MODEL PROJECT

FOR

CPS

TRAFFIC VOLUME ANALYSIS

BY

R.S.JAYAKUMAR

1. Introduction

* Traffic congestion is a significant challenge in urban areas, affecting productivity, fuel consumption, and environmental sustainability.
* Traffic pattern analysis helps identify peak hours, understand congestion trends, and optimize transportation systems.
* This project simulates traffic data and analyzes flow and congestion patterns across hours and days, incorporating weather conditions to examine their influence on traffic density.

2. Problem Statement

* Traffic congestion leads to delays, fuel wastage, and increased pollution in cities.
* Understanding traffic flow patterns is essential for optimizing transportation infrastructure.
* Weather conditions like rainfall and temperature fluctuations further impact traffic density.
* The lack of accessible tools for analyzing traffic patterns limits data-driven decisions.
* This project addresses these challenges by providing insights into traffic flow trends and their relationship with weather.

3. Objectives

* Simulate and analyze traffic volume data for a city across hours and days.
* Identify peak traffic hours and days with the highest congestion.
* Explore the correlation between traffic volume and weather conditions (e.g., temperature, rainfall).
* Visualize traffic density using heatmaps, scatter plots, and line graphs.
* Provide actionable insights to optimize traffic management and resource allocation.
* Create a reusable Python-based framework for traffic analysis.

4. Python Libraries Used in the Project

NumPy:

* Handles numerical operations such as random data generation and statistical analysis.
* Pandas:
* Manages tabular data with easy-to-use functions for grouping, filtering, and aggregation.

Seaborn:

* Visualizes data trends and patterns through advanced charts like heatmaps and scatter plots.
* Matplotlib:
* Customizes plots such as line graphs and bar charts for clear representation.

SciPy:

* Computes correlations between traffic volume and weather conditions using statistical tests.
* Random (Optional):
* Generates synthetic datasets for traffic and weather conditions.

5. Modules of the Project

Data Simulation:

* Generates synthetic traffic volume, temperature, and rainfall data for 24 hours across 7 days.
* Ensures realistic randomness in values for analysis.

Statistical Analysis:

* Calculates hourly and daily averages of traffic volume.
* Identifies correlations between weather conditions (temperature, rainfall) and traffic congestion.

Visualization:

* Line plots for hourly traffic trends.
* Bar plots for daily averages.
* Heatmaps for traffic density by hour and day.
* Scatter plots to examine relationships between traffic volume and weather conditions.

Insights:

* Extracts actionable insights like peak traffic hours and weather-induced congestion.

7. Output Screenshots

* Line Plot (Hourly Average Traffic Volume):
* Visualizes traffic volume peaks during rush hours (e.g., 8 AM and 6 PM).
* Bar Plot (Daily Average Traffic Volume):
* Highlights higher traffic on weekdays compared to weekends.

Heatmap:

* Identifies congestion hotspots by hour and day.
* Scatter Plot (Traffic Volume vs Temperature):
* Shows how traffic volume changes with temperature and rainfall.

8. Applications of the Project

Traffic Management:

* Optimizes signal timing based on peak hours.

Public Transport Scheduling:

* Allocates buses and trains during high-demand periods.
* Weather-Responsive Planning:
* Implements strategies for rainy or hot days to reduce congestion.

Event Planning:

* Schedules events during low-traffic times to minimize disruptions.

Delivery Optimization:

* Suggests ideal delivery times to avoid traffic delays.

Infrastructure Planning:

* Identifies areas requiring additional lanes or alternate routes.

Environmental Impact:

* Analyzes the effects of traffic on emissions.

Emergency Services:

* Helps route ambulances and fire trucks efficiently.

Smart City Projects:

* Integrates into IoT systems for real-time traffic monitoring.

Education:

* Provides a learning tool for students in data analytics and urban planning.

9. Limitations of the Project

Synthetic Data:

* Relies on simulated data, which may not fully represent real-world traffic patterns.

Limited Scope:

* Focuses on hourly and daily trends without real-time updates.

Weather Impact:

* Only considers temperature and rainfall; ignores wind, fog, etc.

No Road-Specific Analysis:

* Does not analyze traffic flow on specific roads or intersections.

Static Analysis:

* Does not adapt dynamically to changing data.

Scalability:

* May require optimization for larger datasets.
* Correlation, Not Causation:
* Correlations do not imply direct relationships between variables.

No Predictive Modeling:

* Does not forecast future traffic trends.
* Visualization Limitations:
* Additional charts (e.g., stacked bar plots) could enhance insights.

No User Interaction:

* Lacks a user interface for custom input or exploration.

10. Bibliography

* NumPy Documentation: https://numpy.org/doc/
* Pandas Documentation: https://pandas.pydata.org/docs/
* Seaborn Documentation: https://seaborn.pydata.org/
* Matplotlib Documentation: https://matplotlib.org/stable/contents.html
* SciPy Documentation: https://scipy.org/
* Research Paper: Traffic Congestion Analysis - IEEE Journals.
* City Traffic Trends Reports - Local Government Portals.
* Data Science in Transportation - O’Reilly Media.
* Real-Time Traffic Monitoring - ResearchGate.
* Python Visualization Techniques - Packt Publishing.

## CODING

import numpy as np

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

# Step 1: Simulate Traffic Data

np.random.seed(42) # For reproducibility

# Generate synthetic data for traffic patterns

hours = np.arange(0, 24) # 24 hours in a day

days = ["Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"]

# Traffic volume varies across hours and days

traffic\_data = {

"Hour": np.tile(hours, len(days)),

"Day": np.repeat(days, len(hours)),

"Traffic Volume": np.random.randint(50, 500, len(hours) \* len(days)) # Random traffic volume

}

# Create a DataFrame

traffic\_df = pd.DataFrame(traffic\_data)

# Step 2: Statistical Analysis

# Calculate average traffic per hour

hourly\_avg = traffic\_df.groupby("Hour")["Traffic Volume"].mean()

# Calculate average traffic per day

daily\_avg = traffic\_df.groupby("Day")["Traffic Volume"].mean()

# Step 3: Visualizations

# a. Hourly Traffic Pattern

plt.figure(figsize=(12, 6))

sns.lineplot(x=hourly\_avg.index, y=hourly\_avg.values, marker="o", color="blue")

plt.title("Hourly Average Traffic Volume", fontsize=16)

plt.xlabel("Hour of the Day", fontsize=12)

plt.ylabel("Average Traffic Volume", fontsize=12)

plt.grid()

plt.xticks(range(0, 24, 1))

plt.show()

# b. Daily Traffic Pattern

plt.figure(figsize=(10, 6))

sns.barplot(x=daily\_avg.index, y=daily\_avg.values, palette="viridis")

plt.title("Daily Average Traffic Volume", fontsize=16)

plt.xlabel("Day of the Week", fontsize=12)

plt.ylabel("Average Traffic Volume", fontsize=12)

plt.show()

# c. Heatmap of Traffic Volume by Hour and Day

# Fix: Calculate average traffic volume for each Hour-Day combination

traffic\_pivot = traffic\_df.groupby(["Hour", "Day"])["Traffic Volume"].mean().unstack()

plt.figure(figsize=(12, 8))

sns.heatmap(traffic\_pivot, annot=False, cmap="coolwarm", cbar=True)

plt.title("Traffic Volume Heatmap (Hour vs. Day)", fontsize=16)

plt.xlabel("Day of the Week", fontsize=12)

plt.ylabel("Hour of the Day", fontsize=12)

plt.show()

# Step 4: Insights

print("Hourly Average Traffic Volume:")

print(hourly\_avg)

print("\nDaily Average Traffic Volume:")

print(daily\_avg)

A graph with blue lines

Description automatically generated

<ipython-input-1-c707466d22d8>:44: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

sns.barplot(x=daily\_avg.index, y=daily\_avg.values, palette="viridis")

A graph of a number of traffic

Description automatically generated with medium confidenceA diagram of a heatmap

Description automatically generated

Hourly Average Traffic Volume:

Hour

0 263.285714

1 254.857143

2 317.428571

3 239.142857

4 257.571429

5 278.285714

6 254.285714

7 344.000000

8 181.142857

9 331.714286

10 240.428571

11 284.285714

12 193.571429

13 298.000000

14 250.714286

15 280.428571

16 326.714286

17 215.571429

18 409.142857

19 246.000000

20 307.428571

21 341.428571

22 259.428571

23 306.714286

Name: Traffic Volume, dtype: float64

Daily Average Traffic Volume:

Day

Friday 277.541667

Monday 269.500000

Saturday 251.791667

Sunday 285.250000

Thursday 286.625000

Tuesday 284.750000

Wednesday 293.333333

Name: Traffic Volume, dtype: float64

GITHUB LINK

<https://github.com/jk08-maximus/TRAFFIC-VOLUME-ANALYSIS.git>