

Intel Unnati Industrial Training – Summer 2023

Project Model

Problem Statement: Accident Locations on Indian Roads

Team Name: Ashutosh

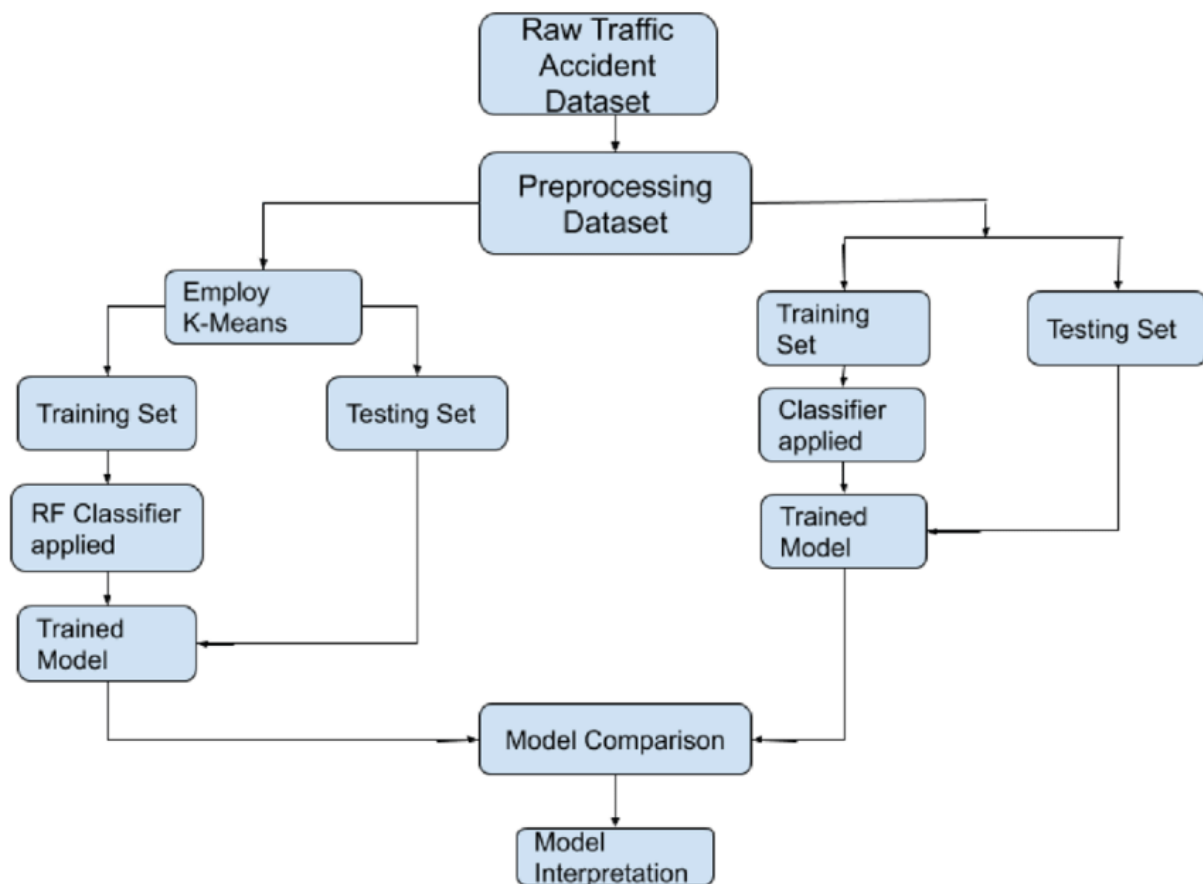
Members: 1

Member Name: Ashutosh Pratap Singh

(ashutosh.21SCSE1010538@galgotiasuniversity.edu.in)

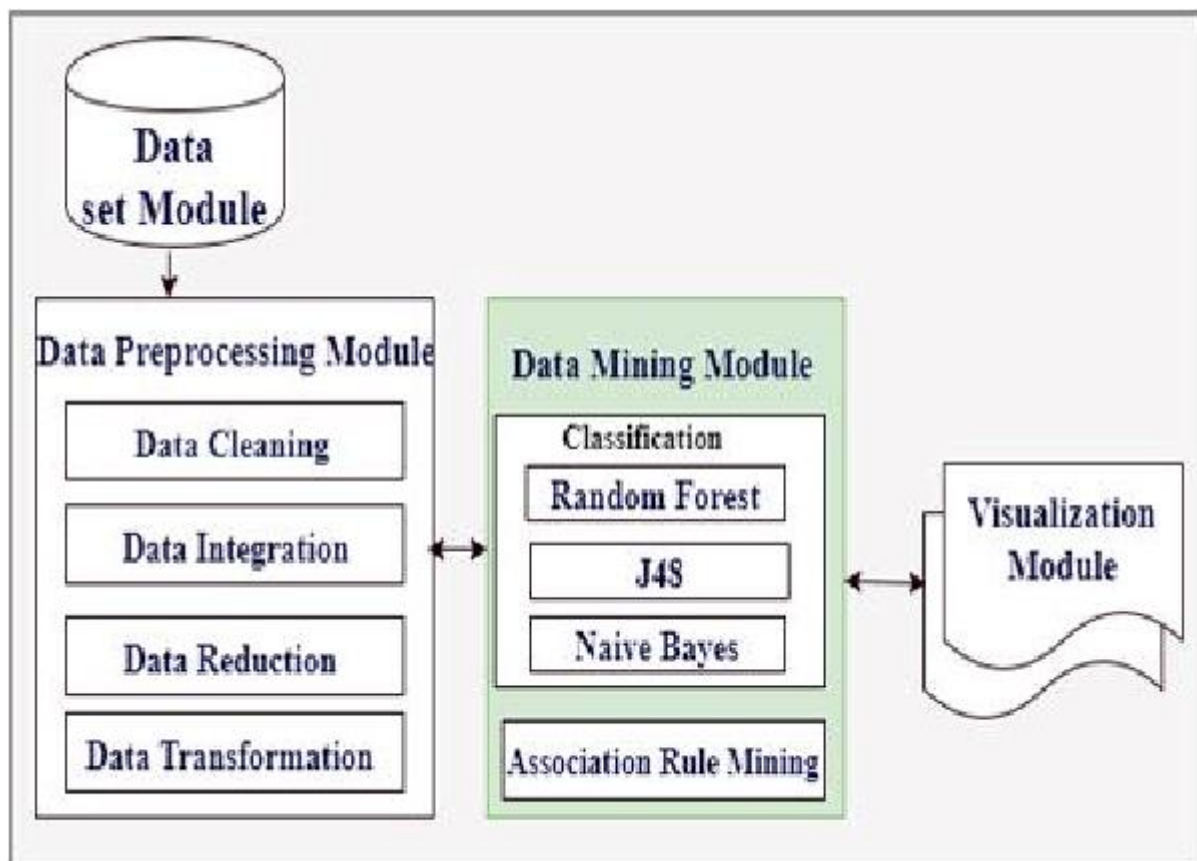
1.Data Collection:

Gather data related to accident locations on Indian roads. This data can be obtained from various sources, such as government reports, police records, insurance agencies, or any other relevant data repositories. Additionally, consider incorporating other data points like road infrastructure, traffic density, weather conditions, and time of the day to enhance the analysis.



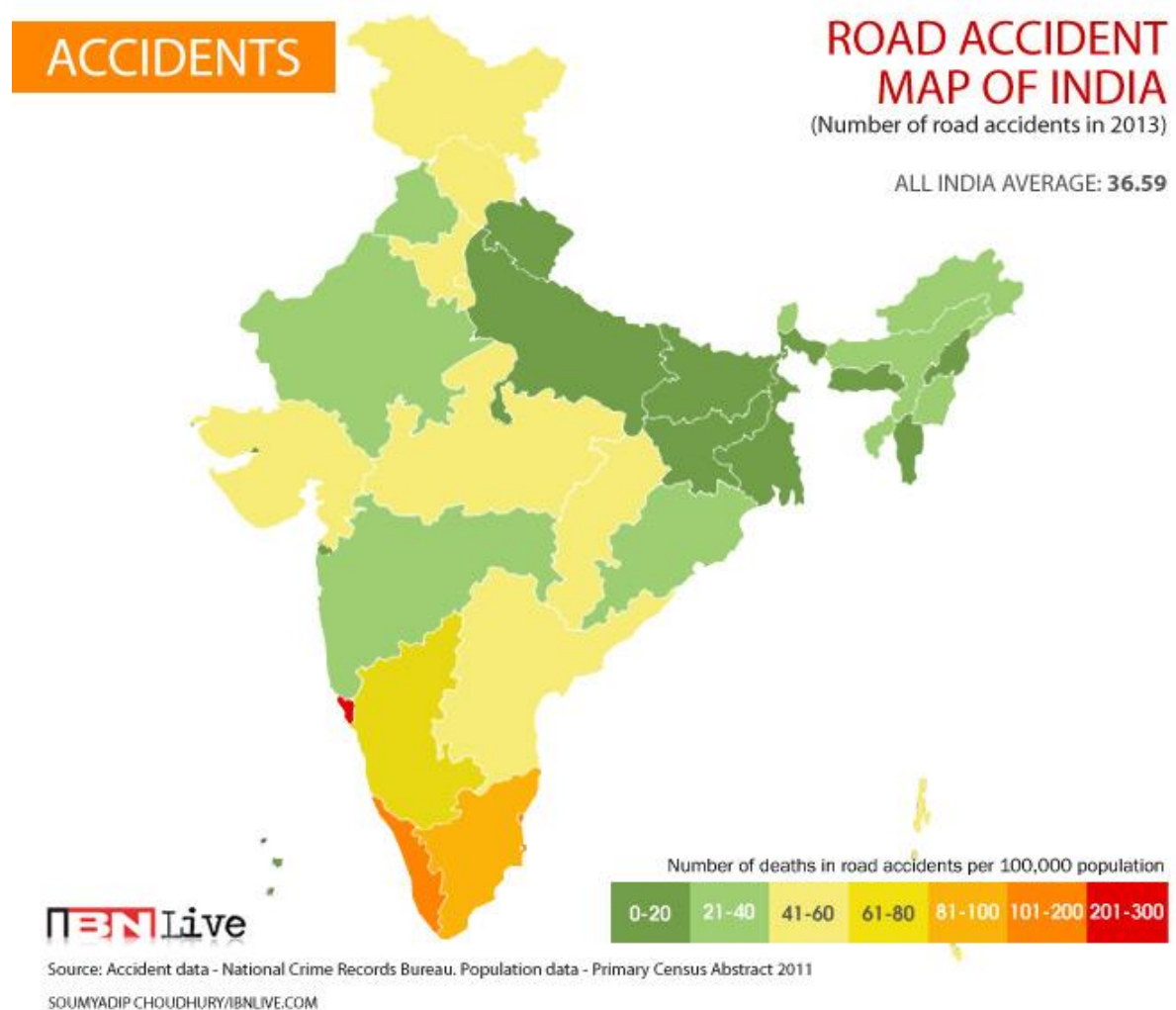
2. Data Preprocessing:

Clean and preprocess the collected data to ensure its quality and consistency. This step may involve removing duplicates, handling missing values, standardizing data formats, and transforming variables into appropriate representations.



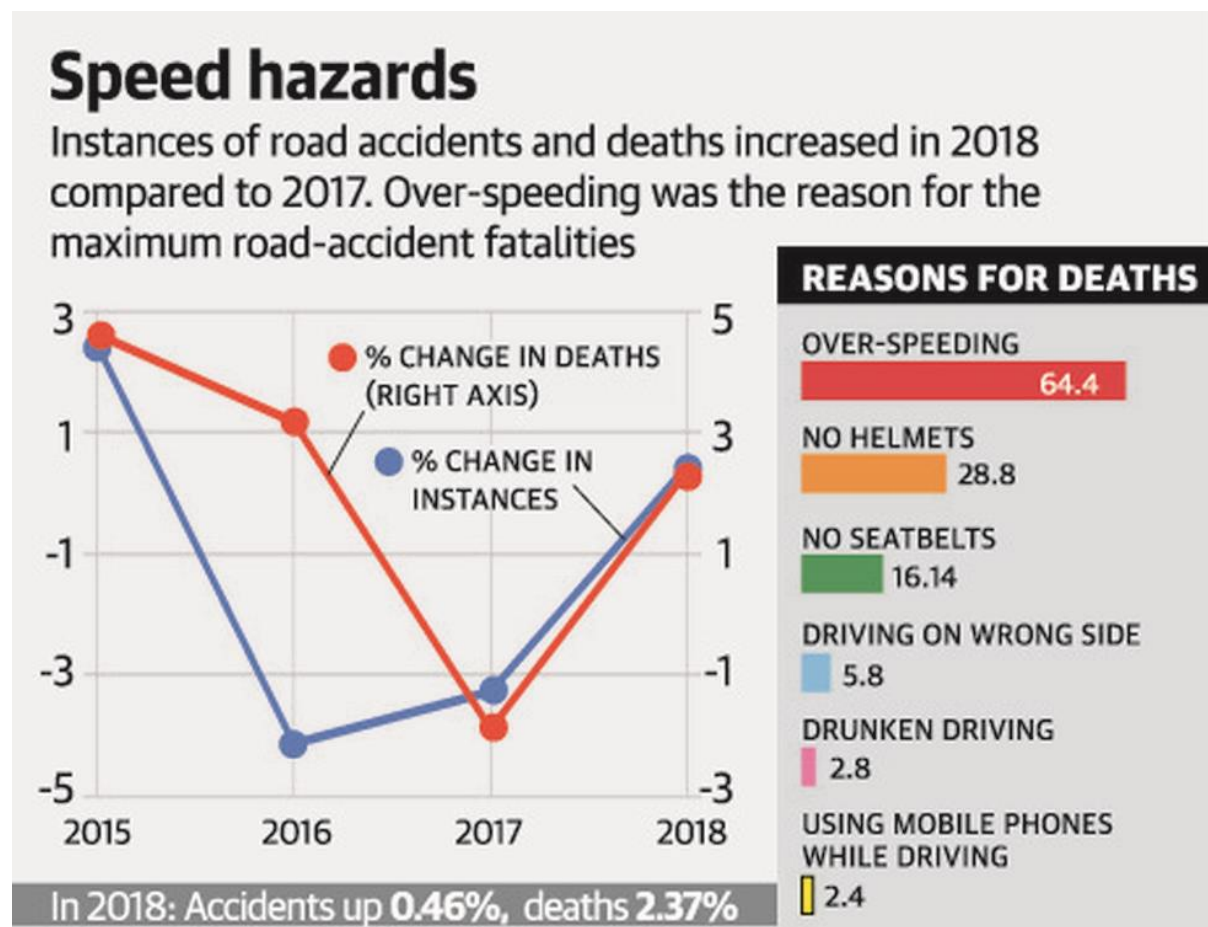
3.GeoCoding:

Geocode the accident location data by converting addresses or descriptions of accident locations into latitude and longitude coordinates. This step is crucial for spatial analysis and mapping.



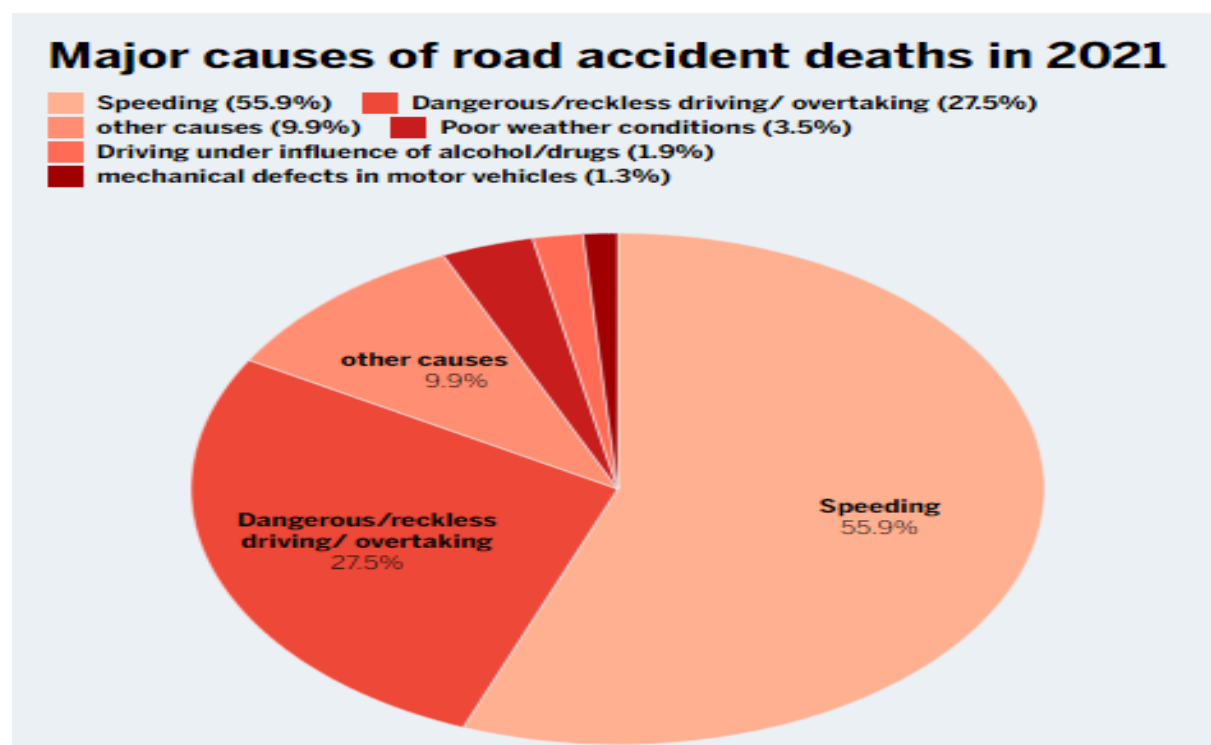
4. Exploratory Data Analysis:

Perform exploratory data analysis to understand the characteristics of the accident data. Visualize the data through plots, charts, and maps to identify patterns, trends, and potential factors contributing to accidents on Indian roads.



5. Feature Engineering:

Create additional features that might be useful for predicting accident locations. For example, you could extract features like road type, speed limit, distance from intersections, presence of traffic signals, or proximity to hospitals or schools. These engineered features can provide valuable insights to the model.



6. Model Selection:

Choose an appropriate machine learning or statistical model to predict accident locations. Some commonly used models for spatial analysis include decision trees, random forests, support vector machines (SVM), or neural networks. Consider the nature of your data, the complexity of the problem, and the available computing resources when selecting the model.



7. Model Training:

Split your data into training and testing sets. Use the training data to train the selected model using appropriate algorithms and techniques. Ensure you account for any class imbalance in the accident data by using techniques like oversampling or undersampling.

8. Model Evaluation:

Evaluate the trained model's performance using appropriate evaluation metrics such as accuracy, precision, recall, or F1-score. Additionally, consider using spatial evaluation measures such as spatial accuracy or distance-based metrics to assess the model's predictive capabilities for accident locations.

9. Model Deployment:

Deploy the trained model into a production environment, such as a web application or an API, to make predictions on new or unseen accident data. Ensure the model is scalable, efficient, and accessible to end-users.

10. Continuous Improvement:

Monitor the model's performance in the production environment and collect feedback from users. Incorporate user feedback and new data to continuously improve the model's accuracy and relevance.