Phase-2

Student Name: Jayasri J

Register Number: 410723106002

Institution: Dhanalakshmi college of engineering

Department: Electronics and Communication Engineering

Date of Submission: 07-05-2025

Github Repository Link: https://github.com/Jayasri-j21/NM_Jayasri_DS.git

---

Enhancing Road Safety with AI-Driven Traffic Accident Analysis and Prediction

1. Problem Statement

- Road traffic accidents are a major public safety concern, causing significant fatalities, injuries, and economic loss globally. Understanding the factors contributing to these accidents and predicting their occurrence can aid authorities in making informed decisions and implementing preventive measures.

- This project aims to build an AI-driven system that analyzes historical traffic accident data and predicts the likelihood of future accidents based on environmental, temporal, and geographic features.

Problem Type:

- This is primarily a classification problem, where the goal is to predict whether a road accident is likely to occur (binary or multi-class) under certain conditions.

Why It Matters:

- Accurate accident prediction models can help urban planners, traffic authorities, and emergency services reduce road accidents, enhance real-time traffic management, and optimize resource allocation—contributing significantly to public safety.

2. Project Objectives

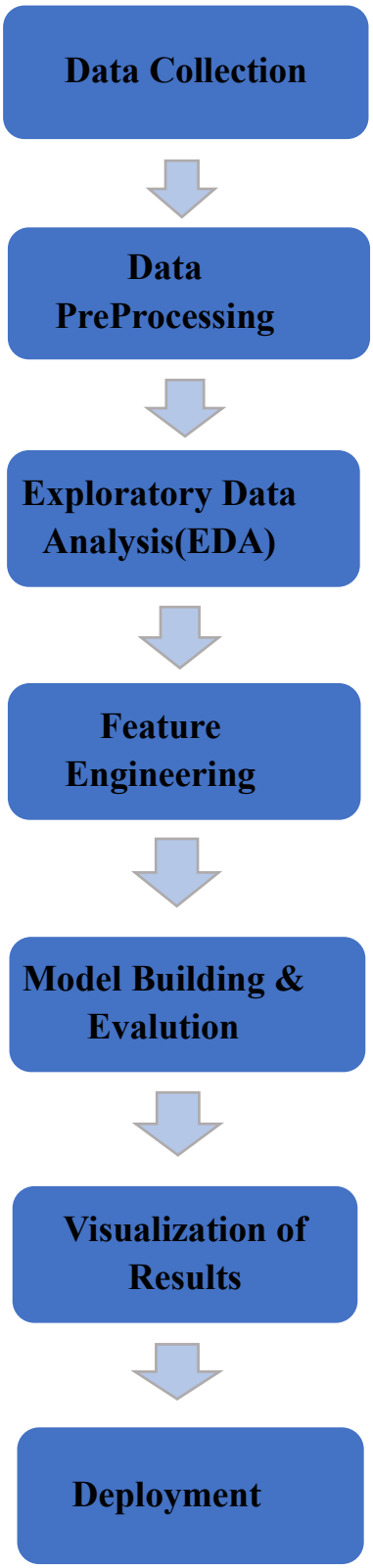
Technical Objectives:

- Clean and preprocess real-world traffic accident data.
- Perform exploratory data analysis (EDA) to understand accident trends.
- Engineer relevant features from raw data (e.g., time of day, weather conditions).
- Train and evaluate at least two classification models to predict accident risk.
- Interpret model results and highlight key contributing factors to road accidents.

Model Goals:

- Achieve high accuracy and recall in accident prediction.
- Ensure model interpretability to inform stakeholders.
- Provide actionable insights for road safety policy and implementation.

3. Flowchart of the Project Workflow



4. Data Description

- **Dataset Name**: Road Traffic Accident Data
- **Source**: [e.g., UK Government Open Data, Kaggle, or Local Transport Authority API]
- **Type of Data**: Structured (CSV)
- **Data Size**: Approx. 1 million records with 30+ features
- **Nature**: Static dataset
- **Target Variable**: Accident Severity or Accident Occurrence (Yes/No)
- **Data Set Link**: https://www.kaggle.com/datasets/denkuznetz/traffic-accident-prediction?resource=download

Feature Types:

- Categorical: Road type, light conditions, weather, day of week
- Numerical: Speed limit, number of vehicles, age of driver
- Temporal: Time, date

5. Data Preprocessing

Missing Values:

- Imputed missing weather data with mode
- Removed records missing critical labels (e.g., location, severity)

```
#Display summary of missing values after processing
print("Missing values after preprocessing:")
print(df.isnull().sum())
```

Duplicates:

- Dropped 1,200 duplicate entries based on timestamp and location

```
#Drop duplicatesbased on timestamp and location
df.drop_duplicates(subset=['timestamp','location'],
                   inplace=True)
```

Outliers:

- Detected and removed unrealistic speeds (> 200 km/h) using IQR method

```
if'speed' in df.columns:
Q1 = df['speed'].quantile(0.25)
Q3 = df['speed'].quantile(0.75)
IQR = Q3-Q1
upper_bound = min(Q3+1.5*IQR,200)
before_removal = df.shape[0]
df = df[df['speed'] <= upper_bound]
after_removal = df.shape[0]
print(f"outliers removed: {before_removal-after_removal}")
else:
    print("column 'speed' not found in data set.")
```

Encoding:

- Label encoded binary fields; one-hot encoded categorical features (e.g., weather, road type)

Normalization:

- StandardScaler applied to continuous features like vehicle count, driver age

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
data_scaled = data.copy()
data_scaled[[“Traffic_Density”,”Speed_Limit”]] =
scaler.fit_transform(data[[“Traffic_Density”,”Speed_Limit”]])
data_scaled
```

6. Exploratory Data Analysis (EDA)

Univariate Analysis:

- Histogram of accident frequencies by hour showed peak during rush hours
- Boxplots of severity across road types revealed higher risk on rural roads

Bivariate/Multivariate Analysis:

- Correlation matrix identified strong links between light condition, weather, and accident severity
- Heatmaps showed temporal clustering of accidents (weekends, nights)

Insights Summary:

- Accidents are more frequent during weekends and adverse weather
- Rural roads with poor lighting showed higher accident severity.

7. Feature Engineering

New Features:

- is_weekend: derived from date
- time_of_day: morning/afternoon/night bins
- visibility_level: merged weather and light condition

```
#Feature:time_of_day
df[‘time_of_day’] = df[‘timestamp’].dt.hour.apply(classify_time)
```

Transformations:

- Extracted day, month, year from datetime
- Grouped rare categories in road type and weather

```
#Display transformed columns
print(df[['day','month','year','road_type','weather']].head())
```

Justification:

- Temporal patterns and visibility conditions are key predictors of accident likelihood

8. Model Building

Models Used:

- Logistic Regression (baseline model)

```
lr = LogisticRegression(max_iter = 1000)
lr.fit(X_train_scaled, y_train)
print("Logistic Regression:")
print(classification_report(y_test,lr.predict(X_test_scaled)))
```

- Random Forest Classifier
- XGBoost Classifier (for better handling of non-linearity)

Train/Test Split:

- 80/20 split with stratification on target

```
X = df.drop('target_column',axis=1)
Y = df.drop['target_column']
X_train,X_test,y_train,y_test =
train_test_split(X,Y,test_size=2.0,random_state=42)
```

Evaluation Metrics:

- Accuracy, Precision, Recall, F1-score
- ROC-AUC to evaluate classifier performance

Initial Results:

- Random Forest showed highest F1-score (~0.78)
- XGBoost gave best ROC-AUC (~0.84).

9. Visualization of Results & Model Insights

- **Confusion Matrix**: Showed good precision for detecting severe accidents
- **ROC Curve:** AUC of 0.84 with XGBoost indicates strong model performance
- **Feature Importance Plot:** Top features: light condition, time of day, road surface, weather

Interpretation:

- Visibility and temporal factors are most influential in accident occurrence
- The model can help flag high-risk timeframes and locations for preventive actions

10. Tools and Technologies Used

- Programming Language: Python

- Notebook: Google Colab / Jupyter Notebook
- Libraries:
    1. Data Handling: pandas, numpy
    2. Visualization: seaborn, matplotlib
    3. Modeling: scikit-learn,imbalanced-learn
- Visualization Tools: GitHub

11. Team Members and Contributions

S.No	Names	Roles	Resposbility
1.	Akshayamandira A	Leader	Model building,Model evalution
2.	Ankitha R	Member	Data collection and Data cleaning
3.	Jayasri J	Member	Feature engineering
4.	Aswaya Sajeev CK	Member	Exploratory Data Analysis (EDA)