Enhancing road safety with AI driven traffic accident analysis and prediction

Student Name: PERAMALINGAM. P
Register Number:732323106038
Institution:SSM COLLEGE OF ENGINEERING
Department: BE(ECE)-2YEAR
Date of Submission:29:04:2025
GitHub Repository Link:https://github.com/peramalingam007/Naanmudhalvan/blob/main/Enhancing%20road%20safety%20with%20AI%20driven%20traffic%20accident%20analysis%20and%20prediction.py

1. Problem Statement:

Road traffic accidents remain one of the leading causes of injury and death worldwide, often resulting from a combination of human error, environmental factors, and traffic congestion. Traditional methods of accident prevention rely heavily on historical data and reactive measures, which are often insufficient for timely intervention. There is a pressing need for intelligent systems capable of analyzing large volumes of traffic data in real time to identify patterns, assess risk factors, and predict potential accidents before they occur.

This project aims to enhance road safety by leveraging artificial intelligence (AI) to conduct comprehensive traffic accident analysis and prediction. By integrating AI-driven models with historical and real-time data, the system will provide valuable insights for proactive decision-making and the deployment of preventive measures, ultimately reducing the frequency and severity of road accidents.

2. Project Objectives:

1. To analyze historical traffic accident data using AI techniques to identify patterns and high-risk areas.

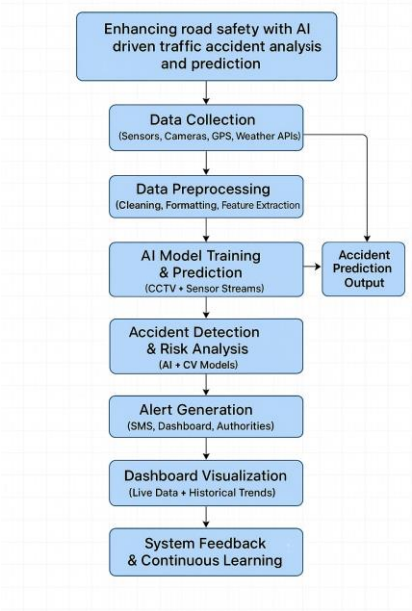
2. To develop a predictive model that forecasts potential traffic accidents based on real-time and historical data.

3. To enhance situational awareness for traffic management authorities through intelligent data visualization and alerts.

4. To reduce traffic-related injuries and fatalities by enabling proactive safety measures and decision-making.

5. To provide a scalable AI-based framework that can be integrated with existing traffic monitoring systems for broader applicability.

3. Flowchart of the Project Workflow:



4. Data Description :

1. Accident Metadata:
Date and time of the accident
Location coordinates (latitude and longitude)
Weather conditions (rain, fog, snow, clear, etc.)
Road type and condition (wet, dry, icy, under construction)
Light conditions (daylight, nighttime, street lighting)

2. Vehicle and Driver Information:
Number and types of vehicles involved
Vehicle speeds before the crash (if available)
Driver demographics (age, gender, driving experience)

Presence of alcohol or drugs (if reported)

3. Environmental and Traffic Factors:
Traffic density at the time of the accident
Nearby traffic signals or signs
Urban vs. rural setting

4. Accident Outcomes:
Severity level (minor, major, fatal)
Number of injuries and fatalities
Emergency response time

5. Historical Trends (for predictive modeling):
Accident frequency by location and time
Seasonal or day-of-week patterns
Repeat accident zones (hotspots)
The data is preprocessed using cleaning, normalization, and feature engineering techniques to handle missing values, remove duplicates, and convert categorical data into model-friendly formats. The final dataset serves as the foundation for training machine learning models aimed at predicting the likelihood and severity of accidents based on current traffic and environmental conditions.
Data link: https://www.kaggle.com/datasets/sobhanmoosavi/us-accidents

5. Data Preprocessing:

1. Data Collection and Integration:
Gather data from multiple sources (e.g., government traffic records, weather APIs, GPS logs, traffic cameras).
Merge datasets based on common fields like timestamps and geolocations to create a unified dataset.

2. Handling Missing Values:
Identify missing or incomplete entries using data profiling.
Impute missing numerical values using mean, median, or k-NN imputation.
For categorical values, use mode imputation or introduce a new category like “Unknown.”

3. Data Cleaning:
Remove duplicate entries and irrelevant features (e.g., unrecorded or unknown fields with excessive missing values).
Correct inconsistent formatting (e.g., date formats, unit conversions for speed and distance).
Normalize text entries for categorical fields (e.g., “rainy,” “Rainy,” “RAINY” converted to "Rainy").

4. Feature Encoding:
Convert categorical variables (e.g., weather conditions, road type, light conditions) into numerical format using:
One-hot encoding for non-ordinal features.
Label encoding for ordinal features.

5. Feature Engineering:
Create new features such as:
Time of day (morning, afternoon, night)
Day of week, holiday indicator
Accident hotspot zones using clustering of locations
Combine relevant variables (e.g., combining temperature and precipitation into a "weather severity" score).

6. Outlier Detection and Treatment:
Use statistical methods (z-score, IQR) or visualization tools (box plots) to identify outliers.
Decide on treatment: removal, capping, or transformation depending on the impact of the outlier.

7. Scaling and Normalization:
Apply Min-Max scaling or Standardization to numerical features (e.g., speed, distance, traffic density) to ensure model stability.

8. Data Splitting:
Divide the dataset into training, validation, and testing subsets (e.g., 70-15-15 split).
Ensure temporal or geographical consistency if predicting future or regional accident risk.

6. Exploratory Data Analysis (EDA):

1. Univariate Analysis
Accident Severity Distribution:
Bar charts were used to visualize the frequency of accidents by severity (e.g., minor, serious, fatal). This helped identify class imbalance in the dataset.
Time-Based Trends:
Histograms of accidents by hour, day of the week, and month showed peak accident times (e.g., rush hours, weekends).
Location Analysis:
Density maps and histograms of latitude and longitude were used to identify high-risk zones (hotspots).

2. Bivariate and Multivariate Analysis Weather vs. Accident Severity:
Boxplots and heatmaps showed how rainy or foggy conditions correlate with more severe accidents.

Road Type vs. Accident Frequency:
Grouped bar plots highlighted that intersections and curved roads had higher accident rates than straight roads. Speed vs. Severity:
Scatter plots and regression lines revealed that higher speeds are often associated with more severe outcomes.

3. Correlation Analysis
Pearson and Spearman correlation coefficients were calculated to understand relationships between numerical variables like:
Speed
Traffic volume
Response time
Number of vehicles involved
A heatmap was used to visually represent correlations, guiding feature selection for predictive modeling.

4. Geospatial Analysis Accident Heatmaps:
Used GIS tools and heatmaps to visualize accident density by region.
Cluster Analysis:
Applied K-Means clustering to detect accident-prone zones or blackspots based on location data.

5. Outlier and Anomaly Detection
Boxplots and IQR methods were applied to detect unusual values in speed, response time, and vehicle count. These anomalies were further examined to understand edge cases like pile-ups or freak weather conditions.

6. Class Imbalance Check
The class distribution of the target variable (accident severity) was examined, revealing the need for resampling techniques like SMOTE or class weighting during model training

7. Feature Engineering:

1. Time-Based Features
Hour of Day: Extracted from the timestamp to capture rush hours or late-night patterns.
Day of Week: Helps identify weekday vs. weekend accident trends.
Month/Season: Captures seasonal effects like winter-related hazards.
Holiday Indicator: Boolean flag to mark public holidays, which often affect traffic flow and accident rates.

2. Geospatial Features
Accident Zone Clusters: Applied clustering algorithms (e.g., K-Means, DBSCAN) on location coordinates to identify high-risk zones or blackspots.

Urban vs. Rural Label: Derived from geolocation to reflect infrastructure quality and traffic behavior.

3. Weather and Environmental Features

Weather Severity Score: Combined temperature, precipitation, visibility, and wind speed into a single severity metric.
Light Condition Encoding: Encoded categories like daylight, darkness with street lighting, and darkness without lighting into ordinal scores.
Road Condition Flags: Binary flags indicating whether the road was wet, icy, or under construction.

4. Traffic and Movement Features

Traffic Density Estimate: Based on time, location, and known traffic volumes to simulate real-time conditions.
Vehicle Count: Number of vehicles involved in an accident—strongly correlated with severity.
Vehicle Speed Category: Binned speed data into categories (e.g., low, moderate, high) for easier model interpretation.

5. Driver and Behavioral Features (if available)

Driver Age Group: Grouped into brackets (e.g., <25, 25–40, 41–60, >60).
Risk Flags: Created boolean features such as:
Alcohol involved?
Speeding reported?
Seatbelt usage?

6. Historical and Aggregate Features

Accident Frequency at Location: Number of accidents in the same area over a defined time period.
Average Response Time: Average time emergency services take to reach similar locations.
Severity Rate per Area: Proportion of severe accidents at a given location cluster.

7. Encoding and Transformation

One-Hot Encoding: Applied to non-ordinal categorical variables like weather type, road type, and accident type.
Label Encoding: Used for ordinal features such as light conditions or severity level. Scaling: Normalized numerical features (speed, distance, temperature) using Min-Max or StandardScaler for model consistency.

8. **Model Building :**

1. Problem Formulation

Classification Task: Predict the severity of an accident (e.g., Minor, Serious, Fatal).
Input: Preprocessed and engineered features from accident records. Output:
Predicted severity level or accident risk score.

2. Algorithms Used
Several machine learning models were trained and compared:
Logistic Regression: Used as a baseline model for classification.
Decision Tree: Easy to interpret and useful for feature importance analysis.
Random Forest: Ensemble model that improves accuracy and reduces overfitting.
Gradient Boosting (e.g., XGBoost, LightGBM): Delivered the best performance by capturing complex feature interactions.
Support Vector Machine (SVM): Used for high-dimensional data with kernel tuning.
Neural Networks: Deployed for deep learning-based classification with multiple hidden layers.

3. Data Splitting
Training Set: 70% of the dataset used to train the model.
Validation Set: 15% used for hyperparameter tuning and model selection.
Test Set: 15% used for final performance evaluation.

4. Handling Imbalanced Data
SMOTE (Synthetic Minority Over-sampling Technique): Applied to balance the class distribution.
Class Weighting: Adjusted model training to penalize misclassification of minority classes more heavily.

5. Model Evaluation Metrics
Accuracy: Overall correctness of predictions.
Precision, Recall, F1-Score: Especially important for minority classes (e.g., fatal accidents).
Confusion Matrix: To visualize correct and incorrect predictions by class.
ROC-AUC Score: Measured model's ability to distinguish between severity levels.

6. Hyperparameter Tuning
Grid Search & Random Search: Used to optimize key parameters like:
Number of trees (Random Forest, XGBoost)
Learning rate and depth (Boosting models)
Regularization parameters (SVM, Neural Nets)

9. Visualization of Results & Model Insights:

1. Confusion Matrix
Purpose: Shows the true vs. predicted accident severity levels.
Insight: Helped identify whether the model overpredicted minor cases or underpredicted fatal accidents.
Visualization: Heatmap format for easy readability.

2. ROC Curve & AUC Score
Purpose: Evaluates the model's ability to distinguish between different severity classes.
Insight: AUC scores near 1 indicated excellent performance, especially with XGBoost.
Visualization: Multi-class ROC curves for each severity class.

3. Feature Importance Plot
Purpose: Identifies the most influential variables used by the model.
Insight: Key features included:
Time of day
Traffic volume
Road and weather conditions
Accident location cluster
Visualization: Bar chart or horizontal bar plot ranking features by importance score.

4. Partial Dependence Plots (PDP)
Purpose: Show how changes in a specific feature affect the model's predictions.
Insight: For example, accident severity increased sharply during wet or icy conditions.
Visualization: Line plots showing predicted severity as a function of one variable.

5. Accident Hotspot Map
Purpose: Geospatial visualization of accident-prone areas.
Insight: Highlighted high-risk zones for potential intervention (e.g., signal upgrades, speed bumps).
Visualization: Interactive map with heat overlay using latitude and longitude data.
6. Temporal Analysis Charts
Purpose: Show accident frequency over time (hourly, daily, monthly).
Insight: Revealed rush-hour spikes, weekend surges, and seasonal patterns.
Visualization: Line charts and histograms segmented by time attributes.

7. Severity Distribution Before and After SMOTE
Purpose: Compare class distribution before and after balancing the dataset.
Insight: Validated the effectiveness of SMOTE in addressing class imbalance.
Visualization: Side-by-side bar charts.

10. Tools and Technologies Used:

1. Programming Languages:
Python – for AI/ML model development, data analysis, and backend services.
JavaScript (Node.js) – for real-time data processing and web-based dashboards. SQL/NoSQL – for database management (PostgreSQL, MongoDB).

2. AI/ML Frameworks:

TensorFlow / PyTorch – for building and training AI models.
Scikit-learn – for traditional ML algorithms and data preprocessing. OpenCV – for video/image processing from traffic cameras.

3. Data Sources and Integration:
IoT Sensors (Speed, Proximity, LIDAR) – for real-time vehicle and environmental data.
CCTV/Traffic Cameras – integrated with computer vision for accident detection.
GPS & Map APIs (Google Maps, OpenStreetMap) – for location tracking and route analysis.
Weather APIs (OpenWeatherMap) – to include weather conditions in predictions.

4. Cloud and Infrastructure:
AWS / Azure / Google Cloud – for cloud storage, processing, and deployment.
Edge Computing Devices – for processing data closer to the source (e.g., traffic signals).
Docker / Kubernetes – for containerization and scalable deployments.

5. Big Data & Real-Time Processing:
Apache Kafka – for real-time data streaming.
Apache Spark – for large-scale data analytics and ML model training.
Elasticsearch + Kibana – for real-time search, monitoring, and dashboard visualization.

6. Databases:
PostgreSQL / MySQL – for structured data storage.
MongoDB / InfluxDB – for flexible or time-series data like sensor logs.

7. Alert & Notification Systems:
Twilio / Firebase – for sending SMS or push alerts.
MQTT Protocol – for lightweight messaging between IoT devices and servers.

8. Visualization and Frontend Tools:
Power BI / Tableau / Grafana – for visual dashboards.
React.js / Angular – for building the user interface of monitoring tools.

11. Team Members and Contributions:

1.1. [[PERAMALINGAMPERMALINGAM P] P] – –Project Project Leader Leader & & AI AI Developer: Developer:
Led project planning, task delegation, and progress tracking.
Designed and implemented machine learning models for accident prediction. Handled model training, validation, and performance evaluation.

2. [PRAKASH P] – Data Analyst & Integration Specialist:
Collected and preprocessed traffic and accident data from various sources.

Integrated weather data, GPS, and sensor data into the analysis pipeline.
Conducted exploratory data analysis and feature engineering for model input.

3. [PAVAN V] – Backend & System Developer
Developed the backend system for data logging, model integration, and alert management.
Built RESTful APIs to handle real-time data flow and event processing.
Ensured database management and cloud integration for scalable deployment.

4. [PRIYADHARSHINI R] – UI/UX Designer & Testing Lead
Designed the user interface for dashboards displaying traffic status and alerts.
Conducted system testing including real-time simulations and model accuracy checks.
Gathered feedback for usability and refined the interface for better user experience.