Github link: https://github.com/BhavaniR2006/project-Traffic-accident-analysis

Phase-2

**EnhancingroadsafetywithAI-driventrafficanalysisand prediction**

## ProblemStatement:

Despite advancements in transportation infrastructure, road traffic accidents continue to be a major causeoffatalities,injuries,andeconomiclossworldwide.Traditionalmethodsoftrafficaccidentanalysis often rely on historical data and reactive measures, which are insufficient for proactive safety planning.

There isa critical need forintelligent systemsthatcannotonly analyzevastamounts ofaccident-related data but also predict potential accident hotspots and high-risk scenarios in real-time. The absence of such predictive capabilities limits the effectiveness of traffic management strategies and endangers public safety.

This project aims to leverage artificial intelligence and machine learning to develop a robust system that can analyze traffic accident patterns and predict future incidents, enablingauthorities to implement preventive measures and enhance road safety proactively.

## ProjectObjectives:

* 1. **DevelopaPredictiveModelforAccidentHotspots**
     + Utilizemachinelearningalgorithmstoanalyzehistoricalaccidentdata,identifying patterns and factors contributing to accidents.
     + Predicthigh-risklocationsandtimes,enablingproactivesafetymeasures.

## IntegrateReal-TimeTrafficandEnvironmentalData

* + - Incorporatelivedatafromtrafficsensors,weatherstations,andGPSdevicestomonitor current road conditions.
    - Enhancepredictionaccuracybyconsideringdynamicfactorssuchastrafficflowand weather changes.

## ImplementReal-TimeAccidentDetectionandAlertSystem

* + - DevelopanAI-basedsystemtodetectaccidentsastheyoccurusingdatafromcameras and sensors.
    - Automaticallyalertemergencyservicesandnearbyvehiclestoreduceresponsetimes and mitigate further incidents.

## AssessandClassifyAccidentSeverity

* + - ApplyAImodelstoevaluatetheseverityofaccidentsbasedon factorslikeimpactforce, vehicle speed, and location.
    - Prioritizeemergencyresponsesandresourceallocationeffectively.

## OptimizeTrafficManagementandInfrastructurePlanning

* + - Analyzetrafficpatternsandaccidentdatatorecommendimprovementsinroaddesign and traffic signal timings.
    - Supporturbanplannersincreatingsaferandmoreefficientroadnetworks.

## EnhancePublicAwarenessandPolicyMaking

* + - Providedata-driveninsightstopolicymakersforinformeddecision-making.

3.Flowchart:

DataPreprocessing

DataCollection&Integration

Modeldevelopment

ExploratoryDataAnalysis (EDA)

Real-TimeMonitoring

Prediction&Risk Assessment

### ****4.Data Description:****

* **Dataset Name**: US Accidents (2016–2021) Dataset
* **Source**: Kaggle – US Accidents (Kaggle)
* **Type of Data**: Structured, tabular data
* **Records and Features**: Over 3 million accident records with 49 features
* **Target Variable**: Severity (1 to 4 scale indicating accident severity)
* **Static or Dynamic**: Static dataset (collected from APIs like MapQuest, Bing, and transportation departments)
* Dataset link: https://www.kaggle.com/datasets/ankushpanday1/global-road-accidents-dataset

## 5. DataPreprocessing:

#### ****Data Cleaning****

* **Missing Values Handling**:
  + Dropped columns with excessive missing data (e.g., Number, Wind\_Chill(F)).
  + Imputed missing values in key features using:
    - Mean/median for numeric features (Temperature, Humidity)
    - Mode for categorical features (Weather\_Condition, Sunrise\_Sunset)
* **Duplicate Removal**:
  + Identified and removed duplicate records using Start\_Time, Location, and Severity.

#### ****Data Type Conversion****

* Converted datetime fields like Start\_Time and End\_Time to datetime objects.
* Categorical variables such as Weather\_Condition, Side, Sunrise\_Sunset were converted to string type or category type for encoding.

#### ****Feature Extraction****

* Extracted **time-based features**:
  + Hour, Day\_of\_Week, Month, Year, Is\_Weekend, Is\_Rush\_Hour
* Extracted **location-based features**:
  + Binned latitude/longitude into zones or used geospatial clustering (e.g., DBSCAN)
* Created **binary flags** from road feature indicators:
  + Has\_Junction, Has\_Crossing, Has\_Traffic\_Signal, etc.

#### ****Handling Categorical Variables****

* Applied **One-Hot Encoding** to low-cardinality categorical features like Side, Sunrise\_Sunset.
* Used **Label Encoding or Frequency Encoding** for higher-cardinality features like City and Weather\_Condition.

#### ****Outlier Detection and Treatment****

* Used **IQR** or **Z-score methods** to detect outliers in features like Distance, Temperature, and Visibility.
* Capped or removed extreme values where necessary to avoid skewing model performance.

#### ****Feature Scaling****

* Applied **StandardScaler** or **MinMaxScaler** to numerical features for algorithms sensitive to feature magnitude (e.g., SVM, KNN, Neural Networks).

#### ****Train-Test Split****

* Split the dataset into training and testing sets (e.g., 80/20 split).
* Ensured stratification on the Severity class to maintain class balance across sets.

#### ****Balancing the Dataset (if needed)****

* Checked for class imbalance in the Severity target.
* Applied **SMOTE (Synthetic Minority Oversampling Technique)** or **class weighting** if the imbalance was significant.

## 6.Exploratory Data Analysis (EDA):

### UnivariateAnalysis

* **TargetVariableDistribution**:Ifpredictingseverity,inspectitsdistribution.
  + Plot:Barplotorpiechartforaccidentseverityclasses.

## NumericalFeatures:

* + Histogramsforvehiclespeed,numberofvehicles,distancefromjunctions,etc.

## CategoricalFeatures:

* + Countplotsfor:
    - Accidentcause
    - Weatherconditions
    - Roadtype
    - Lightingconditions
    - Timeofday(morning,evening, night)

***Bivariate/MultivariateAnalysis***

## CorrelationMatrix:

* + Heatmapofcorrelationsbetweennumericalvariables.
  + Lookforrelationshipsbetweenweather,speed,andaccidentseverity.

## Severityvs.Time:

* + Lineplotsshowingaccidentfrequencyorseveritybyhour,day,ormonth.

## Severityvs.Weather/RoadConditions:

* + Stackedbarchartsor boxplots.

## Heatmaps:

* + Spatialheatmapsofaccidentlocationsusinglatitudeandlongitude.
  + Time-heatmaps(e.g.,hoursvs.days)tospottemporalaccidentpatterns.

***GeospatialAnalysis***

## MapVisualizations:

* + UsetoolslikeFoliumorPlotlytoplotaccident hotspots.
  + Clustermapstovisualizeaccidentdensity.

## Region-wiseAnalysis:

* + Groupandaggregatebydistrict/citytoanalyzeaccidenttrendsgeographically.

***Outlier Detection***

## Box Plots:

* + Identifyoutliersinspeed,vehiclecount, etc.
* **Z-ScoreorIQRmethods**toflagextremevalues.

### TimeSeriesTrends

* Analyzeaccidentcountsovertime(daily,weekly,monthly).
* Detectseasonalityorspikesduetoevents,holidays,orweatherconditions.

### InsightsSummary

* Summarizekeyfindings,forexample:
  + Mostaccidentsoccurduringrainynightsinintersections.
  + High-speedzonesshowhigherfatalaccidentrates.

.

## 7.FeatureEngineering

#### ****Temporal Features****

* **Hour of Day** – To capture rush hours and night-time accidents.
* **Day of Week** – To differentiate between weekday and weekend traffic patterns.
* **Month & Season** – For identifying seasonal accident trends.

#### ****Geospatial Features****

* **Geohash / Grid Binning** – To spatially cluster nearby accident points.
* **Proximity to Urban Areas** – By categorizing locations as urban, suburban, or rural.
* **High-Risk Zone Identification** – By aggregating accident counts in regions (hotspot detection).

#### ****Weather and Environmental Features****

Using weather-related columns:

* **Weather Risk Level** – Group weather conditions into risk categories (e.g., clear, moderate, hazardous).
* **Poor Visibility Indicator** – Flag when Visibility is below a safety threshold.
* **Wet Surface Flag** – Based on precipitation, humidity, and weather condition.

#### ****Road Condition Features****

From boolean fields like Junction, Crossing, Traffic\_Signal:

* **Intersection Count** – Combine multiple fields to indicate traffic complexity.
* **Road Complexity Score** – Composite score using presence of traffic signals, bumps, crossings, etc.

#### ****Distance & Duration Features****

* **Accident Duration** – Difference between Start\_Time and End\_Time.
* **Accident Area** – Combine Distance(mi) and average speed (if available) to infer impact zone.

#### ****Behavioral or External Flags****

* **Rush Hour Flag** – Based on typical work commute hours (e.g., 7–9 AM, 4–7 PM).
* **Night-Time Indicator** – Using Sunrise\_Sunset or Hour values.
* **Road Type Risk** – If available, encode types of roads (e.g., highway, residential) by observed risk levels.

#### ****Target Encoding (Optional)****

* For high-cardinality categorical variables (like City or Street), compute:
  + **Accident Rate per Category** – Mean target (Severity) grouped by the category.

### ****8.Model Building****

#### ****Problem Framing****

* **Type of Problem**: Multi-class classification (Severity levels from 1 to 4).
* **Target Variable**: Severity
* **Features Used**: Time of day, day of week, weather conditions, road characteristics, traffic control indicators, etc.

#### ****Model Selection****

Several machine learning algorithms were explored to evaluate their effectiveness:

* **Logistic Regression**: Used as a baseline model for comparison.
* **Decision Tree**: Useful for interpreting decision paths and feature importance.
* **Random Forest**: An ensemble model that improves accuracy and reduces overfitting.
* **Gradient Boosting (e.g., XGBoost or LightGBM)**: High-performing model with excellent handling of imbalanced and complex datasets.
* **K-Nearest Neighbors (KNN)**: Evaluated for spatial context but limited by high dimensionality.
* **Support Vector Machine (SVM)**: Tested but scaled poorly with large datasets.
* **Neural Networks**: Considered for deeper modeling but required more data tuning and resources.

#### ****Training and Validation****

* The dataset was split into **training** and **testing** sets (typically 80/20 split).
* **Cross-validation** (e.g., 5-fold) was used to ensure robustness and prevent overfitting.
* **Class imbalance** was addressed using techniques like:
  + Class weighting in model training
  + Oversampling methods like **SMOTE**

#### ****Evaluation Metrics****

To assess model performance, the following metrics were used:

* **Accuracy** – Overall correctness of predictions.
* **Precision, Recall, F1-Score** – Evaluated for each class to account for imbalance.
* **Confusion Matrix** – To visualize classification results and misclassifications.
* **ROC-AUC** – For multi-class discrimination capability (when applicable).

#### ****Model Optimization****

* **Hyperparameter tuning** was conducted using techniques like:
  + Grid Search
  + Randomized Search
  + Bayesian Optimization (in advanced setups)
* Key parameters tuned included:
  + Tree depth, learning rate, number of estimators (for boosting models)
  + Max features, minimum samples split, etc.

#### ****Model Selection Outcome****

After multiple experiments, models like **Random Forest** and **XGBoost** generally provided:

* Higher accuracy
* Better handling of missing values
* Strong feature importance insights

These models were selected for **final deployment** and **interpretation** based on performance and explainability.

### ****9.Visualization of Results & Model Insights****

#### ****Exploratory Data Visualizations****

* **Accidents by Time of Day**:
  + A bar chart showed peak accident occurrences during morning (7–9 AM) and evening (4–7 PM) rush hours.
* **Accidents by Day of the Week**:
  + A heatmap revealed higher incidents on Fridays and weekends, likely due to increased travel and recreational driving.
* **Accidents by Weather Conditions**:
  + A pie or bar chart illustrated that most severe accidents occurred during adverse weather (e.g., rain, fog, snow).
* **Geospatial Maps (Heatmaps)**:
  + Location-based clustering (via latitude & longitude) highlighted **accident hotspots** in urban centers and highway junctions.
* **Road Feature Distribution**:
  + Visuals indicated that roads with traffic signals, crossings, and intersections had a higher frequency of incidents.

#### ****Model Performance Visualizations****

* **Confusion Matrix**:
  + Clearly showed how well each severity class (1–4) was predicted. Class 2 and 3 often had the most confusion due to overlapping features.
* **Precision, Recall, and F1-Score Bar Charts**:
  + Helped compare model performance per class and determine whether the model favored any particular severity level.
* **ROC Curves (if applicable)**:
  + Illustrated the trade-off between true positive rate and false positive rate for multi-class classification.

#### ****Feature Importance Visualization****

* **Feature Importance Bar Plot (Random Forest / XGBoost)**:
  + Identified top predictors of accident severity, such as:
    - **Weather\_Condition**
    - **Visibility**
    - **Hour of Day**
    - **Junction Presence**
    - **Distance**
* **SHAP Summary Plot (Explainable AI Tool)**:
  + Provided deeper interpretability by showing:
    - How each feature influenced predictions
    - Direction of impact (increasing or decreasing accident severity)

#### ****Insights Derived from Visuals****

* **Accidents are more frequent and severe during low visibility and poor weather.**
* **Urban junctions and road intersections are high-risk zones.**
* **Night-time and weekend driving is associated with a higher likelihood of severe outcomes.**
* **Model interprets 'Hour of Day' and 'Weather' as key factors — validating known human and environmental risks.**

### ****10.Tools and Technologies Used****

#### ****Programming Languages****

* **Python**: Core language used for data preprocessing, analysis, visualization, and machine learning model development.
* (Optional) **SQL**: Used for structured data querying if stored in a relational database.

#### ****Libraries and Frameworks****

* **Pandas**: For data manipulation and preprocessing (cleaning, filtering, merging datasets).
* **NumPy**: For numerical operations and array handling.
* **Matplotlib & Seaborn**: For creating data visualizations such as heatmaps, bar charts, and distribution plots.
* **Scikit-learn**: For model building (e.g., Logistic Regression, Decision Trees, Random Forest), evaluation, and feature scaling.
* **XGBoost / LightGBM**: For gradient boosting and improved model performance in classification tasks.
* **SHAP**: For model interpretability and understanding feature impact on predictions.

#### ****Geospatial & Time Series Tools****

* **Folium / Plotly**: For interactive geospatial mapping and accident hotspot visualization.
* **Datetime Module**: For time feature extraction and temporal analysis.

#### ****Data Sources & Storage****

* **Kaggle**: Primary source for the US Accidents dataset.
* **CSV / Excel**: Format used for storing and loading structured datasets.

#### ****Model Development Environment****

* **Jupyter Notebook**: Used for step-by-step development, testing, and presentation of data science workflows.
* **Google Colab / Anaconda**: Alternative platforms used for cloud-based or local development.

#### ****Documentation & Collaboration****

* **Google Docs / Microsoft Word**: For writing reports and documentation.
* **GitHub** (optional): For version control and collaborative code management.

**11.TeamMembersand Contributions:**

# R.Bhavani

* **DataCleaning**:Handledmissingvalues,correctedinconsistententries,andstandardizedformats.
* **ModelDevelopment**:Implemented machinelearningmodelsand tunedthem foroptimal performance.

# C. Bhavani

* **ExploratoryDataAnalysis(EDA)**:Createdvisualizations,analyzedtrends,andidentifiedaccidentpattern
* **FeatureEngineering**:Generatedmeaningfulfeaturesfromrawdatasuchastime,location,andenvironmen

# M.Dhesiya

* **DocumentationandReporting**:Compiledprojectreports,visualizations,and results.
* Assistedwithdashboarddesignand summaryof findingsfor presentation