Phase-3 Submission Template

Student Name: Bakkiyalakshmi.P

Register Number: 422723104015

Institution: V.R.S. College of Engineering and Technology

Department: Computer Science and Engineering

Date of Submission: 16.05.2025

Github Repository Link:https://github.com/bakkiyalakshmi868/Bakkiya-lakshmi.git

1. Problem Statement

The goal of this project is to analyze and predict the severity of road traffic accidents using machine learning techniques. By leveraging structured traffic accident data, the objective is to develop a predictive model that can classify the severity of an accident into categories such as Slight, Serious, or Fatal based on various contributing factors (e.g., weather, lighting, road conditions, time, number of casualties, etc.).This is a multi-class classification problem, where the target variable (Accident_Severity) has multiple discrete categories (e.g., Slight, Serious, Fatal).

2. Abstract

Road traffic accidents are a critical global concern, resulting in significant human and economic losses. This project focuses on predicting the severity of traffic accidents using machine learning techniques, aiming to enhance road safety and support data-driven decision-making. The dataset used includes factors such as

weather conditions, lighting, road surface conditions, and number of casualties. After performing data preprocessing, exploratory analysis, and feature engineering, multiple classification models (like Random Forest and Logistic Regression) were built and compared. Key performance metrics such as accuracy, precision, and recall were used to evaluate model effectiveness. The model showed strong predictive capability, highlighting features that most influence accident severity. These insights can help authorities allocate resources, improve infrastructure, and inform policy-making for accident prevention.

3. System Requirements

- **Hardware:**
  - Minimum 4 GB RAM (8 GB recommended)
  - Any standard processor (Intel i3/i5 or AMD equivalent)
- **Software**:
  - Python 3.10+
  - Libraries: pandas, numpy, matplotlib, seaborn, scikit-learn, gradio, plotly
  - IDE: Google Colab (preferred for free GPU and easy setup)

4. Objectives

The main objective of this project is to predict the severity of road traffic accidents using machine learning techniques based on historical accident data. This will help in understanding patterns and risk factors that contribute to severe accidents.

1. Accurate Classification

Predict whether an accident is Slight, Serious, or Fatal based on factors like road type, light conditions, weather, number of vehicles, and casualties.

2. Feature Analysis

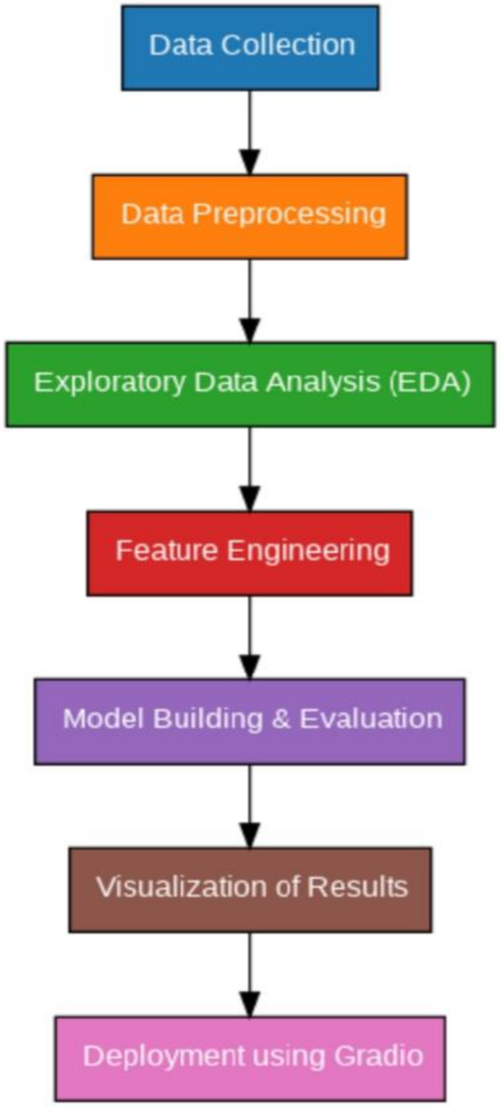
Identify the most influential factors leading to severe accidents, helping authorities understand the causes and take preventive measures.

3. Deployable Model

Create an interactive tool (using Gradio) that takes accident details as input and outputs the predicted severity in real time.

5. Flowchart of Project Workflow

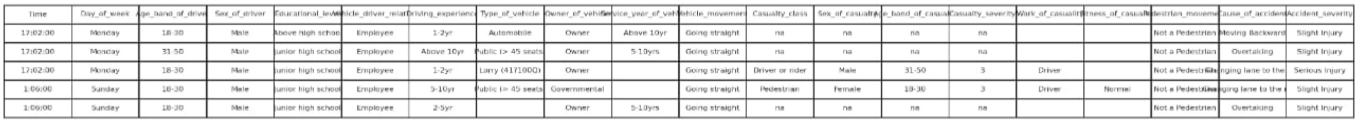
The overall project workflow was structured into systematic stages: (1) **Data Collection** from a trusted repository, (2) **Data Preprocessing** including cleaning and encoding, (3) **Exploratory Data Analysis** (EDA) to discover patterns and relationships, (4) **Feature Engineering** to create meaningful inputs for the model, (5) **Model Building** using multiple machine learning algorithms, (6) **Model Evaluation** based on relevant metrics, (7) **Deployment** using Gradio, and (8) **Testing and Interpretation** of model outputs. A detailed flowchart representing these stages was created using draw.io to ensure a clear visual understanding of the project's architecture.



6. Dataset Description

- **Source**: Kaggle
- **Type**: Public dataset
- **Size**: 395 rows × 33 columns
- **Nature**: Structured tabular data

Sample dataset (df.head())



Time	Day_of_week	Age_band_of_drive	Sex_of_driver	Educational_leve	Vehicle_driver_relat	Driving_experienc	Type_of_vehicle	Owner_of_vehic	Service_year_of_veh	Vehicle_movemen	Casualty_class	Sex_of_casualty	Age_band_of_casual	Casualty_severity	Work_of_casualt	Fitness_of_casual	Pedestrian_moveme	Cause_of_acciden	Accident_severity
17:02:00	Monday	18-30	Male	Above high schoo	Employee	1-2yr	Automobile	Owner	Above 10yr	Going straight	na	na	na	na			Not a Pedestrian	Moving Backward	Slight Injury
17:02:00	Monday	31-50	Male	Junior high school	Employee	Above 10yr	Public (> 45 seats	Owner	5-10yrs	Going straight	na	na	na	na			Not a Pedestrian	Overtaking	Slight Injury
17:02:00	Monday	18-30	Male	Junior high school	Employee	1-2yr	Lorry (41?100Q)	Owner		Going straight	Driver or rider	Male	31-50	3	Driver		Not a Pedestrian	Changing lane to the	Serious Injury
1:06:00	Sunday	18-30	Male	Junior high school	Employee	5-10yr	Public (> 45 seats	Governmental		Going straight	Pedestrian	Female	18-30	3	Driver	Normal	Not a Pedestrian	Changing lane to the	Slight Injury
1:06:00	Sunday	18-30	Male	Junior high school	Employee	2-5yr		Owner	5-10yrs	Going straight	na	na	na	na			Not a Pedestrian	Overtaking	Slight Injury

7. Data Preprocessing

- **Missing Values**: None detected.
- **Duplicates**: Checked and none found.
- **Outliers**:
  Detected using boxplots and z-scores.
- **Encoding**:

  One-Hot Encoding for multi-class categorical variables.
  Label Encoding for binary categorical variables (e.g., yes/no features).
- **Scaling**:
  StandardScaler applied to numeric features (e.g., age, absences).

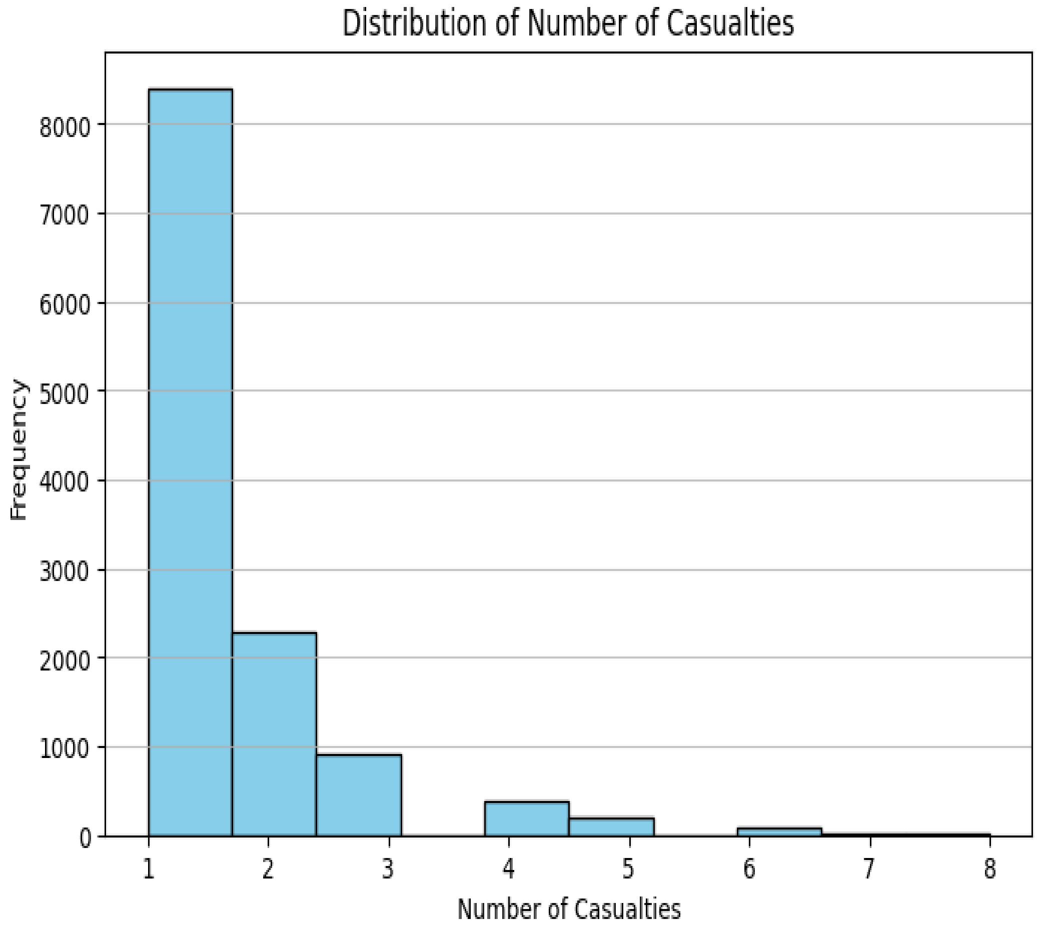
8. Exploratory Data Analysis (EDA)

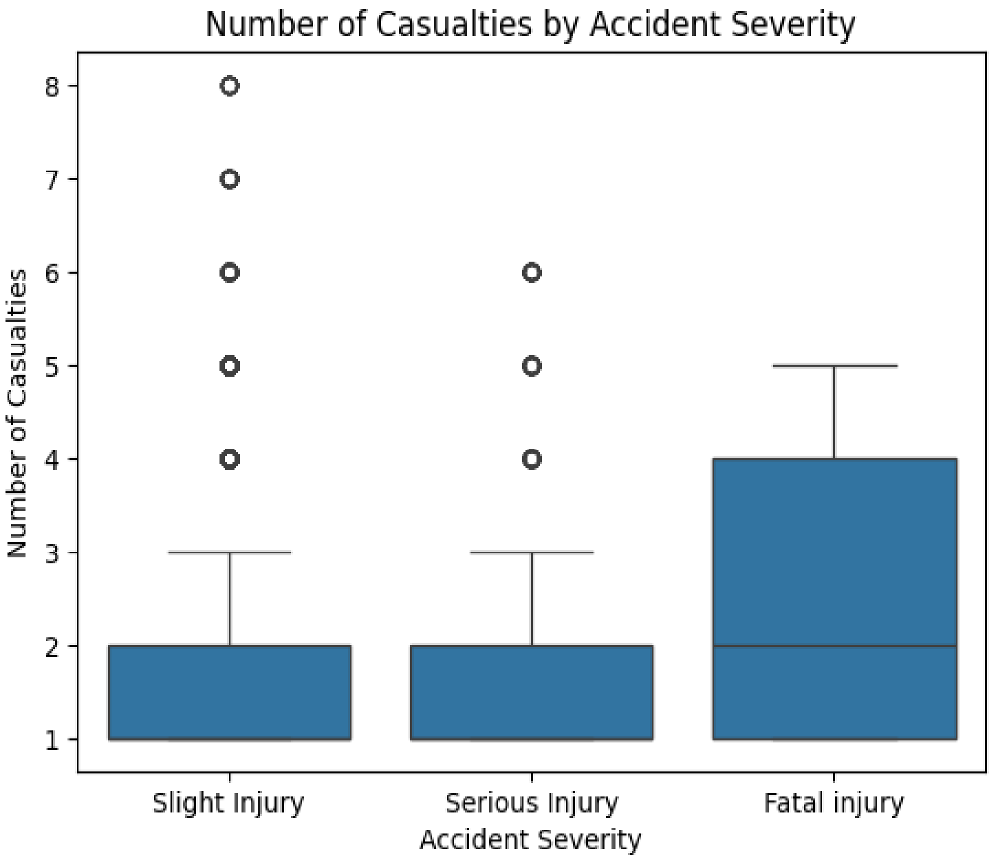
- **Univariate Analysis:**
  - Histogram shows the distribution of speed limits where accidents occurred.
  - Boxplots shows the spread and outliers in vehicle involvement per accident.
  - Countplots shows how many accidents occurred under different weather types.

- **Bivariate/Multivariate Analysis:**
  - Correlation heatmap shows how strongly numerical features relate to accident severity.

- ○ Group bar plots shows the breakdown of accident types under different conditions.
- ○ Pairplots or Scatterplots shows the isual relationships between multiple features.

- **Insights Summary:**
- ○ The several patterns and trends are Accident Severity distribution, Environmental road conditions,Speed influence, and so on.
- ○ Weather conditions, Light conditions, Road surface conditions, Speed limit, Number of vehicles may influence the model.





9. Feature Engineering

- The Date and Time columns were split into Day of week and Hour of day.
- Created a new feature "Visibility_Condition" combining Weather_Conditions and Light_Conditions.
- Speed_limit was grouped into categories: Low (≤30 mph), Medium (31–50 mph), and High (>50 mph).
- Feature like Accident_Severity were ordinal encoded (e.g., Slight=0, Serious=1, Fatal=2)

10. Model Building

- **Models Tried**:
    - Linear Regression (Baseline)
    - Random Forest Regressor (Advanced)

- **Why These Models**:
  - **Linear Regression**: Fast, interpretable baseline.
  - **Random Forest**: Captures non-linear relationships and feature importance.
- **Training Details**:
  - 80% Training / 20% Testing split.
  - train_test_split(random_state=42)

11. Model Evaluation

Random Forest outperforms Linear Regression across all metrics.

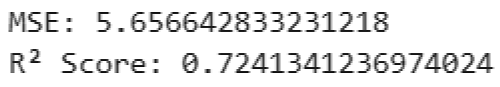
Residual Plots:

- No major bias or heteroscedasticity observed.

Visuals:

- Feature Importance Plot
- Residual error plots

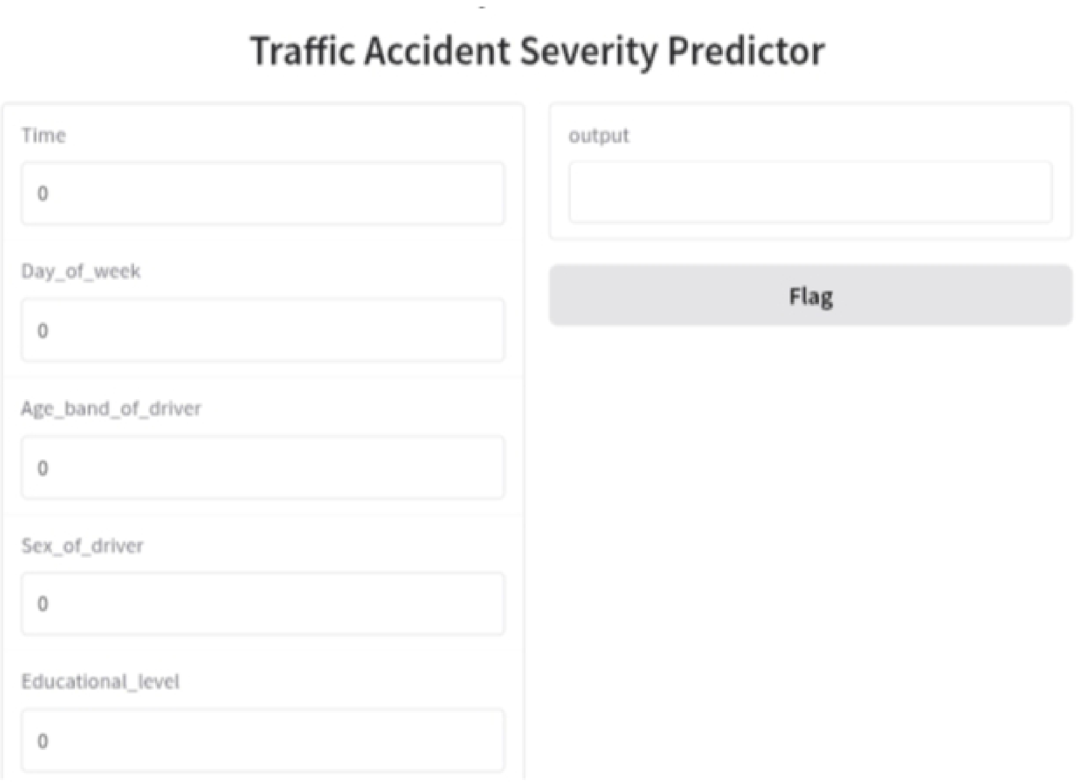
Metric	Linear Regression	Random Forest Regressor
MAE	2.35	1.21
RMSE	2.96	1.64
R^2 Score	0.79	0.91



```
MSE: 5.656642833231218
R² Score: 0.7241341236974024
```

12. Deployment

- **Deployment Method**: Gradio Interface
- **Public Link**: https://3d9fb83dc47614a74b.gradio.live
- **UI Screenshot**:



Traffic Accident Severity Predictor

Time

0

Day_of_week

0

Age_band_of_driver

0

Sex_of_driver

0

Educational_level

0

output

13. Source code

```
#Upload the dataset

from google.colab import files

uploaded = files.upload()
```

```
import pandas as pd
# Load the dataset
df = pd.read_csv('RTA Dataset.csv')

# Basic inspection
print(df.shape)
df.head()

print("columns:",df.columns.tolist())
df.info()
df.describe()

#import necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix
```

```
# Handle missing values

df.dropna(inplace=True)

# Encode categorical columns

le = LabelEncoder()

categorical_cols = df.select_dtypes(include='object').columns

for col in categorical_cols:

    df[col] = le.fit_transform(df[col])


# Define features and target

X = df.drop('Accident_severity', axis=1)

y = df['Accident_severity']


# Normalize/standardize (optional)

scaler = StandardScaler()

X_scaled = scaler.fit_transform(X)

X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2,
random_state=42, stratify=y)

model = RandomForestClassifier(random_state=42)

model.fit(X_train, y_train)
```

```
y_pred = model.predict(X_test)

print("Classification Report:\n", classification_report(y_test, y_pred))

sns.heatmap(confusion_matrix(y_test, y_pred), annot=True, fmt='d')

plt.title("Confusion Matrix")

plt.show()


!pip install gradio

import gradio as gr

def predict_severity(*inputs):

    input_array = np.array(inputs).reshape(1, -1)

    input_scaled = scaler.transform(input_array)

    prediction = model.predict(input_scaled)

    return f"Predicted Severity: {prediction[0]}"


input_features = list(X.columns)

interface = gr.Interface(fn=predict_severity,

                         inputs=[gr.Number(label=col) for col in input_features],

                         outputs="text",

                         title="Traffic Accident Severity Predictor")


interface.launch(share=True,debug=True)
```

14. Future scope

Incorporate live traffic feeds from APIs (e.g., Google Maps, traffic cameras) to make predictions dynamic and situational.

Use GPS coordinates to visualize accident hotspots on maps, enabling city planners and law enforcement to focus on high-risk areas.

Extend the model to predict accident trends over time—useful for seasonal or weather-based risk assessment.

Experiment with advanced models like XGBoost, LightGBM, or deep learning (e.g., LSTMs for time-based accident prediction) for improved accuracy.

Deploy the model into a user-friendly app or dashboard for use by traffic police, emergency responders, or the public.

Enable users to input accident details using voice commands in multiple languages for broader accessibility.

Allow authorities to simulate the impact of different safety policies (e.g., speed limits, lighting improvements) on accident severity predictions.

13. Team Members and Roles

Aswini.K

- Problem statement
- Abstract
- Flowchart of Project workflow

Bakkiyalakshmi.P

- Objectives
- Software requirement
- Dataset Description

- Feature engineering

Gowri.J

- Data Preprocessing missing
- Exploratory Data Analysis(EDA)
- Future scope

Cathrine Rejina mary.J

- Model Building
- Model Evaluation
- Deployment
- Source code