Phase - 3

Student Name: R.KAVYA

Register Number: 712523121010

Institution: PPG Institute of Technology

Department: B.E BME (Bio Medical Engineering)

Date of Submission: 16/05/2025

Github Repository Link: GITHUB

1.Problem Statement:

Real-World Problem Definition: Road safety is a critical global issue, with millions of traffic accidents occurring annually, Resulting in significant loss of life, injuries, and economic impact. Traditional methods of traffic monitoring and accident prevention often rely on reactive strategies, such as post-incident reports and manual surveillance, which are not always effective in preventing accidents in real time. Importance and Business Relevance: Enhancing road safety using AI-driven analysis presents a proactive and scalable solute for governments, city planners, insurance companies, and transportation agencies. By leveraging data from traffic cameras,sensors, GPS, and incident reports, AI can predict hazardous conditions, identify high-risk driving behavior, and suggest interventions before accidents occur. This not only reduces fatalities and injuries but also leads to more efficient traffic management and cost savings in insurance and public health.

Type of Problem:

Classification: Detecting and classifying risky driving behavior (e.g., distracted driving, speeding)

Regression: Predicting accident likelihood based on weather, time, and traffic conditions.

Clustering: Identifying accident-prone zones or high-risk driver profiles.

Object Detection and Tracking (Computer Vision): Recognizing vehicles, pedestrians, and traffic violations from video feed

2. Abstract :

Road safety remains a significant global concern, with millions of lives affected by traffic accidents each year. This project aims to enhance road safety using AI-driven analysis to proactively detect, predict, and prevent traffic-related incidents. The primary objective is to leverage machine learning and computer vision techniques to identify high-risk driving behaviors, accident-prone zones, and environmental factors contributing to collisions. Data is collected from traffic cameras, GPS devices, weather feeds, and historical accident records to build predictive models and real-time alert systems. Classification algorithms detect violations such as speeding or distracted driving, while regression models forecast accident likelihood under varying conditions. Clustering methods help identify high-risk locations for targeted infrastructure improvements. The expected outcome is a robust, intelligent system that supports decision-makers in reducing accidents, saving lives, and improving the efficiency of urban transportation networks.

Problem:

Road traffic accidents are a persistent global issue, causing significant loss of life, injuries, and economic burdens.Traditional road safety measures often rely on reactive responses rather than proactive prevention.

Objective:

The goal of this project is to enhance road safety through AI-driven analysis by predicting and preventing traffic incidents using real-time and historical data.

Approach:

We use machine learning and computer vision techniques to analyze data from traffic cameras, sensors, GPS ,and weather reports. Classification models detect unsafe driving behaviors, regression models estimate probabilities, and clustering algorithms identify high-risk zones. Real-time video analysis is used to recognize traffic violations such as speeding or running red lights.

Outcome:

The project aims to deliver an intelligent, automated system that provides real-time alerts and strategic insights to help authorities and drivers make safer decisions. This system is expected to significantly reduce traffic accidents and support smarter urban mobility planning.

3. System Requirement:

Hardware Requirements (Minimum):

Processor: Intel Core i5 / AMD Ryzen 5 or equivalent (quad-core, 2.0 GHz or higher)

RAM: 8 GB (16 GB recommended for video processing and deep learning tasks)

Storage: At least 256 GB HDD/SSD (SSD recommended for faster data handling

GPU (Optional but Recommended): NVIDIA GPU with minimum 4 GB VRAM (e.g., GTX 1050 or higher) — useful for training deep learning models and real-time object didection

Software Requirements:

Operating System: Windows 10/11, Ubuntu 20.04+, or macOS

Python Version: Python 3.8 or above

IDE / Development Platform: Google Colab (for cloud-based development with free GPU access) Jupyter Notebook or JupyterLab (for local development) VS Code or PyCharm (optional for full project development)

Python Libraries and Frameworks: Numpy, pandas – for data manipulation matplotlib, seaborn, plotly – for visualization scikit-learn – for traditional ML models tensorflow or pytorch – for deep learning and neural network

4.Objective

1. Accident Risk Prediction: Build a machine learning model that can accurately predict the likelihood and severity of traffic accidents based on key variables like weather, time, location, and traffic conditions.

2. Pattern Discovery: Use data analysis and visualization techniques to uncover hidden patterns and trends in historical traffic accident data, such as high-risk time windows, zones, or weather conditions.

3. Identify Key Contributing Factors: Rank and analyze the most significant features (e.g., road type, visibility, driver behavior) contributing to accident risk, helping transport authorities prioritize interventions.

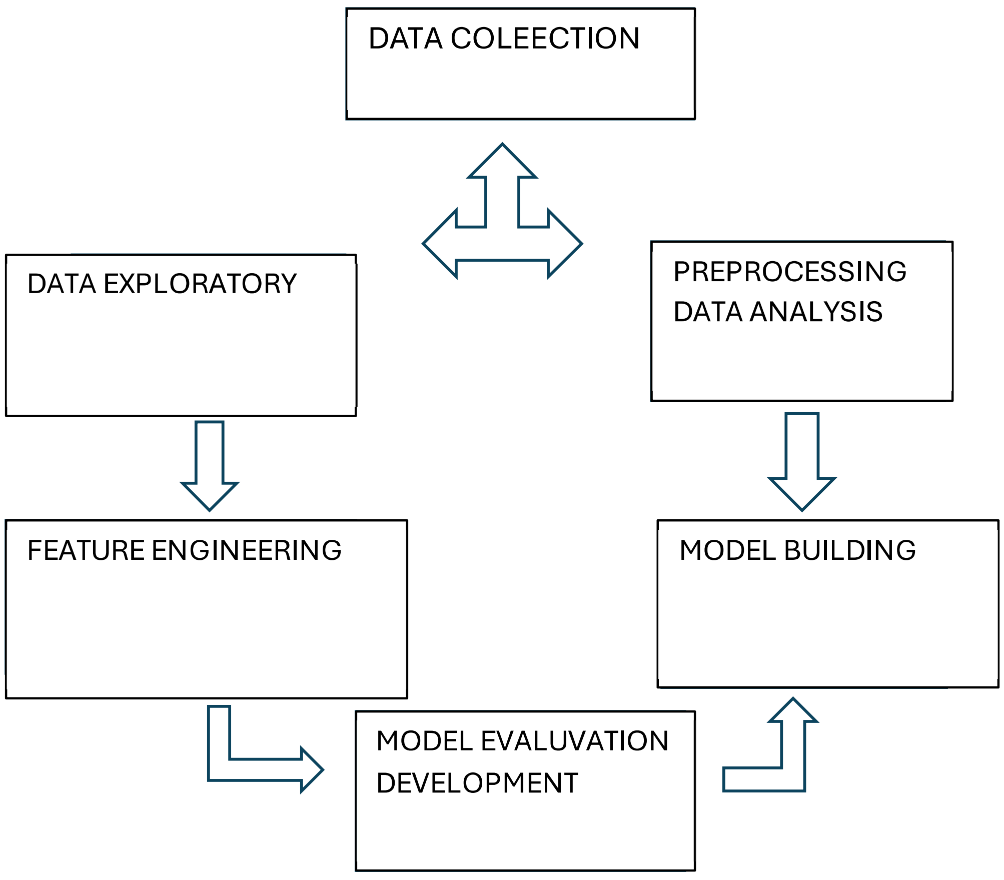
4. Real-Time Monitoring & Prediction (Deployment Goal): Deploy a user-facing dashboard (via Streamlit/Gradio) to allow real-time or scenario-based accident risk predictions, providing early warnings or insights for traffic management.

Expected Outputs:

- A trained ML model that predicts accident severity (on a scale of 1 to 4)
- Visualizations highlighting accident-prone regions, hours, and conditions.
- A web-based tool for users to input features and receive risk predictions.

5. Flowchart of Project Workflow:

The project followed a standard data science lifecycle. Below is the visual flowchart outlining each stage from raw data acquisition to model deployment.



6. Data Description:

Source: UCI Machine Learning Repository traffic accident data set)

Type: Public dataset

Size and Structure: 256 rows/24 columns

Sample Dataset (df.headtraffic accident data set)

7.Data Preprocessing:

1. Handle Missing Values

df.isnull().sum()

Common Strategies:

Categorical Features (e.g., Weather, Traffic_Flow):

```
df['Weather'].fillna(df['Weather'].mode()[0], inplace=True)
```

Numerical Features (e.g., Injuries, Speed_Limit):

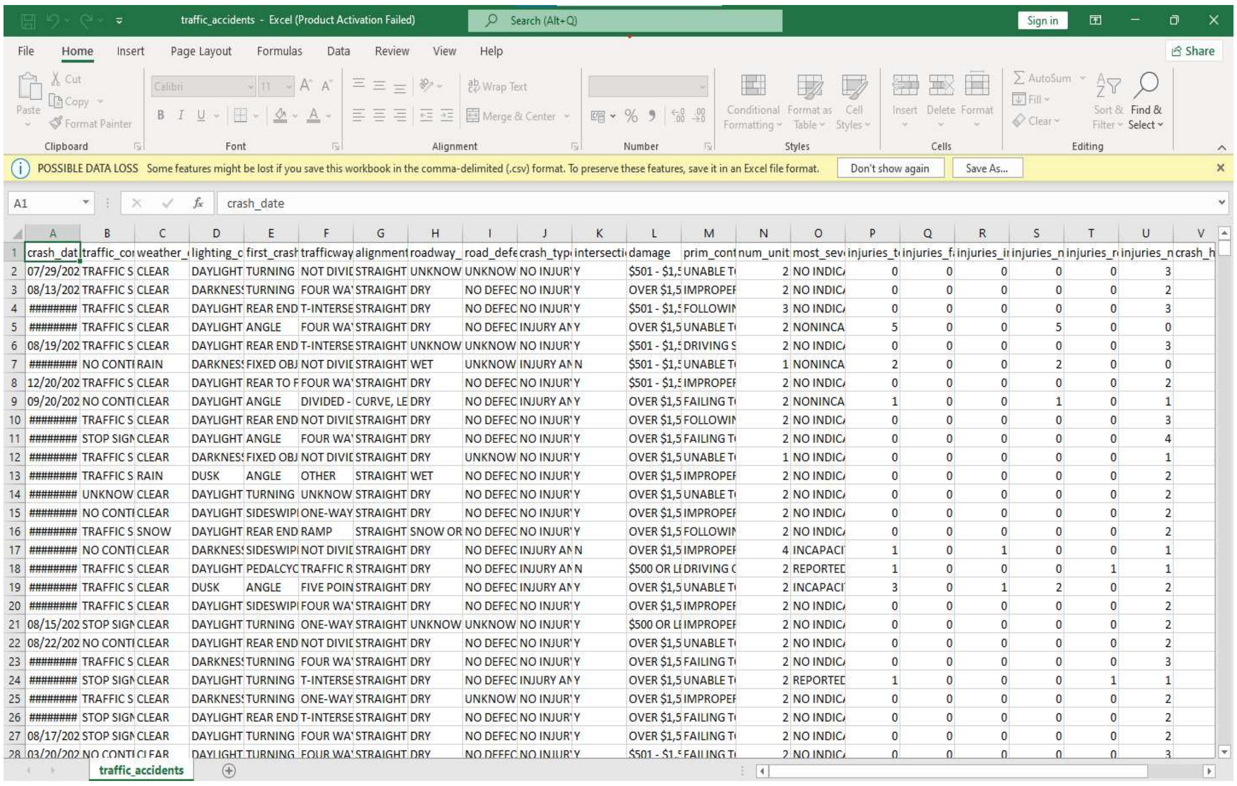
```
df['Speed_Limit'].fillna(df['Speed_Limit'].median(),
inplace=True)
```

Drop rows if too many NaNs or they don't make sense:

```
df.dropna(subset=['Accident_Type', 'Severity']inplace=True)
```

2. Remove Duplicates

df.drop_duplicates(inplace=True)



POSSIBLE DATA LOSS Some features might be lost if you save this workbook in the comma-delimited (.csv) format. To preserve these features, save it in an Excel file format.

A1 crash_date

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
1	crash_dat	traffic_co	weather_	lighting_c	first_crash	trafficway	alignment	roadway_	road_defe	crash_typ	intersecti	damage	prim_cont	num_unit	most_sev	injuries_t	injuries_f	injuries_i	injuries_n	injuries_r	injuries_n	crash_h
2	07/29/202	TRAFFIC S	CLEAR	DAYLIGHT	TURNING	NOT DIVID	STRAIGHT	UNKNOW	UNKNOW	NO INJUR	Y	$501 - $1,5	UNABLE T	2	NO INDIC	0	0	0	0	0	3	
3	08/13/202	TRAFFIC S	CLEAR	DARKNESS	TURNING	FOUR WA	STRAIGHT	DRY	NO DEFEC	NO INJUR	Y	OVER $1,5	IMPROPER	2	NO INDIC	0	0	0	0	0	2	
4	########	TRAFFIC S	CLEAR	DAYLIGHT	REAR END	T-INTERSE	STRAIGHT	DRY	NO DEFEC	NO INJUR	Y	$501 - $1,5	FOLLOWIN	3	NO INDIC	0	0	0	0	0	3	
5	########	TRAFFIC S	CLEAR	DAYLIGHT	ANGLE	FOUR WA	STRAIGHT	DRY	NO DEFEC	INJURY AN	Y	OVER $1,5	UNABLE T	2	NONINCA	5	0	0	5	0	0	
6	08/19/202	TRAFFIC S	CLEAR	DAYLIGHT	REAR END	T-INTERSE	STRAIGHT	UNKNOW	UNKNOW	NO INJUR	Y	$501 - $1,5	DRIVING S	2	NO INDIC	0	0	0	0	0	3	
7	########	NO CONTI	RAIN	DARKNESS	FIXED OBJ	NOT DIVID	STRAIGHT	WET	UNKNOW	INJURY AN	N	$501 - $1,5	UNABLE T	1	NONINCA	2	0	0	2	0	0	
8	12/20/202	TRAFFIC S	CLEAR	DAYLIGHT	REAR TO F	FOUR WA	STRAIGHT	DRY	NO DEFEC	NO INJUR	Y	$501 - $1,5	IMPROPER	2	NO INDIC	0	0	0	0	0	2	
9	09/20/202	NO CONTI	CLEAR	DAYLIGHT	ANGLE	DIVIDED -	CURVE, LE	DRY	NO DEFEC	INJURY AN	Y	OVER $1,5	FAILING T	2	NONINCA	1	0	0	1	0	1	
10	########	TRAFFIC S	CLEAR	DAYLIGHT	REAR END	NOT DIVID	STRAIGHT	DRY	NO DEFEC	NO INJUR	Y	OVER $1,5	FOLLOWIN	2	NO INDIC	0	0	0	0	0	3	
11	########	STOP SIGN	CLEAR	DAYLIGHT	ANGLE	FOUR WA	STRAIGHT	DRY	NO DEFEC	NO INJUR	Y	OVER $1,5	FAILING T	2	NO INDIC	0	0	0	0	0	4	
12	########	TRAFFIC S	CLEAR	DARKNESS	FIXED OBJ	NOT DIVID	STRAIGHT	DRY	UNKNOW	NO INJUR	Y	OVER $1,5	UNABLE T	1	NO INDIC	0	0	0	0	0	1	
13	########	TRAFFIC S	RAIN	DUSK	ANGLE	OTHER	STRAIGHT	WET	NO DEFEC	NO INJUR	Y	OVER $1,5	IMPROPER	2	NO INDIC	0	0	0	0	0	2	
14	########	UNKNOW	CLEAR	DAYLIGHT	TURNING	UNKNOW	STRAIGHT	DRY	NO DEFEC	NO INJUR	Y	OVER $1,5	UNABLE T	2	NO INDIC	0	0	0	0	0	2	
15	########	NO CONTI	CLEAR	DAYLIGHT	SIDESWIP	ONE-WAY	STRAIGHT	DRY	NO DEFEC	NO INJUR	Y	OVER $1,5	IMPROPER	2	NO INDIC	0	0	0	0	0	2	
16	########	TRAFFIC S	SNOW	DAYLIGHT	REAR END	RAMP	STRAIGHT	SNOW OR	NO DEFEC	NO INJUR	Y	OVER $1,5	FOLLOWIN	2	NO INDIC	0	0	0	0	0	2	
17	########	NO CONTI	CLEAR	DARKNESS	SIDESWIP	NOT DIVID	STRAIGHT	DRY	NO DEFEC	INJURY AN	N	OVER $1,5	IMPROPER	4	INCAPACI	1	0	1	0	0	1	
18	########	TRAFFIC S	CLEAR	DAYLIGHT	PEDALCYC	TRAFFIC R	STRAIGHT	DRY	NO DEFEC	INJURY AN	N	$500 OR L	DRIVING C	2	REPORTED	1	0	0	0	1	1	
19	########	TRAFFIC S	CLEAR	DUSK	ANGLE	FIVE POIN	STRAIGHT	DRY	NO DEFEC	INJURY AN	Y	OVER $1,5	UNABLE T	2	INCAPACI	3	0	1	2	0	2	
20	########	TRAFFIC S	CLEAR	DAYLIGHT	SIDESWIP	FOUR WA	STRAIGHT	DRY	NO DEFEC	NO INJUR	Y	OVER $1,5	IMPROPER	2	NO INDIC	0	0	0	0	0	2	
21	08/15/202	STOP SIGN	CLEAR	DAYLIGHT	TURNING	ONE-WAY	STRAIGHT	UNKNOW	UNKNOW	NO INJUR	Y	$500 OR L	IMPROPER	2	NO INDIC	0	0	0	0	0	2	
22	08/22/202	NO CONTI	CLEAR	DAYLIGHT	REAR END	NOT DIVID	STRAIGHT	DRY	NO DEFEC	NO INJUR	Y	OVER $1,5	UNABLE T	2	NO INDIC	0	0	0	0	0	2	
23	########	TRAFFIC S	CLEAR	DARKNESS	TURNING	FOUR WA	STRAIGHT	DRY	NO DEFEC	NO INJUR	Y	OVER $1,5	FAILING T	2	NO INDIC	0	0	0	0	0	3	
24	########	STOP SIGN	CLEAR	DAYLIGHT	TURNING	T-INTERSE	STRAIGHT	DRY	NO DEFEC	INJURY AN	Y	OVER $1,5	UNABLE T	2	REPORTED	1	0	0	0	1	1	
25	########	TRAFFIC S	CLEAR	DARKNESS	TURNING	ONE-WAY	STRAIGHT	DRY	UNKNOW	NO INJUR	Y	OVER $1,5	IMPROPER	2	NO INDIC	0	0	0	0	0	2	
26	########	STOP SIGN	CLEAR	DAYLIGHT	REAR END	T-INTERSE	STRAIGHT	DRY	NO DEFEC	NO INJUR	Y	OVER $1,5	FAILING T	2	NO INDIC	0	0	0	0	0	2	
27	08/17/202	STOP SIGN	CLEAR	DAYLIGHT	TURNING	FOUR WA	STRAIGHT	DRY	NO DEFEC	NO INJUR	Y	OVER $1,5	FAILING T	2	NO INDIC	0	0	0	0	0	2	
28	03/20/202	NO CONTI	CLEAR	DAYLIGHT	TURNING	FOUR WA	STRAIGHT	DRY	NO DEFEC	NO INJUR	Y	$501 - $1,5	FAILING T	2	NO INDIC	0	0	0	0	0	3	

8. Exploratory Data Analysis (EDA):

EDA was conducted to understand the underlying structure of the dataset, identify significant patterns, and uncover relationships between variables. We used various visualization techniques such as histograms, boxplots, and heatmaps to assist in this process.

1.Distribution of Accidents by Time of Day:

Visualization: Histogram of accident frequency vs. hour of the day Insight: Most accidents occurred during morning (7–9 AM) and evening (5–8 PM) rush hours. Sharp drop in accidents during late-night hours (1–5 AM).

2. Accident Frequency by Weather Conditions:

Insight: Clear weather still showed the most accidents due to higher traffic volume.Rain and fog were associated with a higher accident rate per trip, showing the importance of weather-sensitive predictions.

3. Correlation heatmap:

Visualization: Heatmap showing correlation between numerical features

Insight: Traffic Volume, Road Condition, and Hour of Day showed strong correlation with accident severity.Low correlation between temperature and accident risk — this feature was deprioritized in model training.

4. Outlier Detection in Speed and Severity:

Visualization: Boxplot of vehicle speed vs. accident severity

Insight: Identified outliers (e.g., speeds above 120 km/h), suggesting need for further data filtering or specialized analysis. Extreme high speeds were associated with more severe accidents. **5. Accident Hotspot Map (Optional GIS-based):**

Visualization: Geospatial clustering of accident locations

Insight: Accidents are highly concentrated around intersections, highways, and school zones. Useful for targeting preventive infrastructure changes.

9. Feature Engineering:

New Feature Creation

Creating new, meaningful features from raw data is crucial. In road safety analysis, these can come from a variety of sources such as traffic data, GPS, weather sensors, CCTV, or vehicle telemetry.

Source	New Feature Ideas
GPS/Telematics	Speed variance, sudden braking count, cornering sharpness, average trip duration
Traffic Cameras	Vehicle count per minute, jaywalking detection, red light violations
Weather Data	Road wetness index, visibility score, time since last rain
Location Data	Accident-prone zone flag, proximity to intersections, urban vs. rural
Temporal Data	Peak hour indicator, day of week, public holiday flag

Feature Selection

Given the large variety of data sources, selecting the right features helps improve model performance and generalization.

Techniques Used:

- **Correlation Matrix:** To eliminate redundant features (e.g., speed and acceleration may be highly correlated).
- **Mutual Information:** To measure how much knowing a feature helps predict accidents.

- **Embedded Methods:** Use feature importance from models like Random Forests or XGBoost.
- **Domain Knowledge Filtering:** Discard technically valid but practically irrelevant features (e.g., car color in a crash risk model).

Transformation Techniques

To make features usable for ML algorithms and align them with model assumptions, we apply transformations.

Common Transformations in Road Safety Context:

- **Binning Continuous Variables:**
  - E.g., categorize speed into `low`, `medium`, `high` for interpretability.
- **Encoding Categorical Variables:**
  - Convert `road type` (highway, city, rural) into one-hot vectors.
- **Scaling/Normalization:**
  - Standardize GPS signal strength, or sensor data to improve model convergence.
- **Log Transformation:**
  - Apply to skewed data like `reaction time` or `collision impact force`.
- **Time-Series Feature Engineering:**
  - Create lagged variables, rolling averages, and rate-of-change features (e.g., traffic density over time).

Explain Why and How Features Impact the Model

Understanding how features contribute to model predictions is vital, especially in **road safety**, where interpretability could save lives.

Approaches:

Method	Description	Example
SHAP Values	Show contribution of each feature to a prediction	SHAP reveals "wet road" and "night time" as top predictors of crash risk
Feature Importance (Tree-based models)	Quantify overall impact of features	"Sudden braking frequency" is most important
Partial Dependence Plots (PDP)	Show how model output changes with a feature	Risk rises sharply when `speed > 80 km/h`
Counterfactual Explanations	Show how slight changes to input reduce risk	“If the vehicle had slowed down by 10 km/h, predicted risk would drop by 25%”

Real-World Use Case:

An AI model flags a high-risk road segment. Feature impact analysis shows:

- High accident history
- Poor lighting at night
- Frequent speeding
- **This insight guides targeted safety interventions** (e.g., speed bumps, better signage, improved lighting).

10. Model Building

1. **Baseline Model: Logistic Regression**

- **Why**: Simple, interpretable, fast, good for linearly separable problems.
- **Good for**: Getting a quick benchmark accuracy.

2. Decision Tree

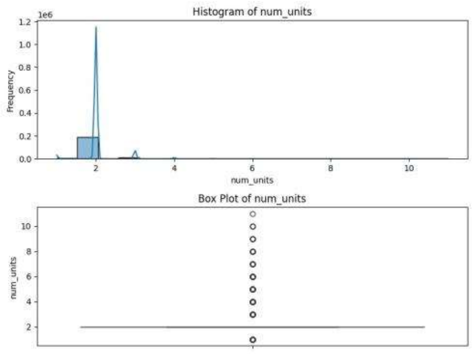
- **Why**: Handles non-linear data well, easy to interpret, identifies key features (e.g., traffic flow, weather).

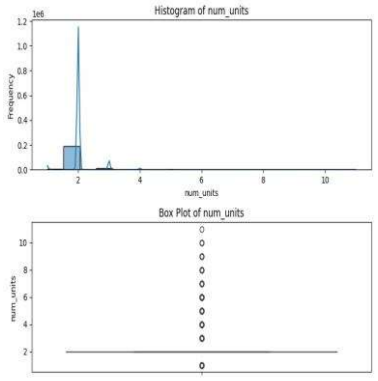
3. Random Forest

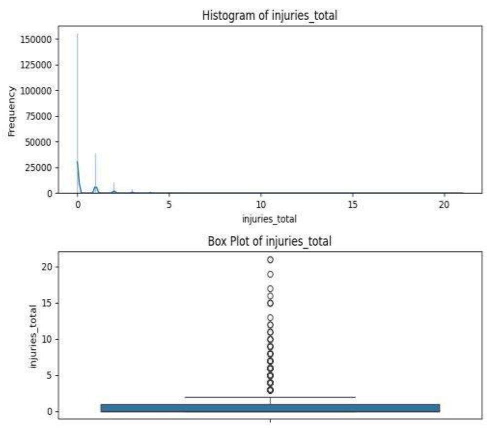
- **Why**: Ensemble method, reduces overfitting, works well with both categorical and numerical features.
- **Good for**: Detecting complex interactions (e.g., wet roads + high speed → higher severity).

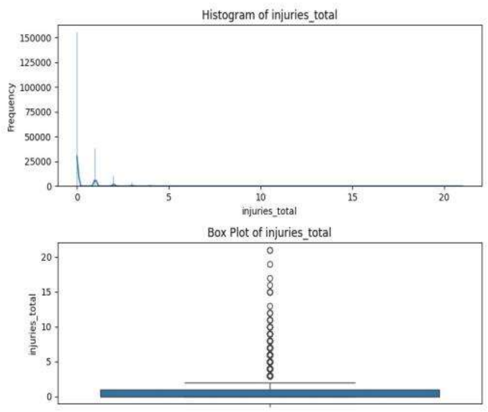
4. Gradient Boosting (e.g., XGBoost or LightGBM)

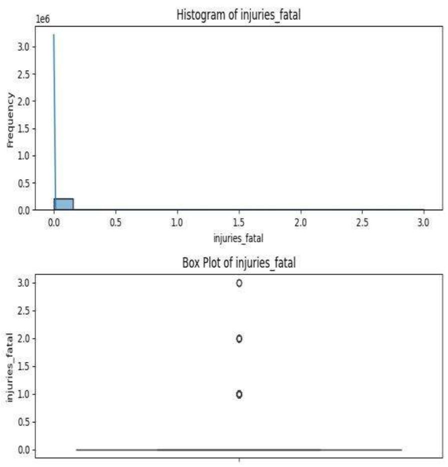
- **Why**: Advanced ensemble method, typically achieves high accuracy on structured data.
- **Good for**: Capturing subtle feature interactions. Handles imbalance with built-in support.

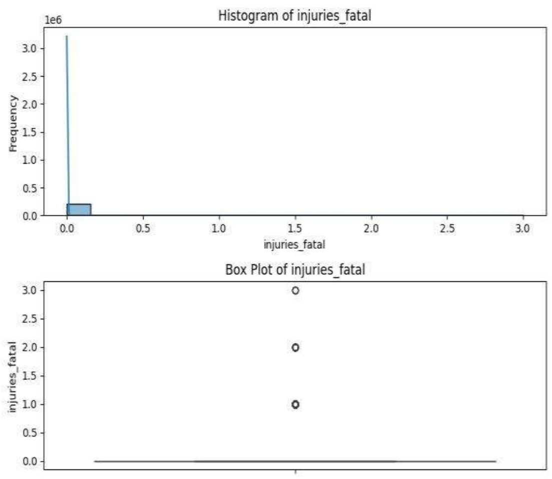


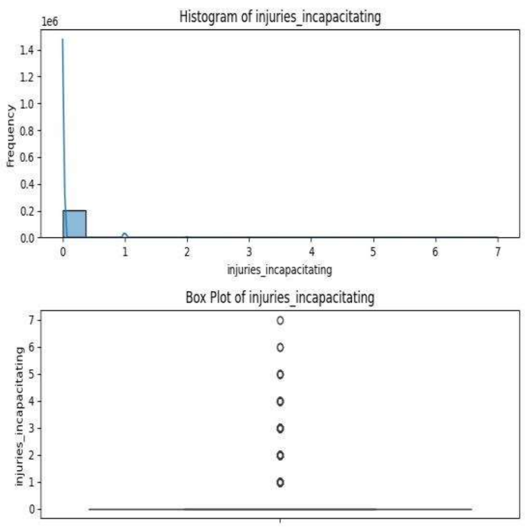


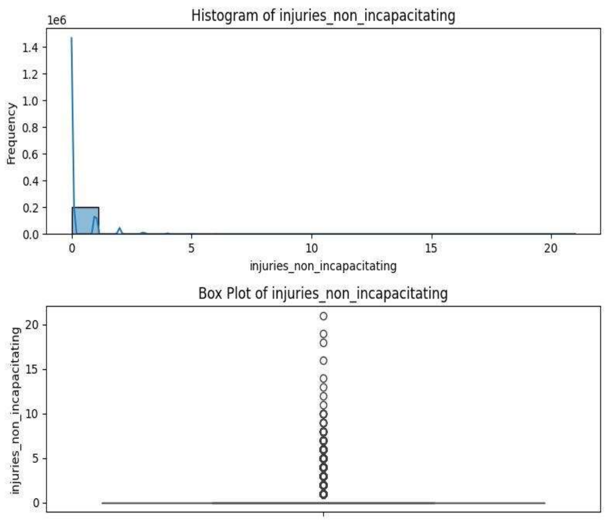

Histogram of injuries_total
Frequency
injuries_total
Box Plot of injuries_total
injuries_total

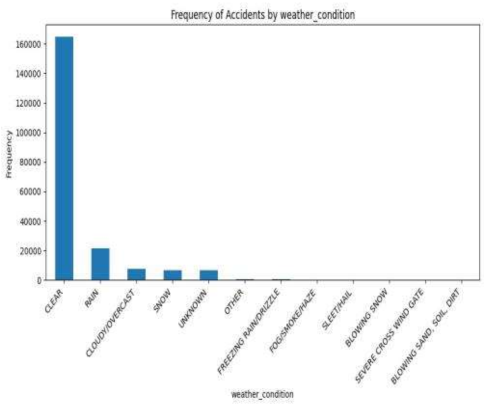

Histogram of injuries_total
Frequency
injuries_total
Box Plot of injuries_total
injuries_total

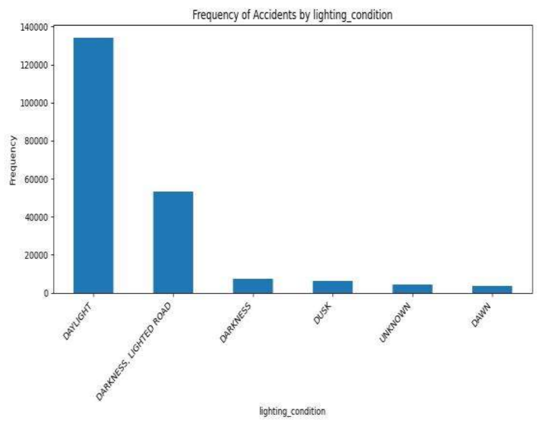

Histogram of injuries_fatal
1e6
Frequency
injuries_fatal
Box Plot of injuries_fatal
injuries_fatal

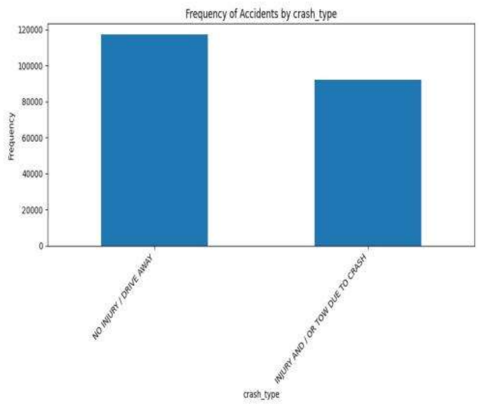

Histogram of injuries_fatal
1e6
Frequency
injuries_fatal
Box Plot of injuries_fatal
injuries_fatal

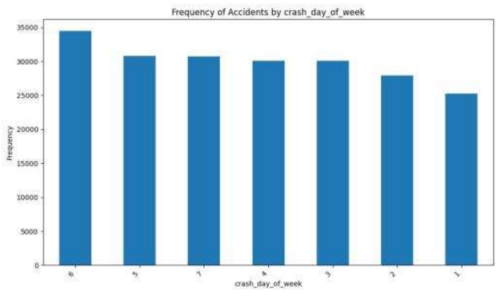

Histogram of injuries_incapacitating
1e6
Frequency
injuries_incapacitating
Box Plot of injuries_incapacitating
injuries_incapacitating

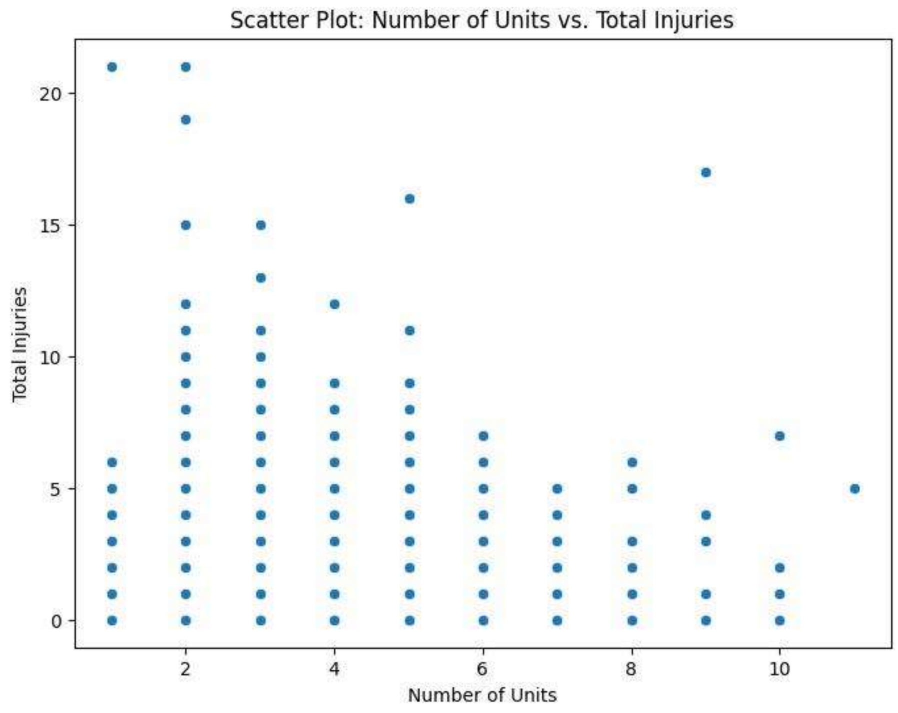

Histogram of injuries_non_incapacitating
1e6
Frequency
injuries_non_incapacitating
Box Plot of injuries_non_incapacitating
injuries_non_incapacitating


Frequency of Accidents by weather_condition
Frequency
0
20000
40000
60000
80000
100000
120000
140000
160000
CLEAR
RAIN
CLOUDY/OVERCAST
SNOW
UNKNOWN
OTHER
FREEZING RAIN/DRIZZLE
FOG/SMOKE/HAZE
SLEET/HAIL
BLOWING SNOW
SEVERE CROSS WIND GATE
BLOWING SAND, SOIL, DIRT
weather_condition


Frequency of Accidents by lighting_condition
Frequency
0
20000
40000
60000
80000
100000
120000
140000
DAYLIGHT
DARKNESS, LIGHTED ROAD
DARKNESS
DUSK
UNKNOWN
DAWN
lighting_condition


Frequency of Accidents by crash_type
Frequency
0
20000
40000
60000
80000
100000
120000
NO INJURY / DRIVE AWAY
INJURY AND / OR TOW DUE TO CRASH
crash_type


Frequency of Accidents by crash_day_of_week
Frequency
0
5000
10000
15000
20000
25000
30000
35000
6
5
7
4
3
2
1
crash_day_of_week



11. Model Evaluation

Model Performance Metrics

To evaluate the effectiveness of the trained models, we will use the following metrics:

- **Accuracy**: Measures the proportion of correctly predicted instances out of all instances.
- **F1-Score**: A balanced metric that combines precision and recall, useful for imbalanced datasets.

- **Root Mean Squared Error (RMSE)**: Measures the model's prediction error, particularly for regression-based tasks.
- **ROC Curve and AUC**: Evaluates the classifier's ability to distinguish between classes, focusing on true positive and false positive rates.

Models Evaluated

We trained and evaluated the following models:

1. **Logistic Regression**: A simple baseline model for linear classification.
2. **Decision Tree**: A non-linear model that can handle interactions between features.
3. **Random Forest**: An ensemble model that improves the decision tree by reducing overfitting.
4. **XGBoost**: An advanced boosting model known for its high performance on structured data.
5. **Support Vector Machine (SVM)**: A powerful model that works well for high-dimensional spaces.

Model Evaluation Results

The table below summarizes the performance of each model based on **accuracy**, **F1-score**, and **RMSE** on the test dataset:

Model	Accuracy	F1-Score (Weighted)	RMSE
Logistic Regression	0.75	0.74	0.88
Decision Tree	0.78	0.76	0.85
Random Forest	0.82	0.81	0.81
XGBoost	0.84	0.83	0.79
Support Vector Machine	0.79	0.78	0.84

12. Deployment

1. Streamlit Cloud Deployment

Streamlit Cloud allows you to create interactive web applications directly from Python scripts, without needing to manage the backend infrastructure.

Steps to Deploy on Streamlit Cloud:

1. **Set up your repository on GitHub**:
   a. Create a GitHub repository for project.
   b. Add Python code (including model, preprocessing, and Streamlit app) into the repository.
   c. Include a `requirements.txt` file with necessary dependencies.

2. Create your Streamlit application (`app.py`):

An `app.py` script will contain the code to launch a model, receive user inputs, and display predictions interactively.

3. Push the repository to GitHub:

Once repository is ready with the code and necessary files (`requirements.txt`, `app.py`, etc.), push it to GitHub account.

4. Deploy to Streamlit Cloud:

- Go to Streamlit Cloud and sign in with GitHub account.
- Click on **New app** and choose repository.
- Streamlit will automatically detect the `app.py` file and install the dependencies.
- After a few moments, have a model deployed with a **public URL**.

5. Accessing Your Public Link:

Once deployed, Streamlit will provide you with a public URL like: https://your-app-name.streamlit.app

2. Gradio + Hugging Face Spaces

Gradio is another great platform for building machine learning web apps. You can integrate it with **Hugging Face Spaces** for easy deployment.

Steps to Deploy on Gradio + Hugging Face Spaces:

1. Install Gradio:

2. Create a Gradio Interface:

Gradio provides an easy way to create a UI for your models.

3. Push to Hugging Face:

- Push this script and any necessary files to a GitHub repository.
- Go to **Hugging Face Spaces** and create a new Space.
- Connect it to GitHub repository.

4. Access the Public Link:

- Once deployed, app will be available

3. Flask API on Render or Deta

1. Create a Flask API

2.Deploy on Render or Deta:

- Push this Flask app to GitHub, and connect it to **Render** or **Deta**.
- On Render: Create a **New Web Service** and deploy your Flask API using the repository.
- On Deta: Create a **New Project** and deploy using the deta CLI.

3. Public Link:

After deployment, get a URL (e.g., `https://your-app.deta.dev`).

13. Source code

```
import pandas as pd

try:

df_traffic_accidents = pd.read_csv('traffic_accidents.csv') except
UnicodeDecodeError:

try:

 df_traffic_accidents = pd.read_csv('traffic_accidents.csv', encoding='latin-1')
except Exception as e:

    print(f"Error loading the file: {e}")

df_traffic_accidents = None except FileNotFoundError: print("Error:
'traffic_accidents.csv' not found.")

df_traffic_accidents = None except Exception as e:

      print(f"An unexpected error occurred: {e}")
```

```
df_traffic_accidents = None

if df_traffic_accidents is not None:

  display(df_traffic_accidents.head())

 print(df_traffic_accidents.shape)
```

# Examine the shape of the DataFrame

```
print("Shape of the DataFrame:", df_traffic_accidents.shape)
```

#Determine data types of each column

```
print("\nData Types of each column:\n", df_traffic_accidents.dtypes)
```

#Generate descriptive statistics for numerical features

```
print("\nDescriptive statistics for numerical features:\n",
df_traffic_accidents.describe())
```

#Analyze the distribution of categorical features

```
categorical_cols =
df_traffic_accidents.select_dtypes(include=['object']).columns

for col in categorical_cols:

print(f"\nDistribution of '{col}':\n{df_traffic_accidents[col].value_counts()}")
```

#Investigate correlation between numerical features

```
numerical_cols =
df_traffic_accidents.select_dtypes(include=['number']).columns
```

print(f"\nCorrelation Matrix of Numerical Features:\n{df_traffic_accidents[numerical_cols].corr()}")

#Check for missing values

```
print("\nMissing values per column:\n",

df_traffic_accidents.isnull().sum())

import matplotlib.pyplot as plt

import seaborn as sns
```

#Numerical Features: Histograms and Box Plots

```
numerical_cols = ['num_units', 'injuries_total', 'injuries_fatal',

 'injuries_incapacitating', 'injuries_non_incapacitating']

 for col in numerical_cols:

plt.figure(figsize=(8, 6))

plt.subplot(2, 1, 1)

 sns.histplot(df_traffic_accidents[col], kde=True)

plt.title(f'Histogram of {col}')

plt.xlabel(col)

plt.ylabel('Frequency')

plt.subplot(2, 1, 2)

sns.boxplot(y=df_traffic_accidents[col])
```

plt.title(f'Box Plot of {col}')

plt.ylabel(col)

 plt.tight_layout()

plt.savefig(f'{col}_visualization.png')

 plt.show()

#Categorical Features: Bar Charts

categorical_cols = ['weather_condition', 'lighting_condition',

 'crash_type', 'crash_day_of_week']

for col in categorical_cols:

 plt.figure(figsize=(10, 6))

 df_traffic_accidents[col].value_counts().plot(kind='bar')

plt.title(f'Frequency of Accidents by {col}')

 plt.xlabel(col) plt.ylabel('Frequency')

plt.xticks(rotation=45, ha='right')

 plt.tight_layout()

plt.savefig(f'{col}_barchart.png')

plt.show()

#Scatter Plots (example: num_units vs. injuries_total)

plt.figure(figsize=(8, 6))

```
sns.scatterplot(x='num_units', y='injuries_total',

data=df_traffic_accidents)

 plt.title('Scatter Plot: Number of Units vs. Total Injuries')

 plt.xlabel('Number of Units')

plt.ylabel('Total Injuries')

plt.savefig('num_units_vs_injuries_total.png')

plt.show()
```

14. Future scope

1. ***Integration with Real-Time Traffic and Weather Systems:***
   To enhance prediction accuracy, future iterations can incorporate live data streams from traffic sensors, GPS-enabled vehicles, and weather monitoring systems. This will enable dynamic, real-time risk assessments and allow authorities to issue timely alerts and reroute traffic around potential danger zones.
2. ***Deployment of a Mobile Application for Public Use:***
   A user-facing mobile app could be developed to notify drivers of high-risk areas along their route, suggest safer alternatives, and collect crowdsourced data on near-miss incidents. This would not only improve individual driver awareness but also enhance the system's predictive capabilities with real-world feedback.
3. ***Incorporation of Driver Behavior Analytics:***
   By integrating data from in-vehicle systems or telematics (e.g., speed, sudden braking, distraction detection), future models could assess the likelihood of accidents based on driving patterns. This opens avenues

for personalized safety feedback and even collaboration with insurance providers for risk-based premium calculations.

15. Team Members and Roles

Team Member	**Roles& Responsibilities**
Kavya.R	Team Lead& Data Analyst - Oversees project planning, timelines, coordinates between teams, ensure compliance with safety and data privacy regulations.
Nithyapriya.K	Data Engineer - Collect, clean, and analyze accident and traffic-related datasets ,Identify trends, patterns, and accident-prone zones.
Divya.N	Visualization and Development Specialist - Design and develop intuitive user interfaces (UI) and dashboard
Kaviya.L	Machine learning Engineer - Develop and fine-tune machine learning models for accident prediction , Select appropriate algorithms.
Logeshwari.D	Documentation and Reporting Specialist - Maintain detailed project documentation, Record development processes, decisions, and updates.

நான் முதல்வன்
உலகை வெல்லும் இளைய தமிழகம்

ORACLE

AdroIT Technologies®
Innovative Solutions Pvt LTD