

## Traffic Prediction System Documentation




### Introduction

This project is a Traffic Prediction System that predicts traffic conditions based on user input like location, date, time, and traffic percentage. It utilizes Flask for the backend, HTML/CSS/JavaScript for the frontend, and Random ForestClassifier for machine learning-based predictions.

### Team Members

- B. Manoj
- R. Lavanya
- CH. Siva ram
- G. Ajay
- B. Ashok

### Project Structure

-  Dataset (traffic\_dataset\_5000.csv) - Contains traffic data for prediction.
-  UI Page (ui.html) - Provides the user interface for data input.
-  Backend (app.py) - Handles data processing, model training, and prediction endpoints.






### Prerequisites

Ensure the following libraries are installed:




- Flask
- Pandas
- Scikit-learn

### Dataset Details




The dataset contains the following key columns:

-  Location (Categorical)
-  Date (Date format)
-  Time (Time format)
-  Traffic\_Status (Target variable)
-  Percentage (Optional feature)

### Key Functionalities

-  Data Preprocessing: Encodes categorical data, converts date and time into numeric formats, and handles missing data.
-  Model Training: Uses RandomForestClassifier with optimized parameters.
-  Prediction Endpoint: Accepts JSON data for predictions and returns the predicted traffic status.

### Frontend Design

-  Clean and user-friendly interface for data input.
-  Provides dropdown options for selecting Andhra Pradesh districts.
-  Displays predicted traffic status dynamically for better clarity.

### Deployment Steps



1. Install required dependencies.



Edit with WPS Office

2. Run the Flask application.
3. Access the UI via <http://localhost:5000>.
4. Enter the required details in the form and click Submit to view the prediction.

### Error Handling

-  Displays specific error messages for missing fields or invalid data entries.
-  Ensures stable performance by handling unexpected server errors gracefully.

### Future Improvements

- Implement real-time data collection for improved accuracy.
- Add visualizations for better data insights.
- Optimize the model with hyperparameter tuning.

### Conclusion

This Traffic Prediction System efficiently predicts traffic conditions using machine learning and provides an intuitive user interface for seamless interaction. The structured codebase ensures easy maintenance and scalability.



Edit with WPS Office