Accident Severity Prediction

# 1. Project Overview

Road accidents are a serious global issue causing fatalities, injuries, and economic losses. This project aims to predict accident severity (Slight, Serious, Fatal) using machine learning models trained on traffic accident datasets.

## Motivation

- Human Impact: Early prediction helps prioritize emergency response and allocate resources effectively.  
- Operational Efficiency: Enables EMS and police to triage faster and respond optimally.  
- Policy & Prevention: Identifies road, vehicle, and environmental risk factors to guide safety measures.

# 2. Dataset Details

## Primary Dataset

Name: Addis Ababa City Road Traffic Accident Severity Dataset (Figshare)  
Records: 1,000+ accident cases  
Target Variable: Accident Severity (Slight, Serious, Fatal)  
Key Predictors:  
 - Number of Vehicles  
 - Number of Casualties  
 - Road Type  
 - Weather Conditions  
 - Light Conditions  
 - Speed Limit  
 - Day, Time  
 - Latitude, Longitude

## Quality & Characteristics

- Source: Official government dataset → reliable and verified  
- Challenges:  
 - Missing values in Weather condition and Road surface type  
 - Class imbalance (Slight >> Serious >> Fatal)  
 - Outliers in casualty counts  
- Course Fit: Tabular dataset, >6 features, >1,000 rows (ideal for ML tasks)

## Alternative Datasets

- UK Road Safety Accidents (2005–2017) – 1.8M records  
- US Accidents (2016–2023) – 7.7M records

# 3. Preprocessing Pipeline

• Handling Missing Data: Imputation: Categorical → Mode, Numeric → Mean

• Encoding Categorical Variables: Target variable → Label Encoding; Predictors → One-Hot Encoding

• Outlier Removal: IQR-based filtering for casualties and vehicles

• Normalization / Scaling: Applied MinMaxScaler to bring numeric features to [0,1]

• Feature Selection: Correlation analysis & variance threshold; dropped weak predictors

• Dimensionality Reduction (PCA): Applied PCA; first 3 components explained ~85% variance

# 4.Group Members and Roles

|  |  |  |
| --- | --- | --- |
| **IT number** | Name | Preprocessing Technique |
| IT24102810 | Fernando W.M.S. | Handling Missing Data |
| IT24102911 | Ravihansi L. U. M. | Encoding Categorical Variables |
| IT24102509 | Hettiarachchi T. J. | Outlier Removal |
| IT24102362 | Teshan Mathintha S.L.A | Normalization / Scaling |
| IT24102603 | Rajapaksha K. M. P. | Feature Selection |
| IT24102275 | Wijayananda W. J. W. | Dimensionality Reduction (PCA) |

# 5. How to Run the Code

## Requirements

Python 3.8+  
Libraries: pandas, numpy, matplotlib, seaborn, scikit-learn

## Steps

1. Load the dataset:  
 import pandas as pd  
 df = pd.read\_csv('accidents.csv')  
  
2. Preprocessing (apply sequentially: missing values → encoding → outliers → scaling → feature selection → PCA)  
  
3. Train-Test Split:  
 from sklearn.model\_selection import train\_test\_split  
 X = df.drop('Accident\_severity', axis=1)  
 y = df['Accident\_severity']  
 X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)  
  
4. Model Training (example: Random Forest):  
 from sklearn.ensemble import RandomForestClassifier  
 model = RandomForestClassifier()  
 model.fit(X\_train, y\_train)  
 y\_pred = model.predict(X\_test)  
  
5. Evaluation: Use accuracy, precision, recall, F1-score; apply SMOTE/class weighting for imbalance if needed.