

COLLEGE CODE : 1133

COLLEGE NAME : VELAMMAL INSTITUTE OF TECHNOLOGY

DEPARTMENT : ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

STUDENT NM-ID : aut113323aia05

ROLL NO : 113323243015

DATE : 03.05.2025

TECHNOLOGY-PROJECT NAME: TRAFFIC PATTERN ANALYSIS

SUBMITTED BY :

- 1. DHARSHINI S**
- 2. ENAPA JEEVA SRI**
- 3. JAGADEESHWARI N**
- 4. NITHYA S**
- 5. UMA MAHESHWARI C**

INDEX OF THE DOCUMENTS

S.NO	CONTENT OF THE DOCUMENTS	PAGE NO
1	Title, Abstract, Project Demonstration	
2	Project Documentation, Feedback, and Final Adjustments	
3	Final Project Report Submission, Project Handover, and Future Works	
4	Handover Details, Sample Code	
5	Code, Outcomes	
6	Outcomes	

Phase 5: Project Demonstration & Documentation

Title: Traffic Pattern Analysis

Abstract:

Traffic Pattern Analysis project works towards enhancing urban mobility and minimizing congestion by leveraging artificial intelligence, real-time data analysis, and visualization techniques. In its last phase, the system combines machine learning models to examine live as well as historical traffic data, identify patterns, and predict congestion. IoT information from intelligent traffic sensors is used to provide realtime updates, and interactive dashboards are used to provide insights to authorities and commuters. Full system demonstration, technical documentation, performance metrics, codebase, and testing outputs are provided in this report. Scalable and highperformance design, the system provides actionable insights to city planners and transport departments. Screenshots, flow diagrams, and code snapshots are provided to represent the complete architecture.

1. Project Demonstration

Overview:

The Traffic Pattern Analysis system will be demonstrated to stakeholders, illustrating how it monitors and analyzes real-time traffic flow, congestion levels, and anomaly detection based on AI and IoT inputs.

Demonstration Details:

- **System Walkthrough:** Live demo of the traffic monitoring dashboard, map-based visualizations, and AI-driven analytics for pattern detection.
 - **AI Model Accuracy:** Demonstration of how traffic prediction models identify congestion patterns based on historical and real-time data.
 - **IoT Integration:** Real-time display of inputs from traffic sensors and cameras such as vehicle number, speed, and density.
 - **Performance Metrics:** Load capacity, dashboard refresh response times, and accuracy in traffic forecasting.
 - **Security & Data Privacy:** Concise explanation of how sensitive information (e.g., user location, license plate data) is anonymized and protected.
- The demo verifies the system's capabilities in practical traffic conditions, providing solid insights and aiding urban mobility planning.

2. Project Documentation

Overview:

Comprehensive documentation summarizing the architecture, models, tools, and procedures involved in building the Traffic Pattern Analysis platform.

Documentation Sections:

- **System Architecture:** Data flow, ML model pipelines, and visualization layer diagrams.
- **Code Documentation:** Description of modules such as real-time data intake, AI analysis, and API integration.
- **User Guide:** Guidelines for city planners and traffic control teams to use the dashboard.
- **Administrator Guide:** System maintenance protocols, such as retraining models and sensor input updates.
- **Testing Reports:** Documentation of accuracy thresholds, load testing, and integration testing with sensor APIs.

Outcome:

Everything about the system is well documented, enabling future maintenance, upgrades, or scale-up deployments in other cities.

3. Feedback and Final Adjustments**Overview:**

Feedback will be gathered from authorities, mentors, and beta users after the live demo to streamline the system.

Steps:

- **Feedback Collection:** Collected by surveys, interviews, and observation during demonstration.
- **System Refinement:** Refining dashboard UX, prediction accuracy, and response times through feedback.
- **Final Testing:** To ensure hassle-free performance after changes.

Outcome:

The system is optimized for deployment with increased stability, accuracy, and usability.

4. Final Project Report Submission Overview:

A comprehensive report encapsulating the phases of the project, challenges, solutions, outcomes, and recommendations.

Report Sections:

- **Executive Summary:** Project objectives and accomplishments overview.
- **Phase Breakdown:** Overview of data collection, model training, real-time analysis, and deployment.
- **Challenges & Solutions:** Problems such as inconsistent sensor data or model drift, with solution strategies.
- **Outcomes:** Emphasizes current system capabilities, scalability, and success measures.

Outcome:

An official report summarizing the entire lifecycle and effect of the Traffic Pattern Analysis system.

5.Handover of the Project and Future Development

Overview:

Final handover of the system and strategizing for longer functionalities.

Handover Details:

- **Next Steps:** Concepts for future enhancements like deployment of edge AI, integration with emergency services, and mobile applications for commuters.

Outcome:

The project is handed over with complete documentation, codebase, and a development roadmap for further enhancement and scalability.

Screenshots Code and Progress of the Project :

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, classification_report

# --- 1. Generate Simulated Traffic Data ---
np.random.seed(42)

data = {
    'vehicle_count': np.random.randint(10, 100, 200),
    'avg_speed': np.random.uniform(10, 70, 200),
    'time_of_day': np.random.randint(0, 24, 200)
}

df = pd.DataFrame(data)

# Define congestion: 0 = Low, 1 = High
df['congestion'] = np.where(
    (df['vehicle_count'] > 60) & (df['avg_speed'] < 30),
    1,
    0
)

# --- 2. Train a Logistic Regression Model ---
features = ['vehicle_count', 'avg_speed', 'time_of_day']
X = df[features]
y = df['congestion']

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

model = LogisticRegression()
model.fit(X_train, y_train)
```

```
# --- 3. Predict and Evaluate ---
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)

print("Model Accuracy:", accuracy)
print("\nClassification Report:\n", classification_report(y_test, y_pred))

# --- 4. Visualizations ---

# Graph 1: Scatter plot - vehicle count vs avg speed, colored by congestion
plt.figure(figsize=(8, 5))
scatter = plt.scatter(
    df['vehicle_count'], df['avg_speed'],
    c=df['congestion'], cmap='coolwarm'
)
plt.title("Traffic Data: Vehicle Count vs Avg Speed")
plt.xlabel("Vehicle Count")
plt.ylabel("Average Speed (km/h)")
plt.colorbar(scatter, label="Congestion Level (0=Low, 1=High)")
plt.grid(True)
plt.show()

# Graph 2: Line plot - average congestion by hour of day
hourly_avg = df.groupby('time_of_day')['congestion'].mean()

plt.figure(figsize=(8, 5))
plt.plot(hourly_avg.index, hourly_avg.values, marker='o', color='green')
plt.title("Average Congestion by Hour of Day")
plt.xlabel("Hour of Day")
plt.ylabel("Average Congestion Level")
plt.xticks(range(0, 24))
plt.grid(True)
plt.show()
```

```
# Graph 3: Bar chart - average vehicle count for low vs high congestion
avg_vehicle_by_congestion = df.groupby('congestion')['vehicle_count'].mean()

plt.figure(figsize=(6, 5))
avg_vehicle_by_congestion.plot(kind='bar', color=['blue', 'red'])
plt.title("Average Vehicle Count by Congestion Level")
plt.xticks(ticks=[0, 1], labels=['Low Congestion', 'High Congestion'], rotation=0)
plt.ylabel("Average Vehicle Count")
plt.grid(axis='y')
plt.show()
```

OUTPUT :





