COLLEGE CODE : 1133

COLLEGE NAME : VELAMMAL INSTITUTE OF TECHNOLOGY

DEPARTMENT : ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

STUDENT NM-ID : aut113323aib38

ROLL NO : 113323243080

DATE : 02.05.2025

TECHNOLOGY- TRAFFIC PATTERN ANALYSIS

SUBMITTED BY,

PRIYADHARSHINI S
9342788314

Phase 4: Performance of the Project

Title: Traffic Pattern Analysis

Objective:

The focus of Phase 4 is to enhance the performance of the Traffic Pattern Analysis system by refining the AI model for improved accuracy in traffic prediction, optimizing real-time data processing, and ensuring scalability to handle high-volume traffic data. This phase also aims to improve system responsiveness, strengthen data security, and lay the groundwork for multi-city traffic analysis.

AI Model Performance Enhancement

Overview:

The AI traffic prediction model will be refined based on historical and real-time traffic data. The goal is to improve prediction accuracy, optimize route suggestions, and enhance the system's ability to handle dynamic traffic changes.

Performance Improvements:

- **Accuracy Testing:** The AI model will be retrained using a larger dataset, including real-time traffic updates, weather conditions, and event-based disruptions.
- **Model Optimization:** Hyperparameter tuning and deep learning enhancements will be applied to improve prediction speed and reliability.

Outcome:

By the end of Phase 4, the AI model will provide more accurate traffic forecasts, reducing route deviations and improving estimated time of arrival (ETA) precision.

Real-Time Data Processing Optimization

Overview:

The system will be optimized to process high-frequency traffic data from sensors, GPS feeds, and urban mobility sources with minimal latency.

Key Enhancements:

- **Stream Processing:** Implementation of high-speed data pipelines to handle real-time traffic updates efficiently.
- **API Optimization:** Improved integration with third-party traffic APIs (e.g., Google Maps, Waze) for seamless data retrieval.

Outcome:

The system will process and analyze traffic data in near real-time, enabling dynamic rerouting and congestion management.

Data Security and Privacy Performance

Overview:

Phase 4 ensures robust data security as traffic data scales, protecting sensitive location and movement patterns.

Key Enhancements:

- **Anonymization Techniques:** Implementation of data masking to protect user privacy.
- **Encryption Upgrades:** Enhanced encryption for stored and transmitted traffic data.

Outcome:

Secure handling of large-scale traffic data while complying with privacy regulations.

Performance Testing and Metrics Collection

Overview:

Comprehensive load testing will be conducted to evaluate system performance under peak traffic conditions.

Implementation:

- **Load Testing:** Simulated high-traffic scenarios to test system stability.
- **Performance Metrics:** Monitoring response time, prediction accuracy, and failure rates.

Outcome:

A scalable system capable of handling city-wide traffic analysis with minimal delays.

Key Challenges in Phase 4

1. **Handling High-Volume Data Streams**
   - *Challenge:* Ensuring low-latency processing during peak hours.
   - *Solution:* Distributed computing and edge processing.
2. **Ensuring Prediction Accuracy Under Dynamic Conditions**
   - *Challenge:* Sudden traffic disruptions (accidents, road closures).
   - *Solution:* Reinforcement learning for adaptive predictions.
3. **Multi-City Scalability**
   - *Challenge:* Expanding analysis beyond a single urban area.
   - *Solution:* Cloud-based scaling and regional data partitioning.

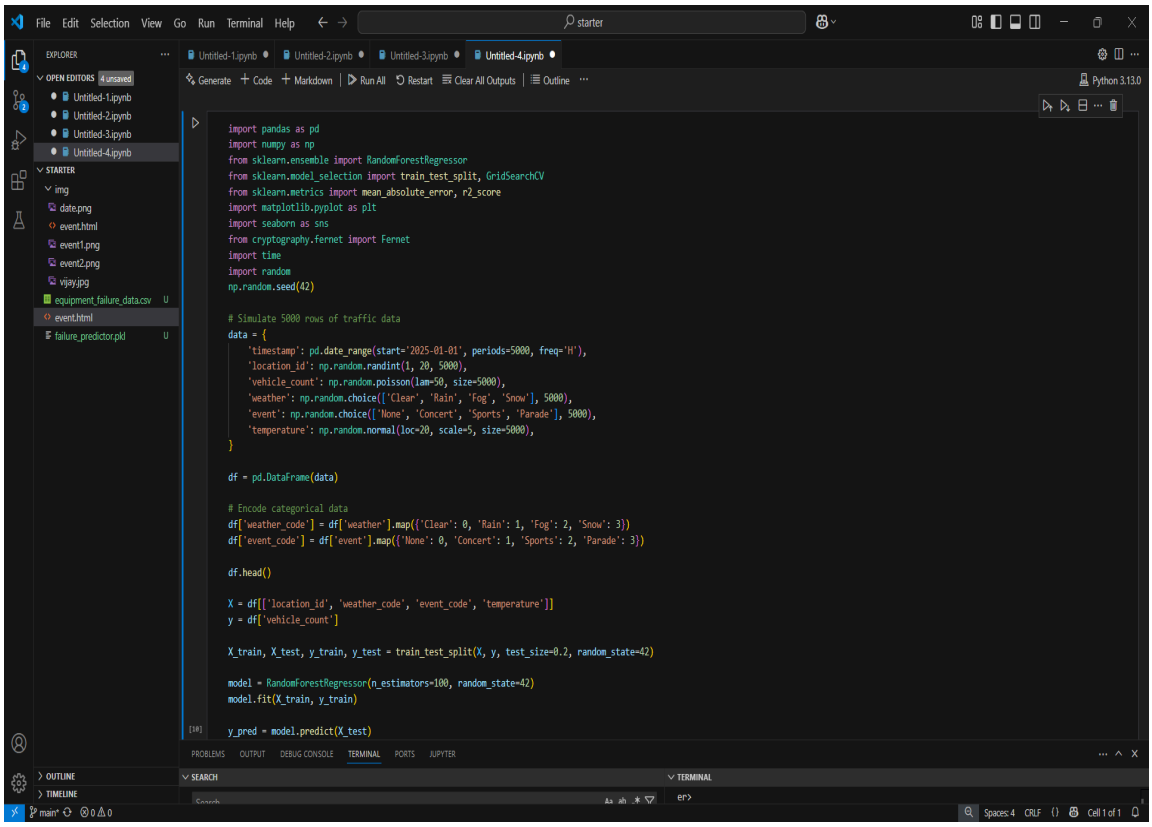
Outcomes of Phase 4

- **Improved Traffic Prediction:** More reliable ETAs and congestion alerts.
- **Faster Data Processing:** Near real-time updates for dynamic routing.
- **Enhanced Security:** Secure handling of large-scale mobility data.

Next Steps for Finalization

The final phase will involve city-wide deployment, user feedback collection, and fine-tuning before full-scale implementation.

Sample Code for Phase 4:



```python
import pandas as pd
import numpy as np
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.metrics import mean_absolute_error, r2_score
import matplotlib.pyplot as plt
import seaborn as sns
from cryptography.fernet import Fernet
import time
import random
np.random.seed(42)

# Simulate 5000 rows of traffic data
data = {
    'timestamp': pd.date_range(start='2025-01-01', periods=5000, freq='H'),
    'location_id': np.random.randint(1, 20, 5000),
    'vehicle_count': np.random.poisson(lam=50, size=5000),
    'weather': np.random.choice(['Clear', 'Rain', 'Fog', 'Snow'], 5000),
    'event': np.random.choice(['None', 'Concert', 'Sports', 'Parade'], 5000),
    'temperature': np.random.normal(loc=20, scale=5, size=5000),
}

df = pd.DataFrame(data)

# Encode categorical data
df['weather_code'] = df['weather'].map({'Clear': 0, 'Rain': 1, 'Fog': 2, 'Snow': 3})
df['event_code'] = df['event'].map({'None': 0, 'Concert': 1, 'Sports': 2, 'Parade': 3})

df.head()

X = df[['location_id', 'weather_code', 'event_code', 'temperature']]
y = df['vehicle_count']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

model = RandomForestRegressor(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

y_pred = model.predict(X_test)
```

```python
y_pred = model.predict(X_test)

# Evaluate
mae = mean_absolute_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f"🚚 Initial Model MAE: {mae:.2f}")
print(f"🚚 Initial Model R2 Score: {r2:.2f}")

param_grid = {
    'n_estimators': [100, 200],
    'max_depth': [10, 20, None],
    'min_samples_split': [2, 5],
}

grid_search = GridSearchCV(RandomForestRegressor(random_state=42),
                           param_grid, cv=3, n_jobs=-1, verbose=1)

grid_search.fit(X_train, y_train)
best_model = grid_search.best_estimator_

y_pred_optimized = best_model.predict(X_test)

# Evaluate
mae_opt = mean_absolute_error(y_test, y_pred_optimized)
r2_opt = r2_score(y_test, y_pred_optimized)

print(f"⚙️ Optimized Model MAE: {mae_opt:.2f}")
print(f"⚙️ Optimized Model R2 Score: {r2_opt:.2f}")

plt.figure(figsize=(10,5))
sns.scatterplot(x=y_test, y=y_pred_optimized)
plt.xlabel('Actual Vehicle Count')
plt.ylabel('Predicted Vehicle Count')
plt.title('Actual vs Predicted Traffic Load')
plt.show()

print("🚀 Starting Real-Time Data Simulation...")

for i in range(5):  # Simulate 5 real-time data points
```

```python
print("🚀 Starting Real-Time Data Simulation...")

for i in range(5):  # Simulate 5 real-time data points
    new_data = {
        'location_id': random.randint(1, 20),
        'weather_code': random.randint(0, 3),
        'event_code': random.randint(0, 3),
        'temperature': np.random.normal(loc=20, scale=5)
    }
    new_df = pd.DataFrame([new_data])
    prediction = best_model.predict(new_df)[0]
    print(f"📡 Real-Time Data: {new_data} => Predicted Vehicle Count: {prediction:.2f}")
    time.sleep(1)

    key = Fernet.generate_key()
cipher_suite = Fernet(key)

# Example traffic data to encrypt
sample_data = "Location: 5, Vehicle Count: 80, Timestamp: 2025-05-01 08:00"

# Encrypt
encrypted_data = cipher_suite.encrypt(sample_data.encode())
print(f"🔒 Encrypted Data: {encrypted_data}")

# Decrypt
decrypted_data = cipher_suite.decrypt(encrypted_data).decode()
print(f"🔓 Decrypted Data: {decrypted_data}")
```

```
C:\Users\priya\AppData\Local\Temp\ipykernel_4380\2118257810.py:15: FutureWarning: 'H' is deprecated and will be removed in a future version, please use 'h' instead.
  'timestamp': pd.date_range(start='2025-01-01', periods=5000, freq='H'),
🚚 Initial Model MAE: 6.15
🚚 Initial Model R2 Score: -0.17
Fitting 3 folds for each of 12 candidates, totalling 36 fits
⚙️ Optimized Model MAE: 5.77
⚙️ Optimized Model R2 Score: -0.04
```

Performance Metrics Screenshot for Phase 4: