FORECASTING SMART CITY TRAFFIC PATTERNS USING MACHINE LEARNING

PREPARED BY
PURVA SALGAONKAR

INTRODUCTION

With the rapid growth of urban populations, managing traffic has become one of the most pressing challenges for city administrations. Governments are working to transform cities into smart cities—digital and intelligent ecosystems that optimize urban infrastructure and services for citizens.

One key issue is traffic congestion, which impacts daily life, public safety, and environmental health. This project focuses on analyzing historical traffic data from four key city junctions to forecast traffic patterns. By understanding how traffic fluctuates during weekdays, weekends, holidays, and special events, the system helps authorities plan infrastructure, manage traffic flow efficiently, and prepare for traffic peaks.

ABOUT THE COMPANY

Upskill Cloud Technologies (UCT) is a training and development organization focused on empowering students and professionals with industry-relevant technical skills. UCT designs structured learning paths that combine theoretical understanding with practical, project-based experience. The company collaborates closely with industry experts to ensure that its programs reflect real market needs and evolving technological trends.

As part of its Machine Learning Internship Program, UCT provides a guided environment where learners work on real datasets, explore end-to-end problem-solving methods, and develop confidence in applying data science techniques. The program emphasizes weekly progress tracking, hands-on implementation, and continuous skill enhancement, enabling participants to gain practical exposure and prepare for real-world roles in analytics, AI, and data-driven decision-making.

DATASET DESCRIPTION

We are working with the government to develop smart cities and improve public services through intelligent systems. One major challenge is traffic management. The government wants a robust system that can predict traffic peaks and understand how traffic changes across four main junctions, especially on holidays and special occasions.

The dataset contains traffic records collected from four city junctions. It includes:

- Date of traffic recording
- Junction identification
- Number of vehicles passing through
- Holiday or event indicators

This data allows analysis of traffic trends, detection of peak hours, and differences in traffic patterns on working days, weekends, and special occasions. It forms the basis for training machine learning models to forecast future traffic levels.

Dataset Link:
https://drive.google.com/file/d/1y61cDyuO9Zrp1fSchWcAmCxk0B6SMx7X/view?usp=sharing

OBJECTIVES

- Analyze traffic patterns across multiple city junctions.
- Predict traffic counts to anticipate congestion and peak hours.
- Assist government authorities in infrastructure planning and traffic management.
- Identify differences in traffic during holidays, events, and normal working days.
- Provide insights for designing robust traffic control systems.

Tools and Technologies Used

- Python: Programming language for data handling and model building
- Pandas: Data preprocessing and manipulation
- NumPy: Numerical calculations
- Scikit-learn: Machine learning model building and evaluation
- Matplotlib & Seaborn: Visualization of traffic trends and predictions
- Google Colab: Cloud-based platform for running code and analysis

WORKING

- Data Extraction: The traffic data is extracted from ZIP files using Python. After extraction, the CSV files are loaded and prepared for analysis.

- Data Cleaning: Unwanted columns are removed, missing values are filled, and categorical fields (junction, holiday, event) are converted into numeric form. Date features like day, month, and year are also extracted.

- Feature Selection: The target variable (vehicles) and all relevant input features are selected to ensure the model learns from meaningful and useful data.

- Model Training: A Linear Regression model is trained to understand how traffic changes across junctions, on holidays, and on different days of the week.

- Prediction & Evaluation: The model predicts vehicle counts for new data, and these predictions are compared with actual values to measure accuracy.

- Visualization: Graphs comparing actual vs predicted traffic help planners clearly understand patterns and evaluate the model's performance.

CODE EXPLANATION

1. Import Libraries: Pandas, NumPy, Scikit-learn, Matplotlib, and Seaborn are imported to process data, build models, and visualize results.
2. Extract Dataset: The main ZIP file and any inner ZIP files are extracted using zipfile to access the CSV data.
3. Load CSV File: The code automatically searches for CSV files, selects one, and loads it using Pandas for preview.
4. Data Cleaning & Feature Engineering: Unnecessary columns are removed, column names are standardized, date columns are converted, and features like day, month, and year are extracted. Categorical fields (junction, holiday, event) are encoded using one-hot encoding.
5. Prepare Features (X) and Target (y): The target is the vehicle count, while all other relevant columns form the input features. Missing values are filled appropriately.
6. Train-Test Split: The dataset is divided into 80% training and 20% testing using train_test_split.
7. Train Model: A Linear Regression model is trained to learn how different features influence traffic counts.
8. Evaluate & Predict: Model accuracy is calculated, predictions are generated, and actual vs predicted values are compared.
9. Visualization: Matplotlib is used to plot actual and predicted traffic counts for easy interpretation.

RESULT

The machine learning model successfully analyzed the traffic dataset and learned the relationships between date-based features, junction locations, holiday indicators, and vehicle counts. After training the Linear Regression model, the system achieved strong predictive performance, showing that traffic patterns follow identifiable trends across different days and events.

The model demonstrated good accuracy when tested on unseen data, with predicted vehicle counts closely matching the actual values. The evaluation results showed that the model was able to capture variations such as higher traffic on weekdays, reduced flow on holidays, and peak hours at specific junctions. Visualizations comparing actual versus predicted values further confirmed the reliability of the model, helping traffic planners clearly understand how traffic behaves across different conditions. These results indicate that machine learning can effectively support smart city planning by providing dependable traffic forecasts.

FUTURE SCOPE

- Incorporate real-time traffic data from sensors or IoT devices.
- Use advanced machine learning models like Random Forest, XGBoost, or LSTM for better prediction accuracy.
- Include weather, roadwork, and accident data for more accurate forecasting.
- Develop a web or mobile application to provide live traffic predictions to citizens and authorities.

ACKNOWLEDGEMENT

I would like to thank the government traffic authorities and open-source Python community for providing datasets, tools, and libraries that made this project possible.

CONCLUSION

This project demonstrates how machine learning can be applied to forecast traffic patterns in a smart city. By analyzing historical data from four junctions and considering factors like holidays and events, the system can accurately predict traffic peaks and trends. This helps city authorities optimize traffic management, improve infrastructure planning, and ensure smooth traffic flow.