Traffictelligence: Advanced Traffic Volume Estimation Using Machine Learning

1. **Introduction**

- **Project Title**: Traffictelligence: Advanced Traffic Volume Estimation Using Machine Learning
- **Team Members**:
    - Aparna Devi Baswa – Frontend Developer & Documentation
    - Geddada Sri Harika –Backend Developer
    - Andraju Navya Sri – Data Analyst
    - Abhinav Manikanta Yedida – Model Developer

2. **Project Overview**

- **Purpose**: To estimate traffic volume using machine learning models based on various input features such as time, weather, and holidays. The goal is to help optimize traffic flow and assist city planners.
- **Features**:
    - Predict traffic volume using input features.
    - User-friendly web interface for input.
    - Scalable backend using Flask and ML models.

3. **Architecture**

- **Frontend**:
    - Built using HTML, CSS, and Bootstrap.
    - Designed to collect user input in the correct feature order.
- **Backend**:
    - Developed using Flask (Python).
    - Handles form submission, model loading, and prediction.
- **Database**:
    - No active database integration, as prediction is based on real-time inputs.

4. **Setup Instructions**

- **Prerequisites**:
    - Python 3.x
    - Flask

- scikit-learn
- xgboost
- pandas, numpy

- **Installation**:
  1. Clone the repository.
  2. Install dependencies using pip install -r requirements.txt.
  3. Place model.pkl and scale.pkl in the root directory.
  4. Run python app.py to start the server.

5. **Folder Structure**

- **Client**:
  - templates/index.html – Input form UI.
  - templates/result.html – Result display page.
- **Server**:
  - app.py – Main backend file for Flask app.
  - model.pkl, scale.pkl – Saved ML model and scaler.

6. **Running the Application**

- **Frontend**: Handled through Flask templates.
- **Backend**:
  - Run python app.py
  - Access application at http://localhost:5000

7. **API Documentation**

- **POST /** /predict
  - **Parameters**: holiday, temp, rain, snow, weather, year, month, day, hours, minutes, seconds
  - **Response**: JSON with predicted traffic volume.

8. **Authentication**

- No user authentication implemented. Future versions can include login/signup features.

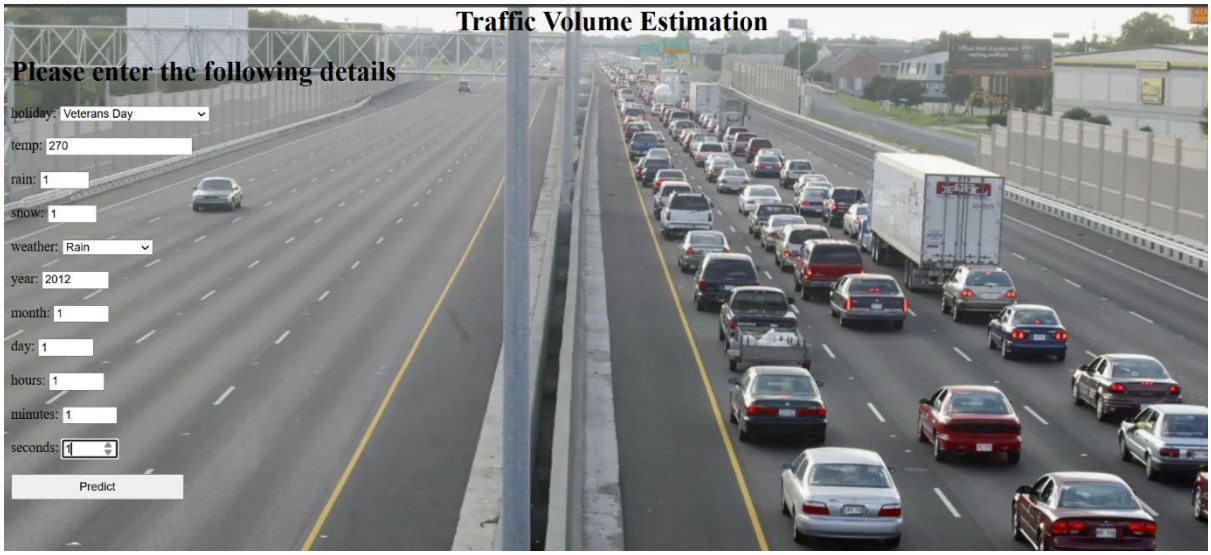
9. **User Interface**

- Input Form: For entering traffic parameters.
- Output Screen: Displays predicted traffic volume.

10. **Testing**

- Manual Testing: Input combinations tested on form.
- Model evaluated using R^2 score and RMSE.

11. **Screenshots or Demo**

- 

- 

12. **Known Issues**

- UI is basic and can be improved.
- No data validation on input fields.

13. **Future Enhancements**

- Integration with live traffic APIs.
- Add user authentication.
- Add map visualization.
- Deploy on cloud using Docker and CI/CD pipelines.