Github Link: **https://github.com/vishnuharichandran66/project-road-safety.git**

Project Title: Enhancing road safety with AI-driven traffic accident analysis and prediction

PHASE-2

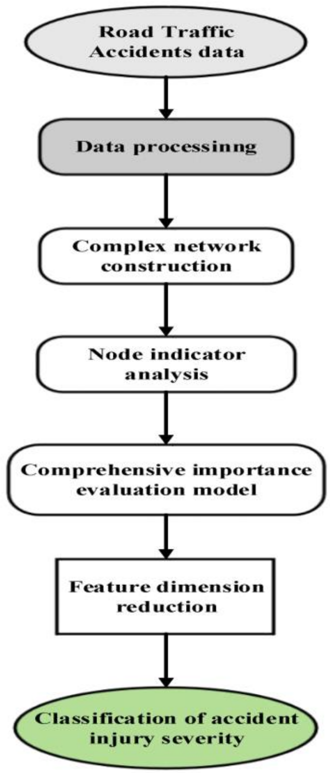
- **Problem Statement**

Despite advancements in transportation infrastructure and vehicular technology, traffic accidents remain a leading cause of injury and death worldwide. Traditional traffic management systems often react to incidents rather than proactively preventing them, resulting in delayed emergency response, increased congestion, and preventable fatalities. Current methods for analyzing accident causes rely heavily on manual reporting and retrospective analysis, which limits their effectiveness in real-time risk assessment and future accident prediction.

Project Objectives

- Develop an AI-based system to analyze historical and real-time traffic data for identifying patterns and causes of road accidents.
- Build predictive models to forecast accident-prone locations and times with high accuracy.
- Visualize risk zones through intuitive heatmaps and dashboards for better decision-making by authorities
- Provide real-time alerts to drivers and traffic managers to help prevent accidents before they occur
- Support policy and planning by offering data-driven insights for improving road safety infrastructure and strategies.

.Flowchart of the Project Workflow



Data Description

To develop an effective AI-based traffic accident analysis and prediction system, a diverse and comprehensive dataset is required. The project will utilize the following types of data:

Historical Accident Data

Source: Government transportation departments, traffic police records

Attributes: Date/time of accident, location (GPS coordinates), accident severity, number of vehicles involved, type of collision, casualties, etc.

Traffic Flow Data

Source: Road sensors, traffic cameras, GPS devices, smart traffic systems

Attributes: Vehicle count, average speed, congestion levels, time of day, lane usage

Weather Data

Source: Meteorological APIs (e.g., OpenWeather, government weather services)

Attributes: Temperature, precipitation, fog, visibility, wind speed, road surface condition

Road Infrastructure Data

Source: Urban planning databases, satellite imagery, GIS data

Attributes: Road type (highway, local street), number of lanes, intersection presence, lighting conditions, signage, speed limits

Real-time Data Streams (for implementation phase)

Source: IoT devices, vehicle telematics, traffic control centers

Attributes: Live traffic updates, ongoing incidents, road closures, emergency vehicle movements

Demographic and Geographic Data (Optional)

Source: Census data, local government databases

Attributes: Population density, land use type, nearby schools or commercial zones

Dataset Link:://www.kaggle.com/datasets/sahityasetu/motor-vehicle-collisions-crashes-usa

Data Preprocessing

Data Cleaning

Handle missing values, remove duplicates, and fix inconsistent formats.

Data Integration

Merge accident, traffic, weather, and road data into a unified dataset.

Feature Engineering

Extract useful features like time of day, accident frequency, or weather severity indicators.

Encoding and Normalization

Convert categorical variables into numeric format and scale numerical data.

Data Splitting

Split data into training, validation, and test sets for model development.

Exploratory Data Analysis (EDA)

Univariate Analysis

Analyze distributions of individual variables (e.g., accident severity, time of day, weather conditions).

Bivariate and Multivariate Analysis

Explore relationships between features (e.g., accidents vs. speed, accidents vs. weather).

Time-Series Analysis

Examine accident trends over time (hourly, daily, seasonal patterns).

Geospatial Analysis

Use maps and heatmaps to identify high-risk accident zones.

Correlation Analysis

Identify relationships between numeric variables to assist with feature selection.

Feature Engineering

Time-Based Features

Extract hour of day, day of week, month, and holiday indicators from timestamps.

Location-Based Features

Encode accident locations as GPS clusters or distance to nearest intersection/school zone.

Weather-Risk Indicators

Create combined features like "rainy and low visibility" or "wet road and high speed."

Traffic Volume Ratios

Calculate vehicle count per lane or speed-to-flow ratios during accident times.

Severity Encoding

Convert categorical severity levels into ordinal values for modeling.

Model Bulding

Model Selection

Choose appropriate algorithms (e.g., Random Forest, XGBoost, Logistic Regression, or Neural Networks) based on the problem type (classification/regression).

Training and Validation

Train models using the preprocessed data and evaluate performance using cross-validation techniques.

Hyperparameter Tuning

Optimize model performance using grid search, random search, or Bayesian optimization.

Handling Class Imbalance

Apply techniques like SMOTE, undersampling, or class weighting if accident severity classes are imbalanced.

Model Evaluation

Use metrics such as accuracy, precision, recall, F1-score, and ROC-AUC to assess model performance.

Visualization of Results & Model Insights

- **Confusion Matrix and ROC Curves**

- Visualize model performance, especially for classification tasks like accident severity prediction.
- **Feature Importance Plot**
- Display which features (e.g., time, weather, speed) had the most influence on predictions.
- **Accident Heatmaps**
- Show high-risk areas using geospatial heatmaps for better spatial understanding.
- **Time-Series Plots**
- Visualize accident trends over time (e.g., daily, weekly, seasonal patterns).
- **Partial Dependence or SHAP Plots**
- Interpret how individual features affect the prediction outcomes.

`Tools and Technologies Used

- **Python**
- Main programming language for data analysis, model development, and visualization.
- **Pandas & NumPy**
- Data manipulation and numerical computations.
- **Scikit-Learn**
- Machine learning library for building and evaluating predictive models.
- **TensorFlow / Keras**
- Deep learning frameworks for advanced modeling (e.g., neural networks).
- **Matplotlib & Seaborn**
- Data visualization libraries for plotting graphs and charts.
- **GeoPandas & Folium**
- Tools for geospatial data processing and map-based visualizations.
- **Jupyter Notebook**
- Interactive environment for code development, testing, and visualization.

Team Members and Contributions

1. M.VISHNU HARICHANDRAN - Data Scientist

- Responsible for data collection, preprocessing, and feature engineering.
- Developed and tuned predictive models using machine learning algorithms

2. D.TAMILMANI - Software Engineer

- Built the system architecture and integrated various data sources.
- Developed the real-time alert system for accident prediction and prevention.

3.R.VITHISH - Data Analyst

- Conducted exploratory data analysis (EDA) to identify key patterns and trends.
- Created visualizations for insights into accident locations, weather conditions, and traffic data.

4. C.SANJAY - Project Manager

- Oversaw the project timeline, ensured task completion, and coordinated team activities.
- Ensured communication with stakeholders and handled documentation.