**Student Name: [*D.Vishwanathan***]

**Register Number:** [622423243030]

# Institution: [SALEM COLLEGE OF ENGINEERING AND TECHNOLOGY]

# Department: [ B.TECH/ARTIFICIAL INTELLIGENCE AND DATA SCIENCE]

# GitHub Repository Link:https://github.com/vishwa1436/Vishwa06

# Date of Submission: [19/5/2025]

**1. Problem Statement**

Road traffic accidents pose a major threat to public safety, resulting in loss of life, injuries, and economic damage. The aim of this project is to leverage AI and Machine Learning models to analyze traffic accident data and predict accident-prone zones or times, enabling authorities to take proactive safety measures.  
✔ **Problem Type:** Multiclass Classification / Regression  
✔ **Business Relevance:** Enhance urban planning, reduce road accidents, support smart city infrastructure.

**2. Abstract**

This project presents an AI-based system for analyzing historical traffic accident data to identify critical factors leading to road incidents and to predict future occurrences. Using machine learning techniques such as Random Forest, Gradient Boosting, and deep learning models, the system identifies accident hotspots and forecasts accident probability based on conditions like weather, time, location, and vehicle type. Visual dashboards provide actionable insights, and the prediction model is deployed through a web interface for real-time alerts.

**3. System Requirements**

**Hardware:**  
✔ RAM: Minimum 8GB  
✔ Processor: Intel i5 / Ryzen 5 or higher  
✔ GPU (Optional for deep learning)

**Software:**  
✔ Python 3.8+  
✔ Libraries: numpy, pandas, matplotlib, seaborn, scikit-learn, xgboost, keras, tensorflow  
✔ IDE: Google Colab / Jupyter Notebook  
✔ Deployment Tools: Streamlit / Flask

**4. Objectives**

✔ Analyze and visualize key patterns in accident data  
✔ Build ML models to predict accident likelihood based on features  
✔ Identify accident hotspots using geospatial mapping  
✔ Provide real-time risk prediction to users through an interactive UI

**5. Flowchart of Project Workflow**

*(Include a block diagram with the following steps:)*  
**Data Collection → Data Preprocessing → EDA → Feature Engineering → Model Training → Evaluation → Deployment → Prediction Dashboard**

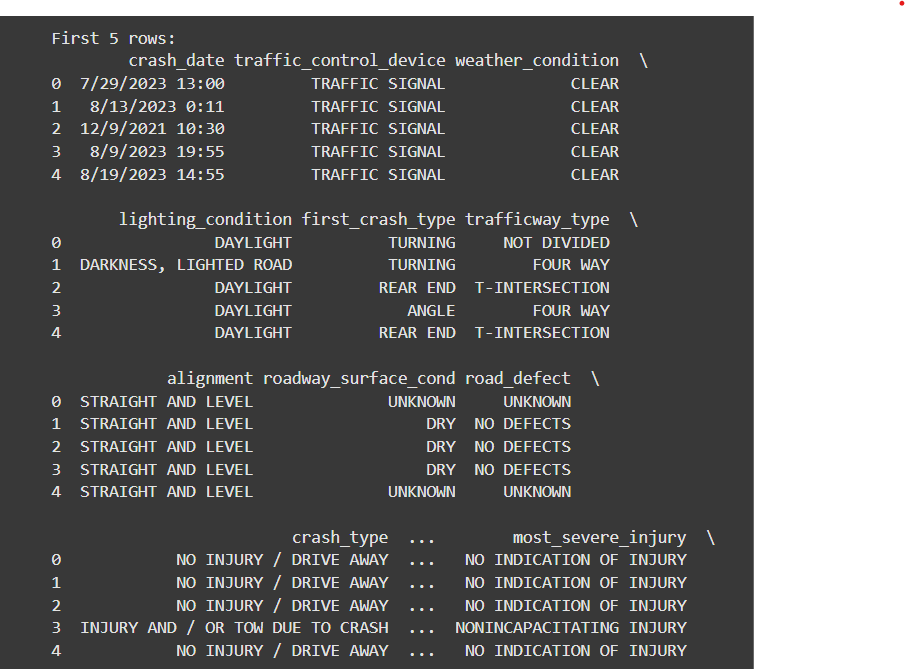
**6. Dataset Description**

✔ **Source:** Kaggle / Government Open Data Portals  
✔ **Type:** Public Dataset  
✔ **Size:** ~10,000+ entries (Columns: Time, Date, Location, Weather, Vehicle Type, Severity, etc.)  
✔ **Example Screenshot:**



**7. Data Preprocessing**

✔ Handle missing values, encode categorical variables  
✔ Normalize continuous features  
✔ Feature extraction (e.g., hour from timestamp)  
✔ Before/After Sample:



**8. Exploratory Data Analysis (EDA)**

✔ Time-series analysis (peak accident hours)  
✔ Weather-condition impact on accident frequency  
✔ Location heatmaps using geospatial data  
✔ Visualizations: Heatmaps, Histograms, Correlation Plots, Bar Charts  
✔ Key Insights: Accidents are more frequent during rainy conditions and peak traffic hours

**9. Feature Engineering**

✔ Encode categorical variables: One-hot/Label Encoding  
✔ Time & weather-based feature extraction  
✔ Feature importance analysis using Random Forest  
✔ Dimensionality reduction (PCA for visualization)

**10. Model Building**

✔ Baseline models: Logistic Regression, Decision Trees  
✔ Advanced models: Random Forest, XGBoost, LSTM (for temporal prediction), ANN  
✔ Final Model: XGBoost (or LSTM if temporal data used)  
✔ Screenshots of training metrics and model performance graphs

**11. Model Evaluation**

✔ Evaluation Metrics: Accuracy, Precision, Recall, F1-score, RMSE (for regression)  
✔ Confusion Matrix and ROC Curve  
✔ Model Comparison Table  
✔ Error Analysis: Common misclassifications and their patterns  
✔ Screenshots of evaluation plots

**12. Deployment**

✔ Deployment Method: Streamlit / Flask on local or cloud server  
✔ Real-time Prediction Demo  
✔ Sample Input: Time, Weather, Vehicle Type → Output: Risk Level  
✔ Sample Screenshot:



**13. Source Code**

✔ **GitHub Repository Link:** https://github.com/vishwa1436/Vishwa06  
**Python Program Snippet (Simplified):**

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import accuracy\_score

# Sample dataset (in a real scenario, this would be loaded from a CSV file)

data = {

'Time': ['Morning', 'Evening', 'Night', 'Morning', 'Afternoon', 'Night'],

'Weather': ['Clear', 'Rainy', 'Clear', 'Foggy', 'Clear', 'Rainy'],

'Road\_Condition': ['Dry', 'Wet', 'Dry', 'Wet', 'Dry', 'Wet'],

'Accident\_Severity': ['Minor', 'Major', 'None', 'Major', 'None', 'Minor']

}

# Create DataFrame

df = pd.DataFrame(data)

# Preprocess data

df = pd.get\_dummies(df, columns=['Time', 'Weather', 'Road\_Condition'])

# Prepare features and target

X = df.drop('Accident\_Severity', axis=1)

y = df['Accident\_Severity']

# Split data

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)

# Train model

model = RandomForestClassifier()

model.fit(X\_train, y\_train)

# Predict and evaluate

predictions = model.predict(X\_test)

accuracy = accuracy\_score(y\_test, predictions)

print("Sample Dataset:")

print(pd.DataFrame(data))

print("\nModel Accuracy:", accuracy)

print("\nSample Prediction on Test Data:")

print(pd.DataFrame({'Actual': y\_test, 'Predicted': predictions}))

**14. Future Scope**

✔ Integrate live traffic and weather feeds for real-time prediction  
✔ Extend system for accident severity estimation  
✔ Mobile app development for accident alerts  
✔ Collaboration with city planning authorities for intelligent traffic control

**15.Team Members and Roles**

|  |  |  |
| --- | --- | --- |
| SNO | *TEAM MEMBERS* | *ROLE & RESPONSIBILITIES* |
| *1* | *D.Vishwanathan* | Objective,scope |
| *2* | *RK.Vishal* | Problem,data sources |
| *3* | *P.Vasanth* | High-Level Methodology,challenges ,risks |
| *4* | *V.Velvizhi* | Tools and Technologies, Expected Outcomes |