Tipping Assistant Based on Fuzzy Logic

```
# Define membership functions
def service_rating(x):
    if x <= 3:
        return 0
    elif x <= 5:
        return (x-3) / 2
    elif x <= 7:
        return (x-5) / 2
    else:
        return 1

def food_rating(x):
    if x <= 3:
        return 0
    elif x <= 5:
        return (x-3) / 2
    elif x <= 7:
        return (x-5) / 2
    else:
        return 1

# Define fuzzy rules
def tip_percentage(service, food):
  if service == 1 and food == 1:
        return 25
  elif service == 0 and food == 0:
        return 0
  else:
    low_tip = min(service, food)
    medium_tip = min(service, 1-food)
    high_tip = min(1-service, 1-food)
    return (low_tip*10 + medium_tip*15 + high_tip*20) / (low_tip + medi
um_tip + high_tip)

# Get user inputs for service and food ratings
service = float(input("Enter service rating (0-10): "))
food = float(input("Enter food rating (0-10): "))

# Apply fuzzy logic to calculate tip percentage
service_level = service_rating(service)
food_level = food_rating(food)
tip = tip_percentage(service_level, food_level)

# Print output
print("Based on your ratings, the suggested tip percentage is:", round(
tip, 2), "%")
```

Colab Link:

https://colab.research.google.com/drive/1fvW-xIwSxMktGFmVcajCSweSOoa6F3TY?usp=sharing

This code is an example of a simple fuzzy logic system that calculates the suggested tip percentage based on the quality of service and the quality of food. The code consists of four main components:

Membership Functions

The “service_rating()” and “food_rating()” functions are used to define the membership functions for the fuzzy sets "poor", "average", “good” and "excellent" for both service and food ratings. These functions take in a numerical rating value (between 0 and 10) and return a degree of membership (between 0 and 1) in the corresponding fuzzy set.

If the rating is less than or equal to 3, the degree of membership is 0 for the "poor" fuzzy set.

If the rating is between 3 and 5, the degree of membership is linearly interpolated between 0 and 1 for the "average" fuzzy set.

If the rating is between 5 and 7, the degree of membership is linearly interpolated between 0 and 1 for the "good" fuzzy set.

If the rating is greater than or equal to 7, the degree of membership is 1 for the "excellent" fuzzy set.

Fuzzy Rules

The “tip_percentage()” function takes in the degree of membership values for the service and food fuzzy sets and uses them to calculate a suggested tip percentage. The function uses the following fuzzy rules:

If both service and food are rated as "excellent", the tip percentage is 25%.

If both service and food are rated as "poor", the tip percentage is 0%.

Otherwise, the degree of membership values for the "low", "medium", and "high" tip fuzzy sets are calculated as follows:

Low Tip: The minimum degree of membership between the service and food "poor" fuzzy sets.

Medium Tip: The minimum degree of membership between the service "excellent" fuzzy set and the complement of the food fuzzy set (i.e., the degree of membership in the "not excellent" fuzzy set).

High Tip: The minimum degree of membership between the complement of the service fuzzy set and the complement of the food fuzzy set.

The tip percentage is then calculated as a weighted average of these three fuzzy sets, where the weights are 10%, 15%, and 20%, respectively.

User Inputs

The user is prompted to enter their service and food ratings, which are stored as floating-point values. By getting the user inputs as floats, the code allows for decimal values to be entered, which can help provide more accurate ratings. If the code were to get the user inputs as integers, any decimal values entered would be truncated, potentially resulting in inaccurate ratings.

Output

The suggested tip percentage is calculated using the service and food ratings and printed to the console, rounded to two decimal places.

In summary, this code demonstrates a simple example of a fuzzy logic system that uses membership functions and fuzzy rules to make a decision based on uncertain or imprecise input values. The use of fuzzy logic allows for more flexibility and nuanced decision-making than traditional boolean logic.