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

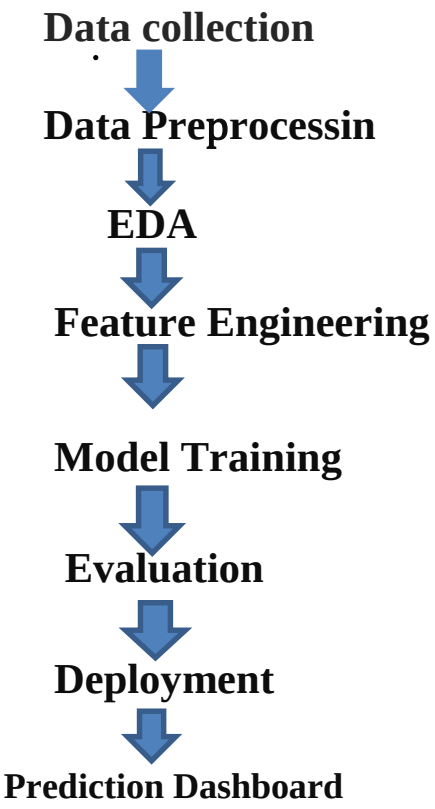
---

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



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:**

```
Sample Dataset:
          Time Weather Road_Condition Accident_Severity
0    Morning   Clear            Dry             Minor
1    Evening   Rainy            Wet             Major
2      Night   Clear            Dry              None
3    Morning   Foggy            Wet             Major
4  Afternoon   Clear            Dry              None
5      Night   Rainy            Wet             Minor

Model Accuracy: 0.0

Sample Prediction on Test Data:
  Actual Predicted
0  Minor      None
1  Major     Minor
```

---

7. Data Preprocessing

✓ Handle missing values, encode categorical variables
✓ Normalize continuous features

✓ Feature extraction (e.g., hour from timestamp)
✓ Before/After Sample:

```
First 5 rows:
        crash_date traffic_control_device weather_condition  \
0  7/29/2023 13:00         TRAFFIC SIGNAL             CLEAR
1   8/13/2023 0:11         TRAFFIC SIGNAL             CLEAR
2  12/9/2021 10:30         TRAFFIC SIGNAL             CLEAR
3   8/9/2023 19:55         TRAFFIC SIGNAL             CLEAR
4  8/19/2023 14:55         TRAFFIC SIGNAL             CLEAR

       lighting_condition first_crash_type trafficway_type  \
0                DAYLIGHT          TURNING     NOT DIVIDED
1  DARKNESS, LIGHTED ROAD          TURNING        FOUR WAY
2                DAYLIGHT         REAR END  T-INTERSECTION
3                DAYLIGHT            ANGLE        FOUR WAY
4                DAYLIGHT         REAR END  T-INTERSECTION

            alignment roadway_surface_cond road_defect  \
0  STRAIGHT AND LEVEL              UNKNOWN     UNKNOWN
1  STRAIGHT AND LEVEL                  DRY  NO DEFECTS
2  STRAIGHT AND LEVEL                  DRY  NO DEFECTS
3  STRAIGHT AND LEVEL                  DRY  NO DEFECTS
4  STRAIGHT AND LEVEL              UNKNOWN     UNKNOWN

                        crash_type  ...        most_severe_injury  \
0           NO INJURY / DRIVE AWAY  ...   NO INDICATION OF INJURY
1           NO INJURY / DRIVE AWAY  ...   NO INDICATION OF INJURY
2           NO INJURY / DRIVE AWAY  ...   NO INDICATION OF INJURY
3  INJURY AND / OR TOW DUE TO CRASH  ...  NONINCAPACITATING INJURY
4           NO INJURY / DRIVE AWAY  ...   NO INDICATION OF INJURY
```

---

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:

```
Sample Dataset:
         Time Weather Road_Condition Accident_Severity
0     Morning   Clear            Dry            Minor
1     Evening   Rainy            Wet            Major
2       Night   Clear            Dry             None
3     Morning   Foggy            Wet            Major
4   Afternoon   Clear            Dry             None
5       Night   Rainy            Wet            Minor

Model Accuracy: 0.0

Sample Prediction on Test Data:
  Actual Predicted
0  Minor      None
1  Major     Minor
```

---

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**