Title: Fuzzy Logic-Based Parking Availability Prediction

1.Method:

- Fuzzy System Definition:
  - Antecedents: Occupancy Level and Time of Day
  - Consequent: Parking Available
  - Membership Functions:
    - Occupancy Level: Low, Medium, High
    - Time of Day: Off-Peak, Peak
    - Parking Available: Low, Medium, High
  - Fuzzy Rules:
    - Rule 1: IF (Occupancy is Low AND Time of Day is Off-Peak) THEN (Parking Availability is High)
    - Rule 2: IF (Occupancy is Medium AND Time of Day is Off-Peak) THEN (Parking Availability is Medium)
    - Rule 3: IF (Occupancy is High AND Time of Day is Off-Peak) THEN (Parking Availability is Low)
    - Rule 4: IF (Occupancy is Low AND Time of Day is Peak) THEN (Parking Availability is Medium)
    - Rule 5: IF (Occupancy is Medium AND Time of Day is Peak) THEN (Parking Availability is Low)
    - Rule 6: IF (Occupancy is High AND Time of Day is Peak) THEN (Parking Availability is Low)
  - Simulation:
    - A Fuzzy control system is created using the defined rules.
    - Random values are assigned to Occupancy Level and Time of Day.
    - The Fuzzy system is simulated, and the Parking Available is completed.

2.Result

- Sample Output

1.

```
Input Occupancy Level: 10.94
Input Time of Day: 7.83
Parking Availability: 66.24%
```

2.

```
Input Occupancy Level: 28.33
Input Time of Day: 2.01
Parking Availability: 53.49%
```

3.

```
Input Occupancy Level: 97.69
Input Time of Day: 13.72
Parking Availability: 23.99%
```

4.

```
Input Occupancy Level: 6.07
Input Time of Day: 13.77
Parking Availability: 69.48%
```

5.

```
Input Occupancy Level: 62.13
Input Time of Day: 3.32
Parking Availability: 48.20%
```

6.

```
Input Occupancy Level: 84.03
Input Time of Day: 4.94
Parking Availability: 38.35%
```

3. Discussion:

- Interpretation of output:
    - The fuzzy logic system provides a percentage indicating the predicted parking availability base on the random input values.
    - In the giving sample output, from 2^{nd} (Result)
- Flexibility and Adaptability:
    - The fuzzy system allows for flexibility in handling imprecise input data, making it suitable for real-world scenarios where parking availability depends on multiple factors.
    - The system can adapt to different input combinations, providing a versatile tool for parking prediction.
- Feature Exploration:
    - The fuzzy logic system can be further enhanced by adding more rules and refining membership functions to improve prediction accuracy.
    - Exploring the impact of additional input variables, such as day of the week or special events, could enhance the system's predictive capabilities.

4.Conclusion:

- The fuzzy logic-based parking availability prediction system demonstrates the feasibility of using fuzzy logic to model and predict parking availability. The system's adaptability and ability to handle imprecise input make it a valuable tool for real-world applications. Further refinement and exploration can contribute to improving the accuracy and reliability of parking predictions.

Code

```python
# Import necessary libraries
import numpy as np
import skfuzzy as fuzz    Import "skfuzzy" could not be resolved
from skfuzzy import control as ctrl    Import "skfuzzy" could not be resolved
import matplotlib.pyplot as plt


# Define fuzzy system
occupancy_level = ctrl.Antecedent(np.arange(0, 101, 1), 'occupancy_level')
time_of_day = ctrl.Antecedent(np.arange(0, 24, 1), 'time_of_day')
parking_availability = ctrl.Consequent(np.arange(0, 101, 1), 'parking_availability')


# Define membership functions
occupancy_level['low'] = fuzz.trimf(occupancy_level.universe, [0, 0, 50])
occupancy_level['medium'] = fuzz.trimf(occupancy_level.universe, [0, 50, 100])
occupancy_level['high'] = fuzz.trimf(occupancy_level.universe, [50, 100, 100])


time_of_day['off_peak'] = fuzz.trimf(time_of_day.universe, [0, 7, 24])
time_of_day['peak'] = fuzz.trimf(time_of_day.universe, [7, 12, 19])



parking_availability['low'] = fuzz.trimf(parking_availability.universe, [0, 0, 50])
parking_availability['medium'] = fuzz.trimf(parking_availability.universe, [0, 50, 100])
parking_availability['high'] = fuzz.trimf(parking_availability.universe, [50, 100, 100])


# Define fuzzy rules
rule1 = ctrl.Rule(occupancy_level['low'] & time_of_day['off_peak'], parking_availability['high'])
rule2 = ctrl.Rule(occupancy_level['medium'] & time_of_day['off_peak'], parking_availability['medium'])
rule3 = ctrl.Rule(occupancy_level['high'] & time_of_day['off_peak'], parking_availability['low'])

# Create control system
parking_ctrl = ctrl.ControlSystem([rule1, rule2, rule3])
parking_sim = ctrl.ControlSystemSimulation(parking_ctrl)


# Set random input values
input_occupancy = np.random.uniform(0, 100)
input_time_of_day = np.random.uniform(0, 24)
parking_sim.input['occupancy_level'] = input_occupancy
parking_sim.input['time_of_day'] = input_time_of_day



# Compute the result
parking_sim.compute()
result = parking_sim.output['parking_availability']

# For loop on time
for i in range(5):
    # create a new Simulator
    parking_sim = ctrl.ControlSystemSimulation(parking_ctrl)

    # set random input value
    input_occupancy = np.random.uniform(0,100)
    input_time_of_day = np.random.uniform(0.24)

    parking_sim.input['occupancy_level' ] = input_occupancy
    parking_sim.input['time_of_day'] = input_time_of_day


    # Compute the result
    parking_sim.compute()
    result =parking_sim.output['parking_availability']

    # Display the result as a percentage
    percentage_result = result / 100.0  # assuming the result is on a 0-100 scale
    print(f"Input Occupancy Level: {input_occupancy:.2f}")
    print(f"Input Time of Day: {input_time_of_day:.2f}")
    print("Parking Availability: {:.2%}".format(percentage_result))
    print("-" * 30)


# Plot the fuzzy membership functions and the result
parking_availability.view(sim=parking_sim)
occupancy_level.view()
time_of_day.view()

plt.show()
```