scatter(inp_cf, 0)
plt.title('Caffeine Intake')
plt.legend()

# Productivity
plt.subplot(3, 2, 4)
plt.plot(productivity, low_pr, label='Low')
plt.plot(productivity, middle_pr, label='Middle')
plt.plot(productivity, high_pr, label='High')
plt.scatter(output_productivity, 0)
plt.title('Productivity')
plt.legend()

# Plots
plt.tight_layout()
plt.show()
```