📘 Final Project Report

Title: *Predictive Modelling for Road Traffic Management: A Data-Driven Approach*

Presented by: Shaik Karishma

1. Abstract

Urban traffic congestion remains one of the most critical challenges in modern transportation systems, impacting travel time, fuel efficiency, and air quality. This project introduces a data-driven approach to traffic volume forecasting by leveraging historical traffic data and advanced predictive modeling techniques. The study implements three distinct models — ARIMA, LSTM, and XGBoost — to identify trends and predict future traffic volume accurately. The findings demonstrate how intelligent forecasting can enable authorities and planners to optimize road usage, reduce congestion, and make informed infrastructure decisions.

---

2. Objectives

- To forecast hourly traffic volume using machine learning and time-series analysis.
- To assist urban traffic authorities in proactively managing congestion and resource deployment.
- To evaluate and compare multiple predictive models for accuracy and performance.
- To visualize and interpret the forecasting results using statistical and graphical methods.

---

3. Dataset Description

- **Filename:** Sample_Traffic_Data.csv
- **Scope:** Simulated traffic volume data collected over 60 days at hourly intervals.
- **Features:**
  - date_time: Timestamp for each traffic observation.
  - traffic_volume: Number of vehicles observed per hour.

This dataset serves as a reliable proxy for real-world data, allowing the evaluation of different modeling strategies in a controlled environment.

---

4. Methodology

Step 1: Data Preprocessing

- Handled missing or duplicate entries.
- Converted datetime fields to a uniform format.
- Resampled and normalized data where necessary.

Step 2: Feature Engineering

- Extracted features such as hour of day and day of week to capture time-related patterns.

Step 3: Model Development

- **ARIMA:** A statistical model designed to capture linear time-series patterns.
- **LSTM:** A deep learning architecture capable of learning from sequential dependencies and long-term temporal patterns.
- **XGBoost:** A high-performance ensemble learning algorithm optimized for tabular datasets with engineered features.

Step 4: Model Evaluation

- Metrics used:
  - **MAE** (Mean Absolute Error)
  - **RMSE** (Root Mean Squared Error)
  - **R^2** (Coefficient of Determination)

Step 5: Visualization

- Time-series plots
- Actual vs predicted comparisons
- Error distribution graphs

---

5. Tools and Technologies Used

- **Programming Language:** Python
- **Development Environment:** Jupyter Notebook
- **Libraries & Frameworks:**

- Data Handling: pandas, numpy
- Visualization: matplotlib, seaborn
- Modeling: statsmodels (ARIMA), tensorflow (LSTM), xgboost, scikit-learn

---

6. Experimental Results

Model	MAE	RMSE	R^2 Score	Remarks
ARIMA	~320	Moderate	Fair	Simple model, useful for baseline results
LSTM	~230	Low	High	Best performance, captures nonlinearities
XGBoost	~250	Low	High	Fast, robust with engineered features

Each model provides different benefits. ARIMA is interpretable and lightweight; LSTM excels in learning time-based sequences; and XGBoost balances performance with training efficiency.

---

7. Visualization Insights

- **Time-Series Trends:** Clear temporal patterns were detected in daily traffic volumes.
- **Prediction Plots:** All three models provided close alignment with actual values.
- **Error Distribution:** LSTM showed the narrowest error band, indicating more consistent accuracy.

---

8. Conclusion

This project successfully demonstrates the application of data science techniques in traffic volume forecasting. The LSTM model outperforms traditional and tree-based approaches in predictive accuracy, though XGBoost offers faster performance and easier deployment. Results show that time-based features alone are strong indicators for traffic volume forecasting when processed with appropriate models.

---

9. Future Scope

- **Data Enrichment:** Incorporate external factors such as weather conditions, road incidents, and public holidays to improve forecasting accuracy.

- **Real-Time Deployment:** Build a web application (e.g., using Streamlit or Flask) for live forecasting and dashboard visualization.
- **Scalability:** Extend the system to handle city-wide or regional traffic networks for smart city integration.
- **Model Optimization:** Apply hyperparameter tuning, deep stacking, or hybrid models for enhanced performance.