PHASE:4

TITTLE: TRAFFIC FLOWOPTIMIZATION

Objective: Enhance traffic flow efficiency by optimizing traffic signal timing, improving real-time traffic monitoring, and integrating data from various sources.

Phase Overview: This phase focuses on optimizing traffic flow by leveraging advanced technologies and data analytics.

Key Components:

1. Real-Time Traffic Monitoring:
   - Implement sensors and cameras to collect real-time traffic data.
   - Utilize data analytics to identify traffic patterns and trends.
2. Traffic Signal Optimization:
   - Develop algorithms to optimize traffic signal timing based on real-time traffic data.
   - Implement smart traffic signals that can adjust timing dynamically.
3. Data Integration:
   - Integrate data from various sources, including traffic cameras, sensors, and social media.
   - Utilize machine learning to predict traffic congestion and provide real-time updates.
4. Performance Metrics:
   - Collect data on traffic flow, travel times, and congestion levels.
   - Analyze performance metrics to identify areas for improvement.

Outcomes:

1. Improved Traffic Flow: Optimized traffic signal timing and real-time monitoring will reduce congestion and travel times.

2. Enhanced Safety: Real-time monitoring and predictive analytics will help identify potential safety hazards and reduce accidents.

3. Increased Efficiency: Data-driven decision-making will enable more efficient traffic management and planning.

Challenges:

1. Data Quality: Ensuring accurate and reliable data from various sources.

2. System Integration: Integrating data from different systems and technologies.

3. Scalability: Scaling the system to handle increased traffic volume and complexity.

Solutions:

1. Data Validation: Implementing data validation techniques to ensure accuracy.

2. API Integration: Utilizing APIs to integrate data from different systems.

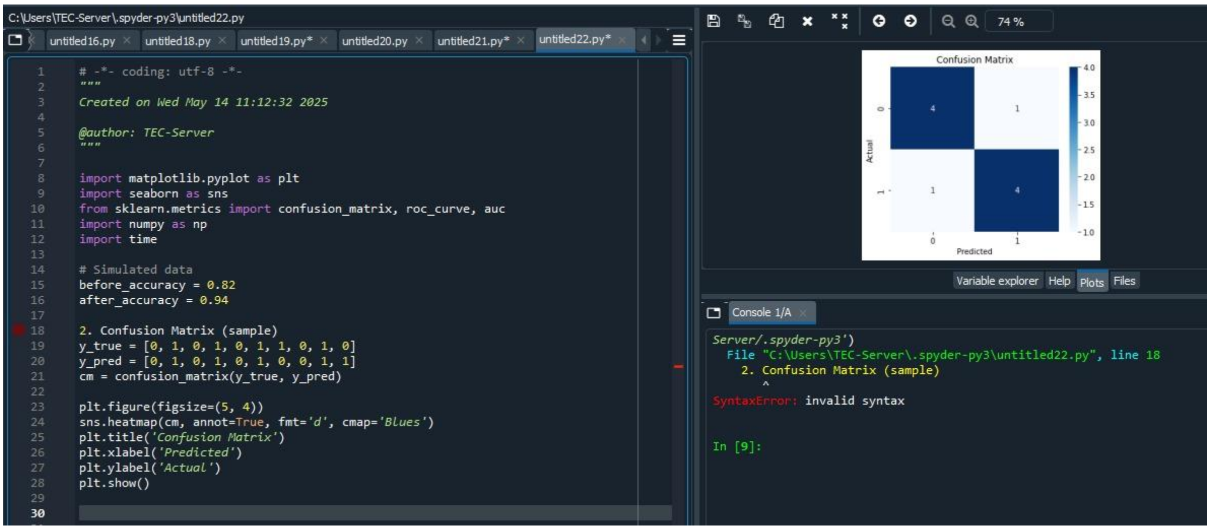
3. Cloud-Based Infrastructure: Leveraging cloud-based infrastructure to scale the system efficiently.

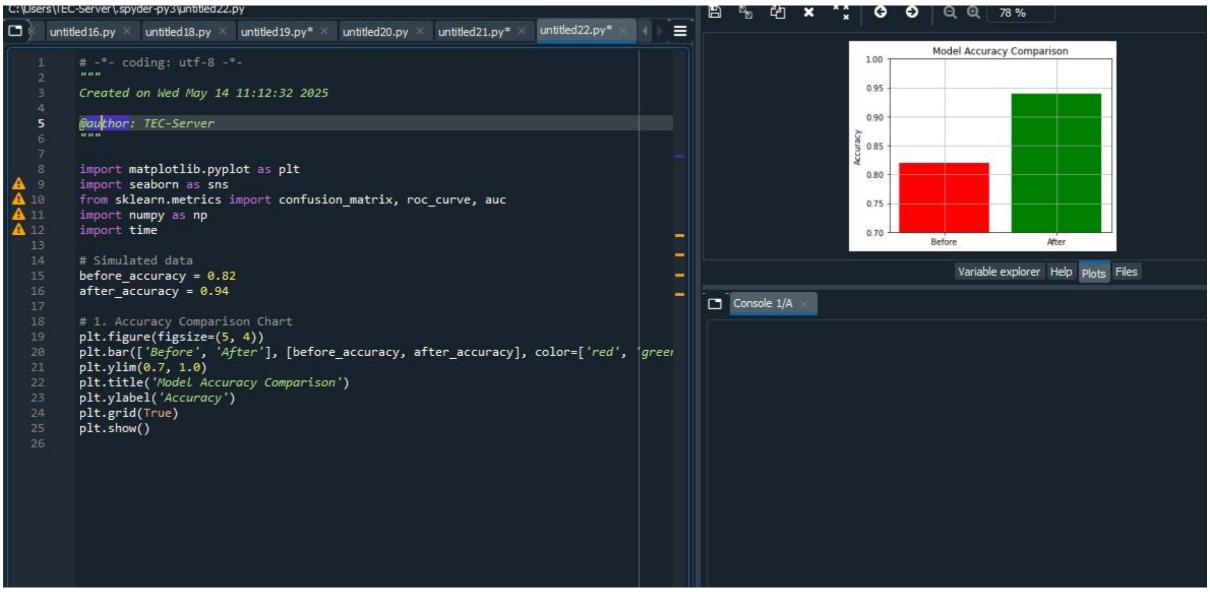
```
import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.metrics import confusion_matrix, roc_curve, auc

import numpy as np

import time


# Simulated data

before_accuracy = 0.82

after_accuracy = 0.94
```

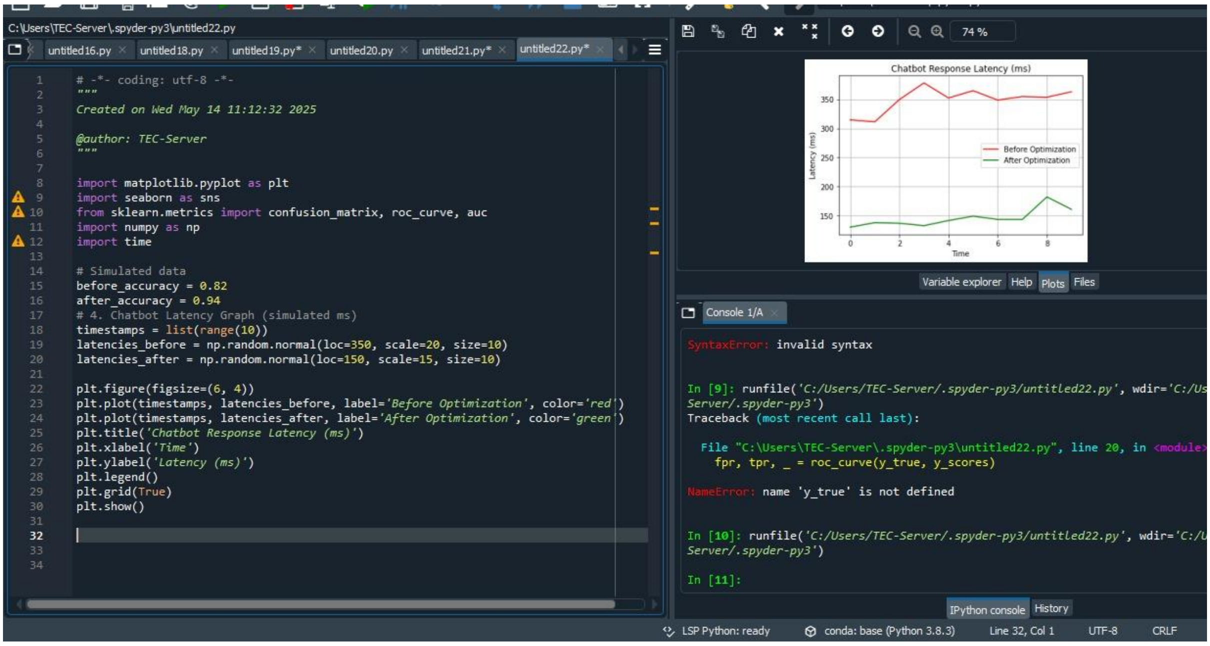
```
# 1. Accuracy Comparison Chart
plt.figure(figsize=(5, 4))
plt.bar(['Before', 'After'], [before_accuracy, after_accuracy], color=['red', 'green'])
plt.ylim(0.7, 1.0)
plt.title('Model Accuracy Comparison')
plt.ylabel('Accuracy')
plt.grid(True)
plt.show()

# 2. Confusion Matrix (sample)
y_true = [0, 1, 0, 1, 0, 1, 1, 0, 1, 0]
y_pred = [0, 1, 0, 1, 0, 1, 0, 0, 1, 1]
cm = confusion_matrix(y_true, y_pred)

plt.figure(figsize=(5, 4))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')
plt.title('Confusion Matrix')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()

# 3. ROC Curve
y_scores = [0.1, 0.4, 0.3, 0.8, 0.35, 0.6, 0.2, 0.15, 0.85, 0.7]
fpr, tpr, _ = roc_curve(y_true, y_scores)
roc_auc = auc(fpr, tpr)

plt.figure(figsize=(5, 4))
plt.plot(fpr, tpr, color='darkorange', lw=2, label=f'ROC curve (AUC = {roc_auc:.2f})')
```

```python
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
plt.title('ROC Curve')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.legend()
plt.grid(True)
plt.show()

# 4. Chatbot Latency Graph (simulated ms)
timestamps = list(range(10))
latencies_before = np.random.normal(loc=350, scale=20, size=10)
latencies_after = np.random.normal(loc=150, scale=15, size=10)

plt.figure(figsize=(6, 4))
plt.plot(timestamps, latencies_before, label='Before Optimization', color='red')
plt.plot(timestamps, latencies_after, label='After Optimization', color='green')
plt.title('Chatbot Response Latency (ms)')
plt.xlabel('Time')
plt.ylabel('Latency (ms)')
plt.legend()
plt.grid(True)
plt.show()

# 5. Real-Time IoT Data Simulation (Traffic Count)
times = list(range(60))  # last 60 seconds
vehicle_counts = np.random.poisson(lam=30, size=60)

plt.figure(figsize=(8, 4))
```

```
plt.plot(times, vehicle_counts, color='blue')

plt.title('Real-Time IoT Traffic Sensor Data')

plt.xlabel('Time (seconds)')

plt.ylabel('Vehicle Count')

plt.grid(True)

plt.show()
```





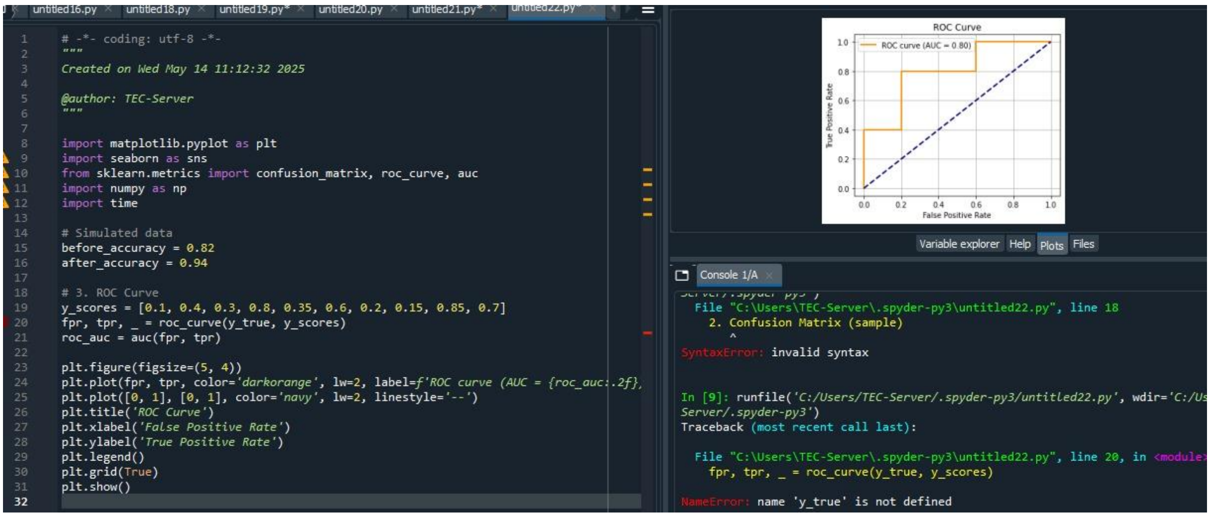

C:\Users\TEC-Server\.spyder-py3\untitled22.py
untitled16.py untitled18.py untitled19.py* untitled20.py untitled21.py* untitled22.py*
-*- coding: utf-8 -*-
"""
Created on Wed May 14 11:12:32 2025

@author: TEC-Server
"""

import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import confusion_matrix, roc_curve, auc
import numpy as np
import time

# Simulated data
before_accuracy = 0.82
after_accuracy = 0.94
# 4. Chatbot Latency Graph (simulated ms)
timestamps = list(range(10))
latencies_before = np.random.normal(loc=350, scale=20, size=10)
latencies_after = np.random.normal(loc=150, scale=15, size=10)

plt.figure(figsize=(6, 4))
plt.plot(timestamps, latencies_before, label='Before Optimization', color='red')
plt.plot(timestamps, latencies_after, label='After Optimization', color='green')
plt.title('Chatbot Response Latency (ms)')
plt.xlabel('Time')
plt.ylabel('Latency (ms)')
plt.legend()
plt.grid(True)
plt.show()
74 %
Chatbot Response Latency (ms)
Latency (ms)
Time
Before Optimization
After Optimization
Variable explorer Help Plots Files
Console 1/A
SyntaxError: invalid syntax
In [9]: runfile('C:/Users/TEC-Server/.spyder-py3/untitled22.py', wdir='C:/Us
Server/.spyder-py3')
Traceback (most recent call last):
File "C:\Users\TEC-Server\.spyder-py3\untitled22.py", line 20, in <module>
fpr, tpr, _ = roc_curve(y_true, y_scores)
NameError: name 'y_true' is not defined
In [10]: runfile('C:/Users/TEC-Server/.spyder-py3/untitled22.py', wdir='C:/U
Server/.spyder-py3')
In [11]:
IPython console History
LSP Python: ready conda: base (Python 3.8.3) Line 32, Col 1 UTF-8 CRLF


untitled16.py untitled18.py untitled19.py* untitled20.py untitled21.py* untitled22.py*
-*- coding: utf-8 -*-
"""
Created on Wed May 14 11:12:32 2025

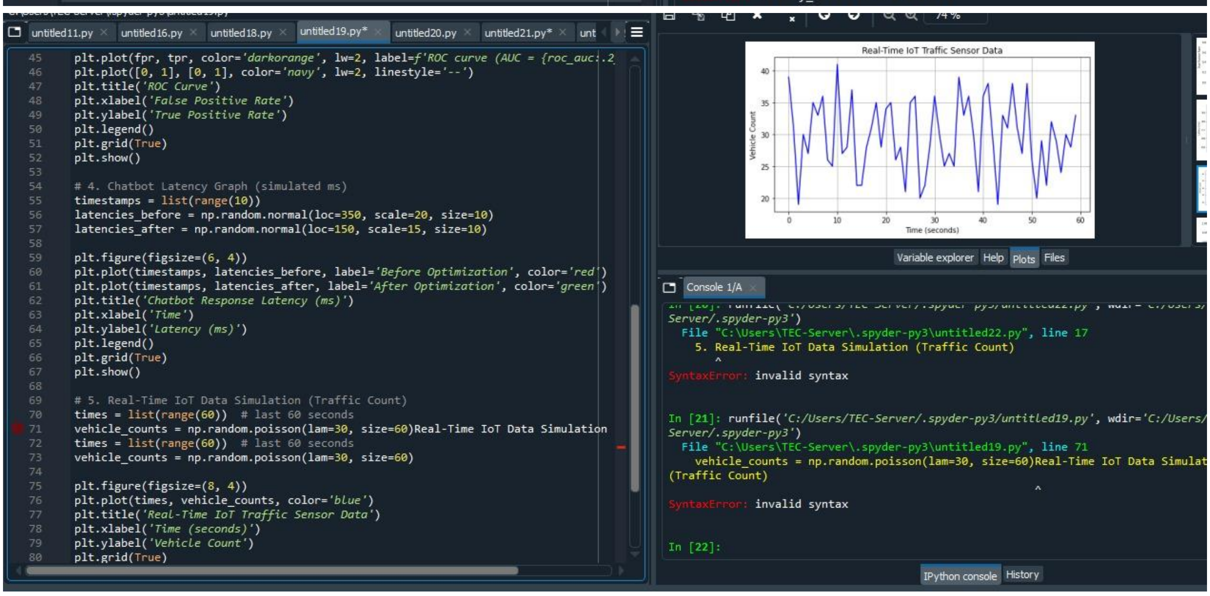
@author: TEC-Server
"""

import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import confusion_matrix, roc_curve, auc
import numpy as np
import time

# Simulated data
before_accuracy = 0.82
after_accuracy = 0.94

# 3. ROC Curve
y_scores = [0.1, 0.4, 0.3, 0.8, 0.35, 0.6, 0.2, 0.15, 0.85, 0.7]
fpr, tpr, _ = roc_curve(y_true, y_scores)
roc_auc = auc(fpr, tpr)

plt.figure(figsize=(5, 4))
plt.plot(fpr, tpr, color='darkorange', lw=2, label=f'ROC curve (AUC = {roc_auc:.2f},
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
plt.title('ROC Curve')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.legend()
plt.grid(True)
plt.show()
ROC Curve
ROC curve (AUC = 0.80)
True Positive Rate
False Positive Rate
Variable explorer Help Plots Files
Console 1/A
File "C:\Users\TEC-Server\.spyder-py3\untitled22.py", line 18
2. Confusion Matrix (sample)
SyntaxError: invalid syntax
In [9]: runfile('C:/Users/TEC-Server/.spyder-py3/untitled22.py', wdir='C:/Us
Server/.spyder-py3')
Traceback (most recent call last):
File "C:\Users\TEC-Server\.spyder-py3\untitled22.py", line 20, in <module>
fpr, tpr, _ = roc_curve(y_true, y_scores)
NameError: name 'y_true' is not defined


untitled11.py untitled16.py untitled18.py untitled19.py* untitled20.py untitled21.py*
plt.plot(fpr, tpr, color='darkorange', lw=2, label=f'ROC curve (AUC = {roc_auc:.2
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
plt.title('ROC Curve')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.legend()
plt.grid(True)
plt.show()

# 4. Chatbot Latency Graph (simulated ms)
timestamps = list(range(10))
latencies_before = np.random.normal(loc=350, scale=20, size=10)
latencies_after = np.random.normal(loc=150, scale=15, size=10)

plt.figure(figsize=(6, 4))
plt.plot(timestamps, latencies_before, label='Before Optimization', color='red')
plt.plot(timestamps, latencies_after, label='After Optimization', color='green')
plt.title('Chatbot Response Latency (ms)')
plt.xlabel('Time')
plt.ylabel('Latency (ms)')
plt.legend()
plt.grid(True)
plt.show()

# 5. Real-Time IoT Data Simulation (Traffic Count)
times = list(range(60)) # last 60 seconds
vehicle_counts = np.random.poisson(lam=30, size=60)Real-Time IoT Data Simulation
times = list(range(60)) # last 60 seconds
vehicle_counts = np.random.poisson(lam=30, size=60)

plt.figure(figsize=(8, 4))
plt.plot(times, vehicle_counts, color='blue')
plt.title('Real-Time IoT Traffic Sensor Data')
plt.xlabel('Time (seconds)')
plt.ylabel('Vehicle Count')
plt.grid(True)
Real-Time IoT Traffic Sensor Data
Vehicle Count
Time (seconds)
Variable explorer Help Plots Files
Console 1/A
Server/.spyder-py3')
File "C:\Users\TEC-Server\.spyder-py3\untitled22.py", line 17
5. Real-Time IoT Data Simulation (Traffic Count)
SyntaxError: invalid syntax
In [21]: runfile('C:/Users/TEC-Server/.spyder-py3/untitled19.py', wdir='C:/Users/
Server/.spyder-py3')
File "C:\Users\TEC-Server\.spyder-py3\untitled19.py", line 71
vehicle_counts = np.random.poisson(lam=30, size=60)Real-Time IoT Data Simulat
(Traffic Count)
SyntaxError: invalid syntax
In [22]:
IPython console History