import numpy as np

import scipy.stats as stats

# Historical flow data (vehicles/hour)

flow\_data = np.array([1200, 1300, 1250, 1400, 1500, 1600, 1450, 1350, 1550, 1650])

# Calculate the mean

mean\_flow = np.mean(flow\_data)

# Calculate the standard deviation

std\_flow = np.std(flow\_data, ddof=1) # Using ddof=1 for sample standard deviation

# Sample size

n = len(flow\_data)

# Z-score for 95% confidence interval

z\_score = stats.norm.ppf(0.975) # 1.96

# Margin of error

margin\_of\_error = z\_score \* (std\_flow / np.sqrt(n))

# Confidence interval

lower\_bound = mean\_flow - margin\_of\_error

upper\_bound = mean\_flow + margin\_of\_error

# Function to classify traffic conditions

def classify\_traffic(observed\_flow):

if observed\_flow > upper\_bound:

return "Crowded"

else:

return "Not Crowded"

# Example of classification

observed\_flow = 1550

traffic\_condition = classify\_traffic(observed\_flow)

# Print results

print(f"Mean Flow: {mean\_flow:.2f} vehicles/hour")

print(f"Standard Deviation: {std\_flow:.2f} vehicles/hour")

print(f"95% Confidence Interval: [{lower\_bound:.2f}, {upper\_bound:.2f}] vehicles/hour")

print(f"Observed Flow: {observed\_flow} vehicles/hour")

print(f"Traffic Condition: {traffic\_condition}")

Explanation

Data Preparation:

flow\_data: An array of historical flow data.

Calculations:

Mean: The average flow.

Standard Deviation: The variability of the flow data.

Z-score: The Z-score for the 95% confidence level.

Margin of Error: Calculated using the Z-score, standard deviation, and sample size.

Confidence Interval: The range within which we expect the true mean flow to lie 95% of the time.

Classification:

The function classify\_traffic takes an observed flow value and classifies it as "Crowded" or "Not Crowded" based on the confidence interval.

Example Usage:

The example observed flow is 1550 vehicles/hour. The code classifies this as "Crowded" if it exceeds the upper bound of the confidence interval.