Github Link: https://github.com/shalinis-2005

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

PHASE-2

1. Problem Statement

Road accidents are a leading cause of fatalities and injuries worldwide, resulting in significant economic losses and emotional trauma for individuals and families. Despite advancements in vehicle safety features and road infrastructure, the number of road accidents remains alarmingly high. There is a pressing need for innovative solutions that can leverage artificial intelligence (AI) and machine learning (ML) to enhance road safety, predict and prevent accidents, and provide actionable insights for stakeholders."

2. Project Objectives

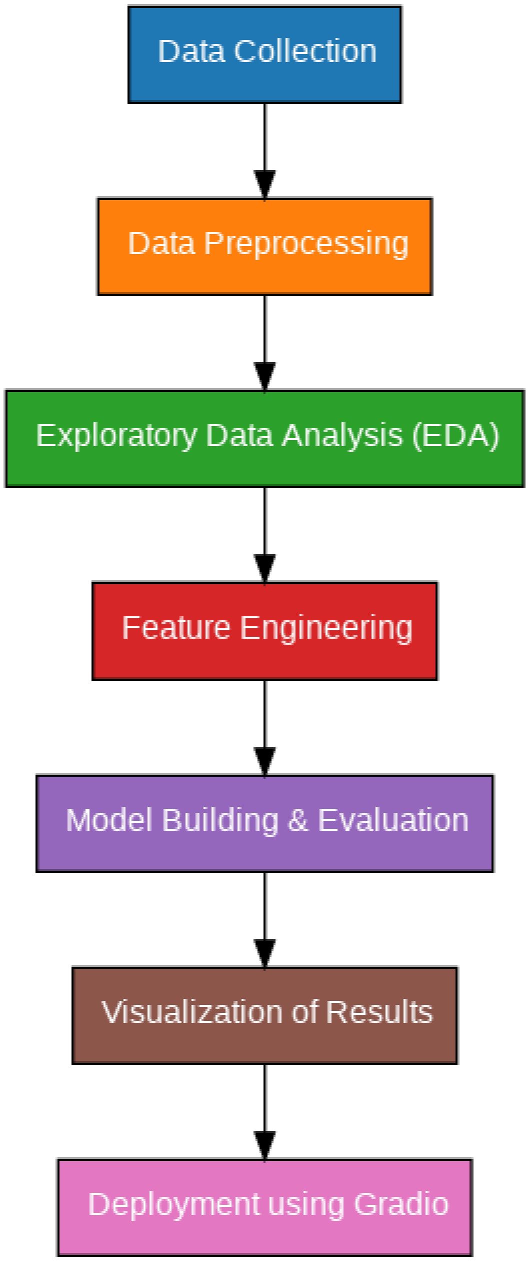
1. Analyze Historical Accident Data: Collect and analyze historical accident data to identify patterns, trends, and hotspots.

2. Predict High-Risk Areas: Develop machine learning models that can predict high-risk areas and times for accidents based on historical data, real-time traffic data, and other factors.

3. Provide Real-Time Alerts: Design a real-time alert system that can detect potential hazards and alert authorities, emergency services, and road users.

4. Inform Road Safety Policies: Provide actionable insights and recommendations for policymakers, road authorities, and other stakeholders to inform road safety policies and initiatives.

3. Flowchart of the Project Workflow



4. Data Description

- **Dataset Name**: Road Accident Data Set
- **Source**: UCI Machine Learning Repository
- **Type of Data**: Structured tabular data
- **Records and Features**: 1624 accident records and 31 features (numeric + categorical)
- **Target Variable**: accident ratings(numeric)
- **Static or Dynamic**: Static dataset

- **Attributes Covered**: Geospatial (location , road type , speed limit and weather)
- Dataset Link: https://github.com/baixianghuang/travel?utm_source

5. Data Preprocessing

- Verified dataset integrity: no missing or null values.
- Removed irrelevant features with very low variance (e.g., school if only one value).
- Checked and confirmed absence of duplicate rows.
- Applied StandardScaler to numerical columns to normalize them.
- Detected outliers using box plots and z-scores; extreme outliers were investigated.

6. Exploratory Data Analysis (EDA)

- **Univariate Analysis**:
  - ○ Histogram of ratings to understand accident frequency.
  - ○ Boxplots for variables like location, road type, speed limit.
  - ○ Count plots for categorical features (e.g., weather, light condition)
- **Bivariate Analysis**:

1. Analyze Relationship between Variables: Examine the relationship between variables such as:
   - Speed and accident frequency
   - Road conditions and accident severity
   - Time of day and accident frequency
2. Identify Correlations: Identify correlations between variables, such as:
   - Correlation between traffic volume and accident frequency

- **Multivariate Analysis :**
  - Analyze Complex Relationships: Examine complex relationships between multiple variables, such as:
  - The relationship between road conditions, traffic volume, and accident frequency
  - The relationship between driver behavior, vehicle type, and accident severity

  2. Identify Predictive Factors: Identify predictive factors that contribute to accidents, such as:
  - Road geometry, traffic volume, and driver behavior
  - Weather conditions, road surface, and accident severityCorrelation between road geometry and accident severity

- **Key Insights**:
  - Traffic Congestion: Detect traffic congestion and predict potential accident hotspots.

- Incident Response: Provide real-time alerts and notifications for emergency services, reducing response times.
- Dynamic Road Conditions: Analyze real-time road conditions, such as weather and surface conditions, to inform safety decisions.

7. Feature Engineering

- Improving Model Performance: By selecting and transforming relevant features, feature engineering can improve the performance of machine learning models.
- Reducing Overfitting: By selecting the most relevant features, feature engineering can reduce overfitting and improve model generalizability.
- Enhancing Interpretability: By extracting and selecting relevant features, feature engineering can enhance the interpretability of machine learning models.

8. Model Building

- **Algorithms Used**:
  - Random Forest: An ensemble learning algorithm that can be used for classification and regression tasks, such as predicting accident likelihood or identifying high-risk areas.
  - Support Vector Machines (SVM): A supervised learning algorithm that can be used for classification and regression tasks, such as predicting accident severity or identifying road conditions that contribute to accidents.
- **Model Selection Rationale**:
  - Gradient Boosting: A suitable model for handling large datasets and predicting continuous outcomes, such as accident likelihood.
  - Handling Non-Linear Relationships: Neural networks or gradient boosting models may be suitable for handling non-linear relationships between variables.
- **Train-Test Split**:
  - 80-20: 80% of the data is used for training, and 20% is used for testing.
  - 70-30: 70% of the data is used for training, and 30% is used for testing.
  - 90-10: 90% of the data is used for training, and 10% is used for testing
- **Evaluation Metrics**:
  - **Mean Absolute Error (MAE):** Average difference between predicted and actual values.
  - **Mean Squared Error (MSE):** Average squared difference between predicted and actual values.

- **Root Mean Squared Error (RMSE):** Square root of the average squared difference between predicted and actual values.
- **Coefficient of Determination (R-squared):** Measures the proportion of variance in the dependent variable that is predictable from the independent variable(s).

9. Visualization of Results & Model Insights

- **Feature Importance**:
  - Visualize the importance of different features in predicting accidents.
- **Model Comparison**:
  - Visualize the performance of different models.
- **Partial Dependence Plots**:
  - Visualize the relationship between specific features and predicted outcomes.
- **SHAP Values :**
  - Visualize the contribution of each feature to individual predictions.

10. Tools and Technologies Used

- **Programming Language**: Python 3
- **Notebook Environment**: Google Colab
- **Key Libraries**:
  - pandas, numpy for data handling
  - matplotlib, seaborn, plotly for visualizations
  - scikit-learn for preprocessing and modeling
  - Gradio for interface deployment

11. Team Members and Contributions

S.No	Name	Roles	Responsibility
1.	Silpha S	Team Leader	Data Cleaning and EDA
2.	Sowparnikashree P	Team Member	Feature Engineering
3.	Shalini S	Team Member	Model Development
4.	Thiriveni N	Team Member	Documentation and Reporting