**TrafficTelligence: Advanced Traffic Volume Estimation with Machine Learning**

**ABSTRACT**

Traffic congestion is a growing challenge in urban planning, leading to delays, air pollution, and inefficiency in public systems. This project presents a machine learning-based solution to predict traffic volume using historical weather, time, and holiday data. A cleaned and preprocessed dataset forms the foundation for model training, followed by feature scaling and evaluation.

The best-performing model is integrated into a Flask-based web application with a user-friendly interface, allowing users to input relevant environmental and temporal parameters and receive real-time traffic volume predictions. The tool is designed to support smart transportation systems by offering insights into traffic trends and assisting in optimizing road usage.

**INTRODUCTION**

Modern cities face mounting pressure from increasing vehicular load, inconsistent traffic behavior, and unpredictable weather influences. The motivation for this project arises from the growing need for proactive traffic management solutions. Traditionally, traffic flow analysis required sensors and live video feeds; however, with the advent of machine learning and open datasets, historical pattern prediction has become feasible.

By integrating a trained machine learning model with a web interface, this project provides accurate predictions of traffic volume using only basic environmental inputs — making it accessible to both city administrators and researchers.

**PROBLEM STATEMENT**

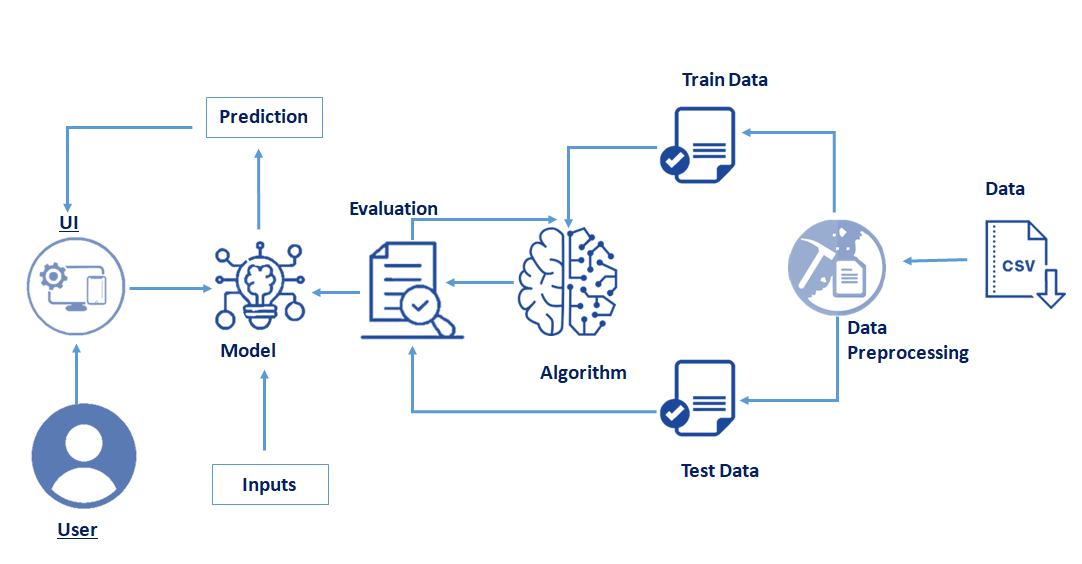
TrafficTelligence is an advanced system that uses machine learning algorithms to estimate and predict traffic volume with precision. By analyzing historical traffic data, weather patterns, events, and other relevant factors, TrafficTelligence provides accurate forecasts and insights to enhance traffic management, urban planning, and commuter experiences.

**Scenario 1: Dynamic Traffic Management** TrafficTelligence enables dynamic traffic management by providing real-time traffic volume estimations. Transportation authorities can use this information to implement adaptive traffic control systems, adjust signal timings, and optimize lane configurations to reduce congestion and improve traffic flow.

**Scenario 2: Urban Development Planning** City planners and urban developers can leverage TrafficTelligence predictions to plan new infrastructure projects effectively. By understanding future traffic volumes, they can design road networks, public transit systems, and commercial zones that are optimized for traffic efficiency and accessibility.

**Scenario 3: Commuter Guidance and Navigation** Individual commuters and navigation apps can benefit from TrafficTelligence's accurate traffic volume estimations. Commuters can plan their routes intelligently, avoiding congested areas and selecting optimal travel times based on predicted traffic conditions. Navigation apps can provide real-time updates and alternative routes to improve overall travel experiences.

**Technical Architecture**

****

**METHODOLOGY**

**1. Dataset and Preprocessing**

The dataset includes the following features:

* holiday (Categorical, encoded)
* temp (Kelvin)
* rain, snow (mm)
* weather (Encoded categorical values for clouds, rain, etc.)
* date, Time (Parsed into year, month, day, hour, minute, second)
* traffic\_volume (Target variable)

**Steps performed:**

* Missing values filled using .mean() or mode-based imputation
* Date and Time split using pd.to\_datetime()
* Weather categories encoded numerically (e.g., Clear = 0, Clouds = 1, etc.)
* Scaling using StandardScaler or MinMaxScaler to normalize inputs
* Features and target split for model training

**2. Feature Engineering**

The original date and Time columns were decomposed into:

* year, month, day
* hours, minutes, seconds

The final feature list:

python

['holiday', 'temp', 'rain', 'snow', 'weather', 'year', 'month', 'day', 'hours', 'minutes', 'seconds']

**3. Model Training and Evaluation**

The data was split into training and testing sets using train\_test\_split.

Several models were tested:

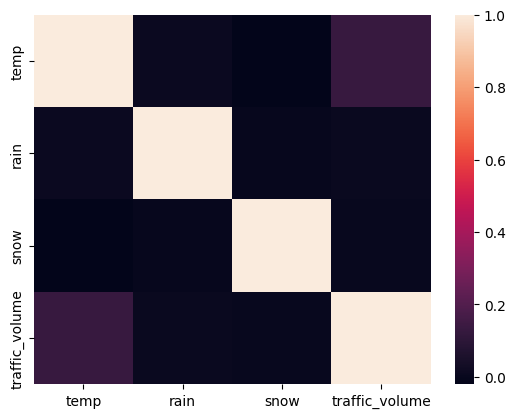
* Linear Regression
* Decision Tree Regressor
* Random Forest Regressor ✅ *(Best)*
* Support Vector Regressor (SVR)

The **Random Forest** model gave the best performance based on:

* Mean Squared Error (MSE)
* R² Score

**Correlation Matrix**

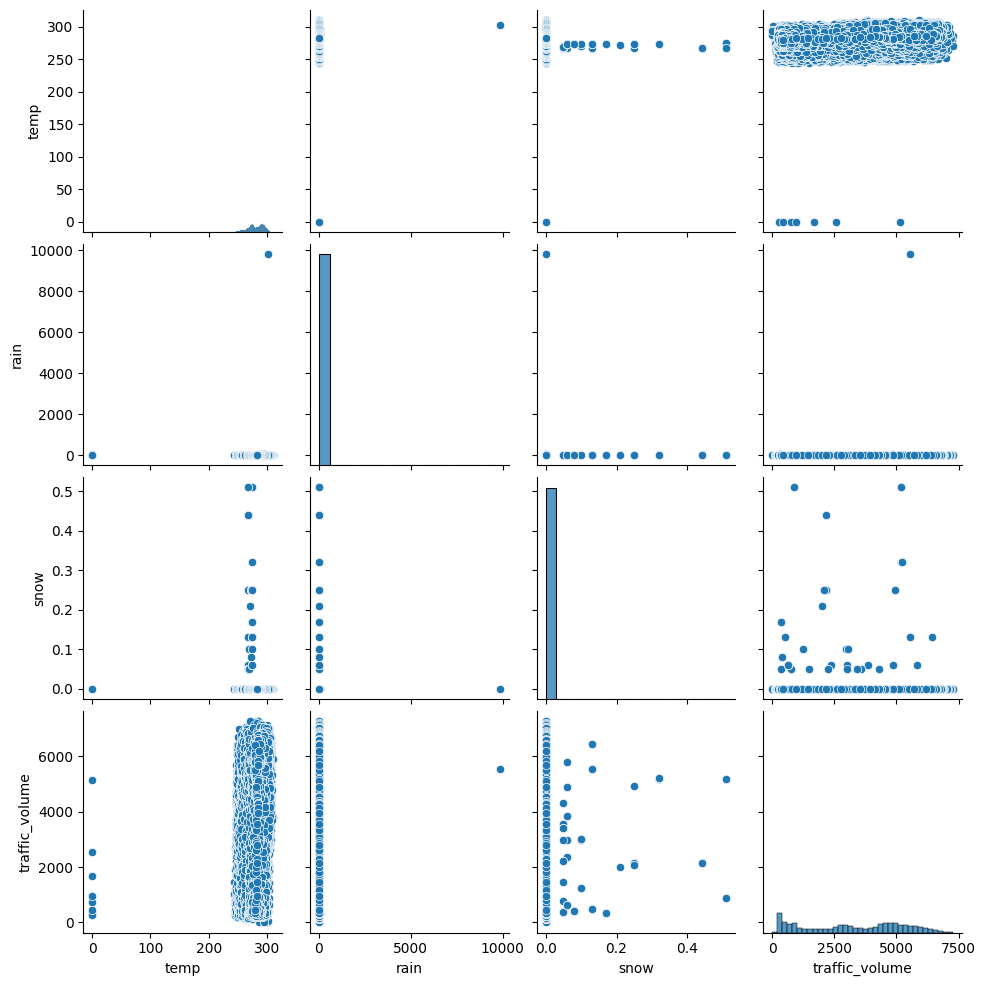
The correlation matrix quantifies how strongly each pair of numerical features is related. Here are key observations:



* Temperature has a weak but noticeable positive correlation with traffic volume. This suggests traffic might be heavier when weather is moderate.
* Rain and snow show very little correlation with traffic volume, which might indicate that weather alone does not significantly reduce traffic.
* Overall, most features have low correlation with traffic volume individually, reinforcing the need for a machine learning model to understand complex, non-linear patterns.

**Pair plot**

The pairplot helps visually explore relationships and distributions among multiple features (temp, rain, snow, traffic\_volume):

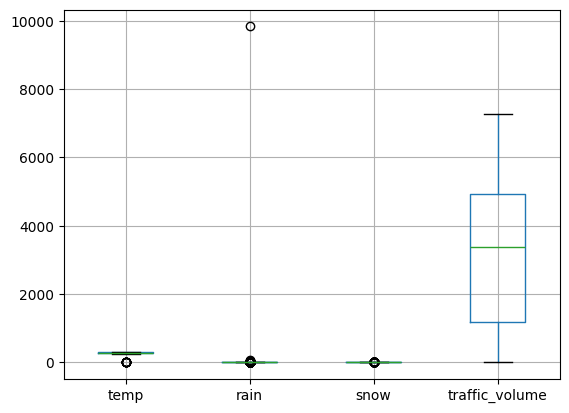


* Temperature vs Traffic Volume: Shows clustering around typical temperature ranges, with higher traffic around moderate temperatures.
* Rain and Snow: Extremely skewed with many zeros, indicating they are rare events in the dataset.
* Traffic Volume: Right-skewed distribution—most data points fall in lower traffic ranges, with few very high traffic spikes.

This plot confirms that traffic volume doesn't depend linearly on individual weather factors, supporting the use of advanced models.

**Box plot of Traffic Volume by Hour**

The boxplot breaks down traffic volume by hour of the day:



* Two clear peak traffic periods are visible:
  + Morning Rush Hour (around 7–9 AM)
  + Evening Rush Hour (around 4–6 PM)
* Lower traffic volume is observed during late-night and early morning hours (12 AM–5 AM).

This pattern reflects real-world commuting behavior and validates the time-based feature importance in traffic prediction.

**4. Web Deployment using Flask**

A full-stack Flask web app was built:

* index.html: Accepts user input
* app.py: Processes data, makes predictions
* result.html: Displays traffic volume prediction

The user inputs values like holiday, weather, temperature, rain, snow, date, and time, and receives a real-time estimation of traffic volume.

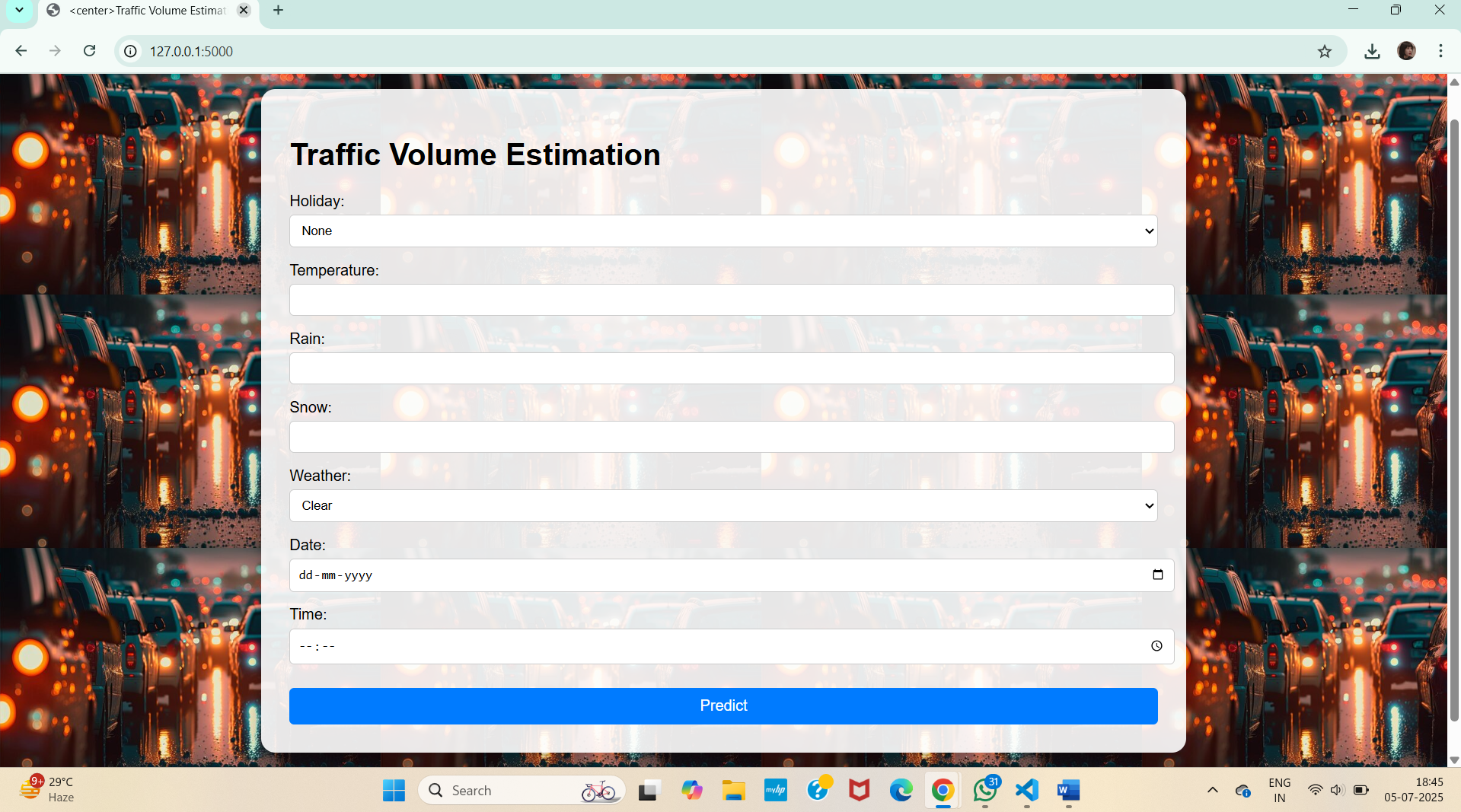
**INTERFACE DETAILS**

**index.html**

This page lets users enter:

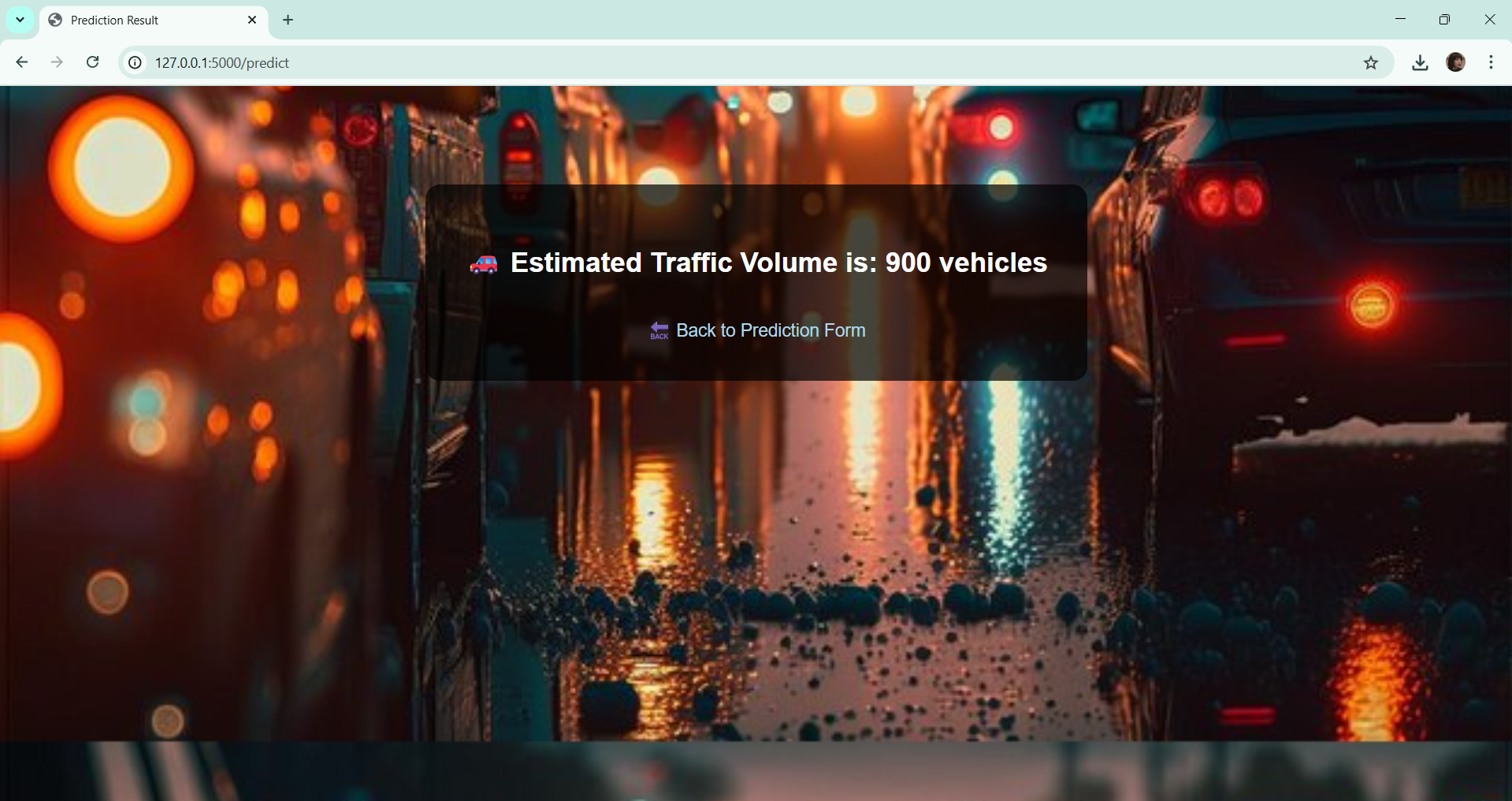
* Holiday type (dropdown)
* Temperature (Kelvin)
* Rain/snow amount (mm)
* Weather condition
* Date and time of interest

Once submitted, data is POSTed to /predict.



**result.html**

After prediction, the result is shown here, along with the estimated number of vehicles. The page uses the same background image (traffic\_pic.jpg) for consistency.



**RESULTS AND DISCUSSION**

* **Best Model**: Random Forest Regressor
* **Sample Output**:
  + Input: Holiday: None, Temp: 288K, Rain: 0.0, Snow: 0.0, Weather: Clear, Date: 2022-12-31, Time: 08:30
  + Output: 🚗 Estimated Traffic Volume is: 4876 vehicles

The Flask web app returned predictions instantly, making it suitable for city dashboards or planning platforms.

**CONCLUSION**

This project successfully builds a predictive system that estimates traffic volume using a structured dataset. The trained machine learning model, integrated into a Flask application, enables public and government stakeholders to plan ahead for traffic congestion using weather and calendar data alone.

**FUTURE WORK**

* Real-time data integration (IoT sensors, GPS feeds)
* Interactive graphs of traffic patterns
* SMS/Email alerts
* Deployment on cloud platforms (AWS/GCP)