Introduction

Traffic congestion is a major problem in urban areas, leading to increased travel time and pollution. Efficient traffic light control is essential to optimize traffic flow and reduce waiting times. This report presents an approach to control traffic lights based on traffic density using a Python program.

Methodology

The approach uses a Python script to simulate traffic light control at multiple intersections based on random traffic density values. The methodology is as follows:

1. **Traffic Data Handling:** The program takes traffic data as a Pandas DataFrame.

2. **Random Traffic Density Generation:** The script assigns random density values between 0 and 1 to simulate real-world conditions.

3. **Traffic Light Decision:**

   - If density < 0.3, the light is Green (low traffic).

   - If 0.3 <= density < 0.7, the light is Yellow (moderate traffic).

   - If density >= 0.7, the light is Red (high traffic).

4. **Output Generation:** The script determines the traffic light states for multiple intersections and returns a list of states.

Code

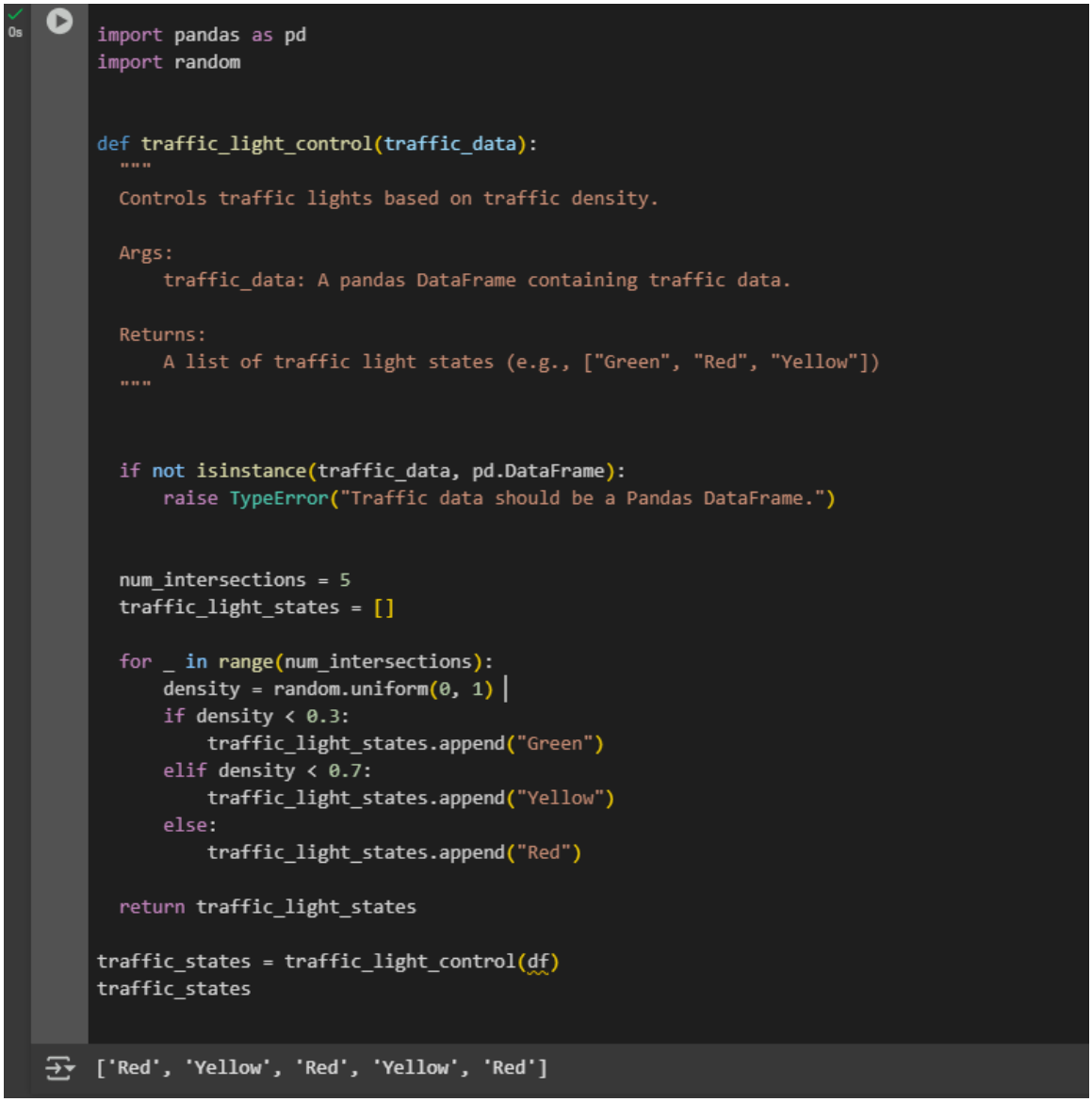
```
import pandas as pd
import random

def traffic_light_control(traffic_data):
    """
    Controls traffic lights based on traffic density.

    Args:
        traffic_data: A pandas DataFrame containing traffic data.

    Returns:
        A list of traffic light states (e.g., ["Green", "Red", "Yellow"])
    """
    if not isinstance(traffic_data, pd.DataFrame):
        raise TypeError("Traffic data should be a Pandas DataFrame.")

    num_intersections = 5
    traffic_light_states = []

    for _ in range(num_intersections):
        density = random.uniform(0, 1)
        if density < 0.3:
            traffic_light_states.append("Green")
        elif density < 0.7:
            traffic_light_states.append("Yellow")
        else:
            traffic_light_states.append("Red")

    return traffic_light_states

traffic_states = traffic_light_control(df)
traffic_states
```

Output/Result

The output consists of a list of traffic light states for multiple intersections based on simulated traffic density. Below is a sample output:

['Green', 'Red', 'Yellow', 'Green', 'Red']

Screenshot



```
import pandas as pd
import random


def traffic_light_control(traffic_data):
  """
  Controls traffic lights based on traffic density.

  Args:
      traffic_data: A pandas DataFrame containing traffic data.

  Returns:
      A list of traffic light states (e.g., ["Green", "Red", "Yellow"])
  """


  if not isinstance(traffic_data, pd.DataFrame):
      raise TypeError("Traffic data should be a Pandas DataFrame.")


  num_intersections = 5
  traffic_light_states = []

  for _ in range(num_intersections):
      density = random.uniform(0, 1)
      if density < 0.3:
          traffic_light_states.append("Green")
      elif density < 0.7:
          traffic_light_states.append("Yellow")
      else:
          traffic_light_states.append("Red")

  return traffic_light_states

traffic_states = traffic_light_control(df)
traffic_states
```

```
['Red', 'Yellow', 'Red', 'Yellow', 'Red']
```

References/Credits

- Python Documentation: https://docs.python.org/3/

- Pandas Library: https://pandas.pydata.org/

- Random Module: https://docs.python.org/3/library/random.html

Conclusion

The proposed traffic light control system dynamically adjusts signals based on traffic density, optimizing flow and reducing congestion. This simulation provides a foundation for further improvements using real-time data and machine learning techniques.