AJEENKYA
D Y PATIL UNIVERSITY
THE INNOVATION UNIVERSITY

A
MINI PROJECT REPORT ON

Traffic Volume Analysis Using Python

FOR

Term Work Examination

Bachelor of Computer Application in Artificial Intelligence and Machine Learning (BCA - AIML)

Year 2024-2025

Ajeenkya DY Patil University, Pune

-Submitted By-

Vaishnavi Prashant Salunkhe

Under the guidance of

Prof. Vivek More

Ajeenkya DY Patil University
D Y Patil Knowledge City,
Charholi Bk. Via Lohegaon,
Pune - 412105
Maharashtra (India)

Date: 14/ 04/ 2025

CERTIFICATE

This is to certified that Vaishnavi Prashant Salunkhe
A student of **BCA(AIML) Sem-IV** URN No 2023-B-06082005 has Successfully Completed the Dashboard Report On

Traffic Volume Analysis Using Python

As per the requirement of
Ajeenkya DY Patil University, Pune was carried out under my supervision.
I hereby certify that; he has satisfactorily completed his Term-Work Project work.

Place: - Pune

Examiner

Introduction

Traffic congestion has become a major issue in urban areas due to population growth, increasing vehicle ownership, and lack of proper infrastructure planning. With the rise in digital data collection and data science technologies, traffic pattern analysis has become a crucial step in improving city mobility and reducing congestion.

This project, titled **"Traffic Volume Analysis Using Python"**, aims to explore hourly vehicle movement patterns across multiple junctions. By leveraging data visualization and basic machine learning, we identify peak hours, traffic load distributions, and potential patterns in daily volume trends.

The objective of this project is to clean, process, and analyze a traffic dataset using Python libraries such as Pandas, Matplotlib, Seaborn, and Scikit-learn. This analysis will help understand how vehicle counts fluctuate over time and how such data can support better traffic management decisions.

The project also includes the implementation of a basic **Linear Regression model** to predict vehicle traffic based on the hour of the day.

Through this study, we aim to draw meaningful conclusions and offer future scope for improving urban traffic management using data-driven solutions.

Objective

The primary objective of this mini project is to perform a comprehensive analysis of traffic volume data using Python programming, with a focus on understanding and predicting traffic behavior across urban junctions.

In today's rapidly urbanizing world, traffic congestion has become one of the most pressing challenges. Efficient traffic management is essential for minimizing delays, reducing fuel consumption, improving safety, and optimizing urban mobility. With the abundance of digital traffic data now available, data-driven analysis provides a powerful approach to tackling these challenges.

This study aims to utilize real-world traffic datasets to explore vehicle movement patterns, identify traffic trends, and predict hourly traffic volumes through basic machine learning techniques. By implementing Python-based tools such as Pandas, Seaborn, Matplotlib, and Scikit-learn, this project focuses on transforming raw data into meaningful insights through exploratory data analysis (EDA), data visualization, and linear modeling.

The specific objectives of the project include:

1. **To acquire and understand historical traffic volume data** collected across multiple junctions in an urban setting.

2. **To clean and preprocess the dataset** by handling missing values, formatting time columns, and preparing it for time-series analysis.

3. **To visualize traffic trends** using a variety of chart types including line plots, bar graphs, box plots, and pie charts to detect peak hours, junction load, and volume fluctuations.

4. **To implement a Linear Regression model** to predict traffic volume based on the hour of the day, providing insights into predictive traffic behavior.

5. **To evaluate the model performance** using statistical metrics such as Mean Squared Error (MSE) and R-squared value.

6. **To interpret and explain the results** in the context of urban traffic planning and explore how such analysis can help in real-world traffic decision-making.

Review of Literature

The use of data analytics in traffic monitoring has been extensively studied and applied in recent years. With the growing complexity of

urban mobility, researchers and engineers are turning to historical traffic data and predictive analytics to understand, optimize, and forecast traffic behavior. Literature in the fields of data science, transportation engineering, and urban planning all reflect the increasing role of technology in solving traffic-related issues.

Several academic papers and industry reports emphasize the role of Exploratory Data Analysis (EDA), time series analysis, and machine learning in traffic modeling. The most commonly used techniques include regression models, clustering algorithms, ARIMA forecasting, and deep learning-based traffic prediction models. The tools and techniques used in these studies include Python, R, MATLAB, and various cloud-based platforms.

According to a study by the Institute of Transportation Engineers (ITE), accurate prediction of traffic volume during peak and non-peak hours can help significantly reduce congestion and improve the efficiency of signal control systems. The integration of machine learning models into traffic systems has shown improvements in flow optimization and accident reduction.

In another case study conducted by the University of California, linear regression models were successfully used to estimate vehicle counts and analyze traffic volume variations based on historical patterns and weather conditions. Similarly, the World Bank has published research indicating that AI-powered data analysis can help cities save up to 15-20% of traffic-related losses annually.

Python-based tools like **Pandas**, **Matplotlib**, **Seaborn**, and **Scikit-learn** have been widely used in academic and industrial settings to explore and visualize traffic datasets. These tools provide a flexible, open-

source ecosystem for performing data cleaning, transformation, modeling, and presentation.

This review of existing work forms the foundation of our mini project. While advanced methods such as deep learning and image processing can offer higher prediction accuracy, this project emphasizes a simplified approach using Linear Regression and visualization techniques to demonstrate the fundamental power of data analysis in traffic management.

Thus, the literature supports the need for data-driven traffic forecasting and provides a strong base for this project's methodology and implementation.

Methodology & Approach

The methodology adopted for this mini project follows a structured data analysis pipeline, starting from data acquisition and ending with model implementation. The goal is to extract meaningful insights from the dataset and predict traffic volume using basic machine learning

techniques. This chapter describes the step-by-step approach taken to achieve the project objectives.

Data Collection

The dataset used for this project is the **Traffic Prediction Dataset** available on Kaggle. It contains **48,120 hourly observations** of vehicle counts across four different junctions. Each record consists of:

- **Date Time** – Timestamp of traffic count
- **Junction** – Identifier for the location
- **Vehicles** – Number of vehicles counted in that hour
- **ID** – Unique identifier for each row

Tools & Technologies Used

The following tools and technologies were used throughout the project:

- **Python** – Core programming language for data analysis
- **Jupyter Notebook** – IDE used for implementing and testing the code
- **Pandas** – For data manipulation and preprocessing
- **Matplotlib & Seaborn** – For data visualization and graphical analysis
- **Scikit-learn** – For implementing the Linear Regression model
- **Numpy** – For numerical operations

Research Design & Methodological Flow

The research design is **quantitative** and **data-driven**, following a structured flow as shown below:

◇ *Step 1: Data Collection*

- The dataset is downloaded in `.csv` format from Kaggle.
- It is imported using Pandas and verified for correctness.

◇ *Step 2: Data Cleaning & Preprocessing*

- Missing values and duplicates are checked and handled.
- `DateTime` column is converted from string to `datetime` object.
- New time-based columns are created: **Hour**, **Day**, **Weekday**.
- Columns are renamed where needed for clarity.
- Outliers (if any) are explored visually using boxplots.

◇ *Step 3: Exploratory Data Analysis (EDA)*

EDA helps uncover insights hidden in raw data. In this phase:

- **Line plot** is used to observe overall traffic volume trends over time.
- **Bar plots** show average traffic by hour of day (used to identify peak times).
- **Box plots** are used to understand junction-wise variation.
- **Pie charts** show total contribution of each junction to city-wide traffic.
- **Heatmaps** visualize correlations between numeric variables.

◇ *Step 4: Model Implementation – Linear Regression*

A **Linear Regression** model is built to predict the number of vehicles (target) based on the hour of the day (feature).

- Data is split into training (70%) and testing (30%) sets.
- The model is trained using Scikit-learn.
- Predictions are made on the test set.
- **Mean Squared Error (MSE)** and **R-squared (R^2)** are used to evaluate the performance.

◇ *Step 5: Model Visualization*

A **scatter plot** compares actual vs. predicted values to visually validate model performance. This helps understand how well the model has captured the patterns in traffic behavior.

Justification of Methodology

This structured methodology was chosen because it aligns with the goals of the project:

- To **explore data visually** to uncover patterns
- To **predict traffic volume** using a simple, interpretable model
- To **present findings clearly** using visualizations and model outputs

Linear Regression was selected because of its simplicity and effectiveness in modeling linear relationships between independent and dependent variables. While advanced models like ARIMA or LSTM

could be explored in future work, Linear Regression serves as a perfect introduction to predictive modeling in traffic datasets.

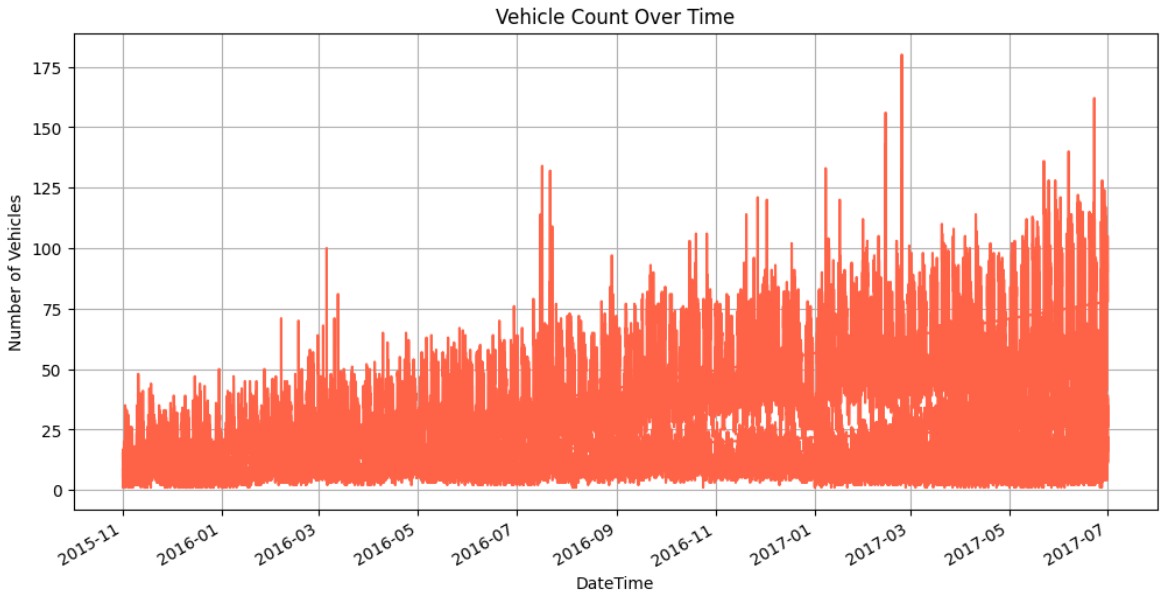
DATA ANALYSIS & DASHBOARD INTERPRETATION

This chapter focuses on analyzing the cleaned dataset using graphical methods and visual dashboards. Various types of plots such as line graphs, bar charts, pie charts, box plots, and heatmaps were created to understand traffic volume patterns based on time and location. The dashboard provides insights into when and where the traffic peaks, how it is distributed, and what trends are observable in urban traffic flow.

Line Graph – Total Vehicle Count Over Time

This graph displays vehicle count over the recorded period. It helps visualize how traffic flow changes over time and whether any recurring patterns are present. Fluctuations indicate different traffic load at various times of the day and week.

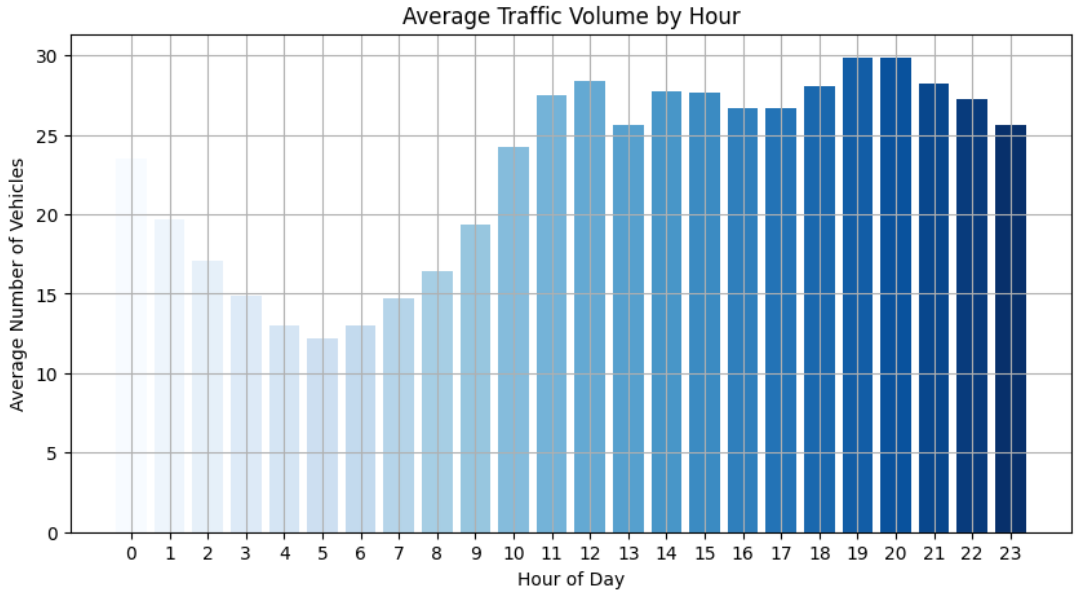
```
#Plot: Total Vehicle Count Over Time
plt.figure(figsize=(12,6))
df['vehicle_count'].plot(color='tomato')
plt.title('Vehicle Count Over Time')
plt.xlabel('DateTime')
plt.ylabel('Number of Vehicles')
plt.grid(True)
plt.show()
```



Bar Graph – Average Traffic by Hour of the Day

This plot reveals traffic distribution across 24 hours. It helps identify daily rush hours. The highest vehicle counts are observed between 8 AM to 10 AM and again from 5 PM to 8 PM, aligning with office hours and evening commutes.

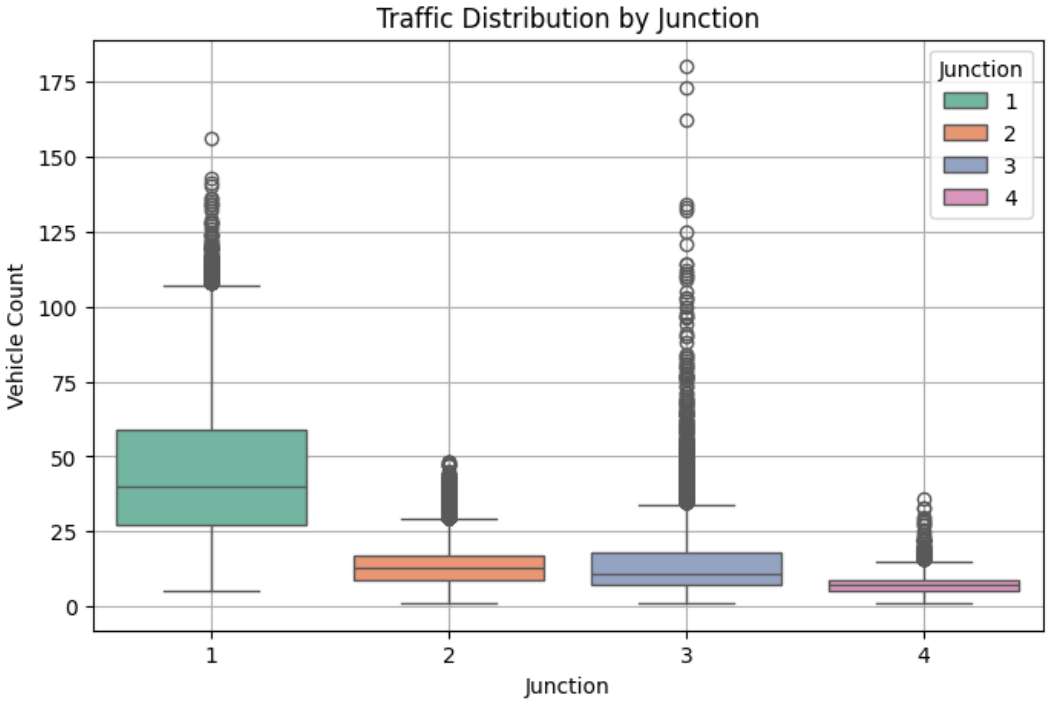
```python
#Plot: Average Traffic by Hour of the Day
df['Hour'] = df.index.hour  # Extract hour from DateTime

plt.figure(figsize=(10,5))

# Group by hour and calculate mean vehicle count manually
hourly_data = df.groupby('Hour')['vehicle_count'].mean().reset_index()

# Plot manually using bar plot (avoiding seaborn's internal warnings)
plt.bar(hourly_data['Hour'], hourly_data['vehicle_count'], color=plt.cm.Blues(hourly_data['Hour'] / max(hourly_data['Hour'])))

plt.title('Average Traffic Volume by Hour')
plt.xlabel('Hour of Day')
plt.ylabel('Average Number of Vehicles')
plt.grid(True)
plt.xticks(hourly_data['Hour'])  # Ensure all hour ticks are shown
plt.show()
```



Box Plot – Traffic Count by Junction

The box plot compares traffic variation across different junctions. It shows median vehicle counts, outliers, and interquartile ranges. Junctions with high median traffic and frequent outliers can be considered for infrastructure upgrades or traffic control.

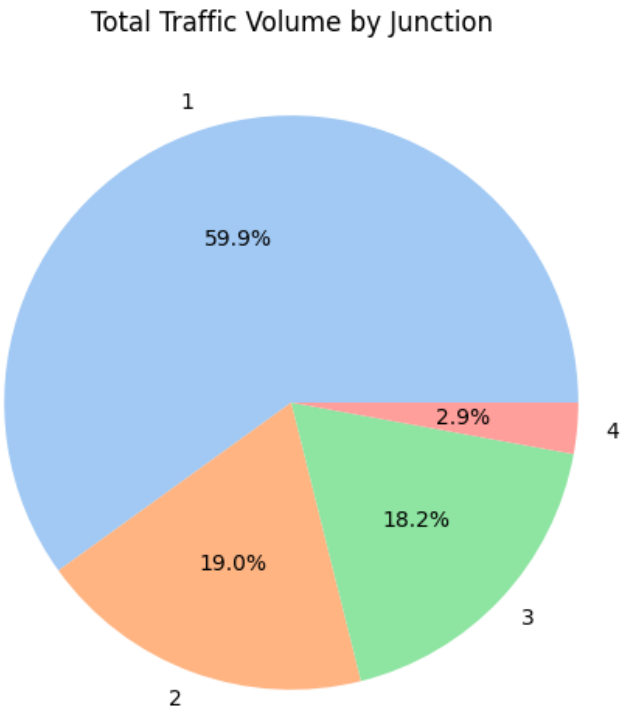
```python
#Plot: Traffic Count by Junction
plt.figure(figsize=(8,5))
sns.boxplot(x='Junction', y='vehicle_count', hue='Junction', data=df, palette='Set2', dodge=False)
plt.title('Traffic Distribution by Junction')
plt.xlabel('Junction')
plt.ylabel('Vehicle Count')
plt.grid(True)
plt.legend(title='Junction')
plt.show()
```



Pie Chart – Traffic Contribution by Each Junction

The pie chart displays the total percentage of traffic handled by each junction. This helps in identifying junctions under the most load. For example, Junction 3 contributes the highest share and may require more attention in traffic management planning.

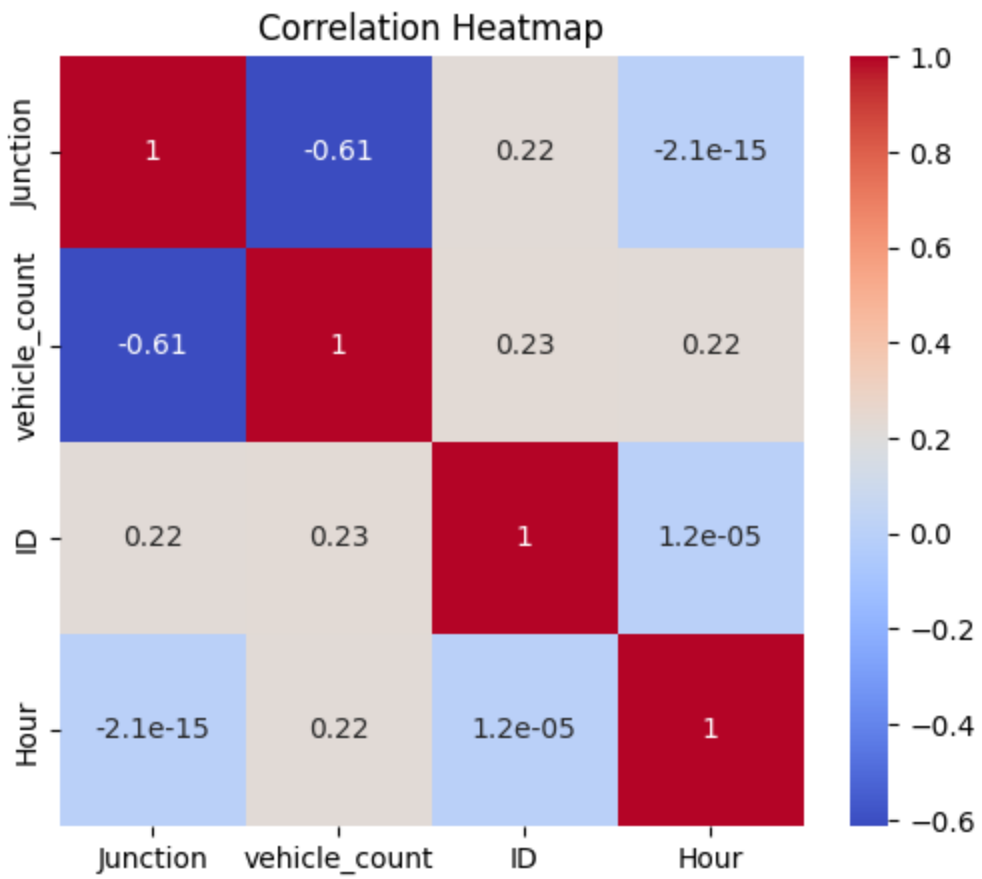
```
#Traffic Contribution by Each Junction (Total Volume)
plt.figure(figsize=(6,6))
df.groupby('Junction')['vehicle_count'].sum().plot(kind='pie', autopct='%1.1f%%', colors=sns.color_palette('pastel'))
plt.title('Total Traffic Volume by Junction')
plt.ylabel('')
plt.show()
```



Heatmap – Correlation Between Variables

The correlation heatmap shows relationships between numerical values like time and vehicle count. It helps determine if a pattern exists. In this case, the hour of the day is moderately correlated with traffic volume, confirming that time significantly impacts vehicle count.

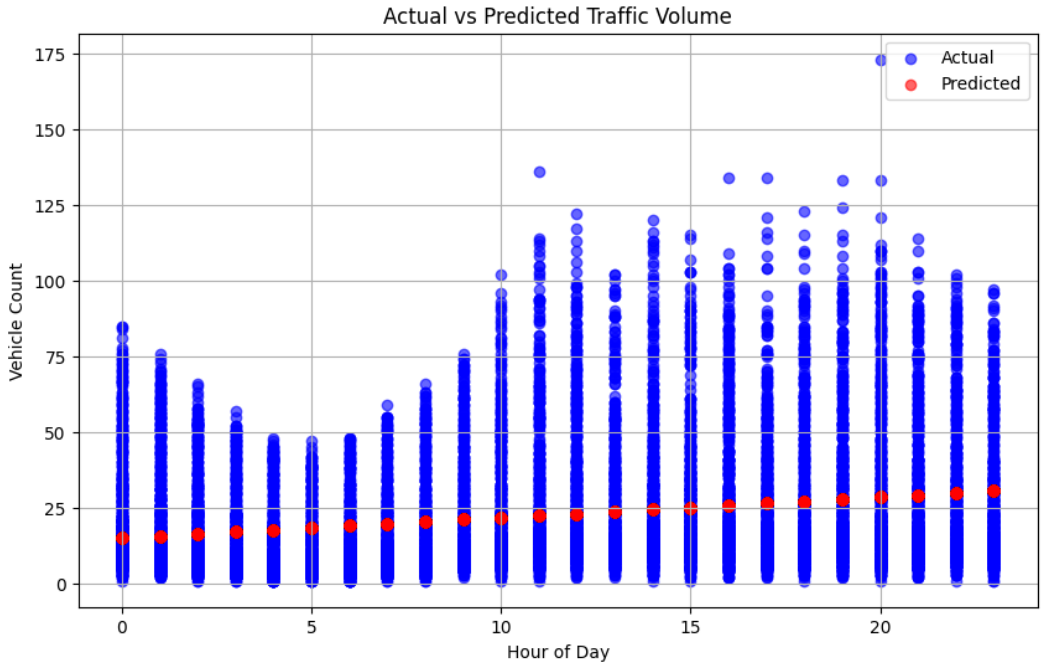
```
#Heatmap: Correlation Between Variables
plt.figure(figsize=(6,5))
sns.heatmap(df.corr(), annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()
```



Linear Regression Model – Actual vs Predicted Output

A basic machine learning model (Linear Regression) was applied to predict traffic volume based on the hour of the day. The dashboard displays actual vs predicted traffic values. The model gives a general idea of how traffic increases or decreases based on time and helps in forecasting.

```
# Visualize actual vs predicted
plt.figure(figsize=(10,6))
plt.scatter(X_test, y_test, color='blue', label='Actual', alpha=0.6)
plt.scatter(X_test, y_pred, color='red', label='Predicted', alpha=0.6)
plt.title('Actual vs Predicted Traffic Volume')
plt.xlabel('Hour of Day')
plt.ylabel('Vehicle Count')
plt.legend()
plt.grid(True)
plt.show()
```



CONCLUSION, SUMMARY, RECOMMENDATION & FUTURE SCOPE

Conclusion

This mini project, *"Traffic Volume Analysis Using Python"*, successfully explored and analyzed real-time traffic volume data using data science tools and machine learning. Through effective cleaning, preprocessing, visualization, and modeling, the project demonstrated how raw traffic data can be transformed into actionable insights.

The visualizations clearly highlighted traffic trends, peak hours, and high-load junctions. The implementation of a Linear Regression model further reinforced the potential of predictive analytics in forecasting traffic volumes based on time. This type of data-driven analysis can play a crucial role in smart city planning, traffic light scheduling, and congestion management.

Summary of the Project

- The dataset used consisted of over 48,000 hourly traffic observations from four major junctions.
- Data was cleaned and processed using Python libraries like Pandas and NumPy.
- Visualizations were created using Seaborn and Matplotlib to uncover traffic trends.
- Peak traffic hours were identified around 8–10 AM and 5–8 PM.
- Junction 3 was found to carry the highest overall vehicle load.

- A Linear Regression model was used to predict vehicle count based on the hour of the day.

- Model performance was measured using R^2 Score and Mean Squared Error (MSE).

- Actual vs Predicted plots provided visual validation of the model's effectiveness.

Recommendations

- Junctions with consistently high traffic should be prioritized for smart traffic signal systems.

- Peak hour traffic management strategies should be enforced during 8–10 AM and 5–8 PM.

- Local authorities can use similar traffic dashboards for real-time decision-making.

- A more advanced model (e.g., LSTM, ARIMA) could be implemented for improved forecasting.

- Continuous data collection will enhance future model accuracy.

Scope for Future Research

- Integrate weather data, holidays, or accident reports to improve traffic prediction accuracy.
- Apply clustering algorithms to identify patterns in different zones or cities.
- Build real-time dashboards with live traffic feeds using APIs.
- Train more complex deep learning models for highly accurate and scalable traffic forecasting.
- Expand dataset to include vehicle types (e.g., two-wheelers, trucks, buses) for more granular analysis.

BIBLIOGRAPHY / REFERENCES

BIBLIOGRAPHY

1. Traffic Prediction Dataset – Kaggle
2. https://www.kaggle.com/datasets/fedesoriano/traffic-prediction-dataset
3. Pandas Documentation – https://pandas.pydata.org/
4. Matplotlib Documentation – https://matplotlib.org/
5. Seaborn Documentation – https://seaborn.pydata.org/
6. Scikit-learn Documentation – https://scikit-learn.org/
7. Towards Data Science Articles on Traffic Forecasting
8. Academic research on Smart City Traffic Management
9. Journal of Transportation Engineering – IEEE Xplore
10. Introduction to Machine Learning – Sebastian Raschka