

Road Accident Prediction Using Machine Learning

A Project Report

Submitted to the FACULTY of ENGINEERING of

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, KAKINADA

In partial fulfillment of the requirements,

for the award of the Degree of

Bachelor of Technology

In

Electronics and Communication Engineering

By

V. Reshma Priya

(20481A04O2)

V. Venkata Vamsi

(20481A04O1)

Sk. BB Ayesha

(20481A04K9)

Under the Guidance of

Ms. Ramya.P

Assistant Professor



Department of Electronics and Communication Engineering
SESHADRI RAO GUDLAVALLERU ENGINEERING COLLEGE
(An Autonomous Institute with Permanent Affiliation to JNTUK, Kakinada)
SESHADRI RAO KNOWLEDGE VILLAGE
GUDLAVALLERU - 521356
ANDHRA PRADESH
2023-24

Road Accident Prediction Using Machine Learning

A Project Report

Submitted to the FACULTY of ENGINEERING of

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, KAKINADA

In partial fulfillment of the requirements,

for the award of the Degree of

Bachelor of Technology

in

Electronics and Communication Engineering

By

V. Reshma Priya

(20481A04O2)

Sk. BB Ayesha

(20481A04K9)

V. Venkata Vamsi

(20481A04O1)

Under the Guidance of

Ms. Ramya.P

Assistant Professor



Department of Electronics and Communication Engineering
SESHADRI RAO GUDLAVALLERU ENGINEERING COLLEGE
(An Autonomous Institute with Permanent Affiliation to JNTUK, Kakinada)
SESHADRI RAO KNOWLEDGE VILLAGE
GUDLAVALLERU – 521356
ANDHRA PRADESH
2023-24

Department of Electronics and Communication Engineering
SESHADRI RAO GUDLAVALLERU ENGINEERING COLLEGE
(An Autonomous Institute with Permanent Affiliation to JNTUK, Kakinada)
SESHADRI RAO KNOWLEDGE VILLAGE
GUDLAVALLERU – 521356



CERTIFICATE

This is to certify that the project report entitled **“Road Accident Prediction using Machine Learning”** is a bonafide record of work carried out by **V. Reshma Priya (20481A04O2), Sk. BB Ayesha (20481A04K9), V. Venkata Vamsi (20481A04O1)** under my guidance and supervision in partial fulfillment of the requirements, for the award of the degree of Bachelor of Technology in **Electronics and Communication Engineering** of **Seshadri Rao Gudlavalleru Engineering** affiliated to **Jawaharlal Nehru Technological University, Kakinada**.

(Ms.Ramya.P)

Project Guide

(Dr.B.Rajasekhar)

Head of the Department

Acknowledgement

We are very glad to express our deep sense of gratitude to **Ms.Ramya.P**, Assistant Professor, Electronics and Communication Engineering for guidance and cooperation for completing this project. We convey our heartfelt thanks to her for her inspiring assistance till the end of our project.

We convey our sincere and indebted thanks to our beloved Head of the Department **Dr.B.Rajasekhar**, for his encouragement and help for completing our project successfully.

We also extend our gratitude to our Principal **Dr. B. Karuna Kumar**, for the support and for providing facilities required for the completion of our project.

We impart our heartfelt gratitude to all the Lab Technicians for helping us in all aspects related to our project.

We thank our friends and all others who rendered their help directly and indirectly to complete our project.

Team Members

V. Reshma Priya (20481A04O2)

Sk. BB Ayesha (20481A04K9)

V. Venkata Vamsi (20481A04O1)

CONTENTS

TITLE	PAGE NO
LIST OF FIGURES	i
LIST OF TABLES	ii
ABSTRACT	iii
1. Introduction	1
1.1 Background	1
1.2 Aim of this Project	2
1.3 Methodology	2
1.4 Significance of this Work	3
1.5 Outline of this Report	3
2. Literature Survey	4
3. Proposed Method	6
3.1 Methodology	6
3.2 Implementation	7
3.2.1 Dataset	7
3.2.2 Data Preprocessing	9
3.2.3 Training and Testing Data	10
3.2.4 Predictive Factors	11
3.2.5 Technologies Used	12
3.2.6 Algorithms	17
4. Results	18
4.1 Web Development	18
4.1.1 Running Flask Web Application	19
4.1.2 Web Interface Results	21
5. Conclusion and Future Scope	22
References	23
Project Outcomes Mapped with Programme Specific Outcomes and Programme Outcomes	26

LIST OF FIGURES

Fig. No.	NAME OF THE FIGURE	Page No.
3.1	Block Diagram of Proposed Method	6
3.2	Dataset	7
3.3	Steps of Data Preprocessing	8
4.1	Running the Application	19
4.2	Web Page	19
4.3	Output for Low Probability of Accident	20
4.4	Output for High Probability of Accident	21

LIST OF TABLES

Fig. No.	NAME OF THE TABLE	Page No.
4.1	Evaluation Metrics of Classification Algorithms	18

ABSTRACT

Due to the escalating number of vehicles on roads, the incidence of daily traffic accidents is rising dramatically. This surge in accidents and fatalities underscores the urgency of accurately forecasting accident rates over time to facilitate informed decisions by transportation authorities. Analyzing accident occurrences can provide valuable insights for developing effective strategies to mitigate accidents. Although accidents often exhibit inherent uncertainty, patterns of regularity emerge over time in specific areas. Leveraging these patterns can enable us to develop robust accident prediction models. Our study focuses on examining the correlations between road accidents, road conditions, and environmental factors to enhance accident prediction accuracy.

Utilizing machine learning techniques, including the K-Nearest Neighbors (K-NN) Algorithm, Support Vector Machines (SVM), Random Forest Algorithm, Naive Bayes Algorithm, and Decision Tree Algorithm, this project aims to construct a predictive model for accident occurrences. By harnessing the power of these algorithms, it seeks to improve understanding of accident dynamics and contribute to the development of proactive measures to reduce accidents on roads.

Keywords: Road Accident, Machine Learning (ML), Dataset.

CHAPTER 1

INTRODUCTION

In today's fast-paced world, road accidents continue to be a pressing concern, affecting millions of lives and economies globally. Despite advancements in vehicle safety technology and road infrastructure, the frequency and severity of road accidents remain a significant challenge. Factors such as increasing urbanization, population growth, distracted driving, and varying weather conditions contribute to the complexity of the road safety landscape.

Due to the exponentially increasing number of vehicles on the road, the number of accidents occurring on a daily basis is also increasing at an alarming rate. With the high number of traffic accidents and deaths these days, the ability to forecast the number of traffic accidents over a given time is important for the transportation department to make scientific decisions. In this scenario, it will be good to analyze the occurrence of accidents so that this can be further used to help us in coming up with techniques to reduce them. Even though uncertainty is a characteristic trait of majority of the accidents, over a period of time, there is a level of regularity that is perceived on observing the accidents occurring in a particular area. We implemented a machine learning model which predicts the accident occurrence.

1.1 Background

Road accidents represent a significant global challenge, with profound implications for public health, economic well-being, and societal stability. According to the World Health Organization (WHO), road traffic injuries are a leading cause of death worldwide, particularly among young adults aged between 15 and 29 years. Moreover, road accidents impose substantial economic burdens on societies, including medical expenses, property damage, and loss of productivity.

Understanding the factors contributing to road accidents is essential for formulating effective prevention strategies and mitigating their impact. Accidents on roads can result from a myriad of factors, including human error, environmental conditions, vehicular characteristics, and infrastructure deficiencies. Traditional approaches to accident prevention have largely relied on

reactive measures, such as law enforcement, road safety campaigns, and infrastructure improvements based on historical accident data analysis.

However, these methods have inherent limitations, as they often fail to anticipate future accidents or address underlying risk factors comprehensively. Furthermore, the complexity and interdependence of variables influencing road safety necessitate more sophisticated analytical tools for accurate prediction and proactive intervention.

1.2 Aim of this Project

The aim of our project "Road Accident Prediction Using Machine Learning" is to develop a predictive model that can accurately forecast the likelihood of road accidents based on various factors such as weather conditions, road infrastructure, traffic density, and historical accident data. By leveraging machine learning algorithms, the project aims to provide insights that can help improve road safety measures, optimize resource allocation for emergency services, and ultimately reduce the number of road accidents and associated casualties. Additionally, the project seeks to create a user-friendly web application interface that allows users to input relevant parameters to receive real-time predictions or risk assessments.

The objectives include developing accurate model to predict the road accidents. The project uses machine learning algorithms such as SVM, K-NN, Random Forest, Decision Tree, Naive Bayes for prediction of road accidents and conclude the fastest algorithm.

1.3 Methodology

The methodology involves several key steps. First, relevant accident data is collected and preprocessed, addressing issues like missing values. Meaningful features are extracted from the data. The dataset is then split into training and testing subsets. Next, machine learning models (including Random Forest, Decision Tree, Naive Bayes, KNN, and SVM) are selected and trained using the training data. By model evaluation metrics, such as accuracy and precision, guide the choice of the best-performing model. Applying the chosen model to the testing data yields accident probability predictions. Finally, result analysis provides insights for decision-makers and safety improvements. Overall, this process aims to enhance road safety and prevent accidents by prediction of accidents.

1.4 Significance of work

The significance of this project is profound, resonating across multiple domains with tangible impacts on public safety and well-being. By harnessing the power of machine learning to predict road accidents based on a range of critical factors such as weather conditions, light settings, vehicle types, and road characteristics, this work stands to save lives and prevent injuries. The ability to predict accidents provides early interventions and optimal resource allocation, allowing law enforcement and emergency services to respond effectively. Furthermore, the insights gained by predictive modelling might help authorities enhance advertising lighting and road design in high-risk regions. From a policy perspective, data-driven evidence provided by this project can drive evidence-based policymaking in road safety, leading to the implementation of targeted measures that address specific risk factors identified through machine learning analysis. Beyond the immediate life-saving impact, preventing accidents translates into substantial cost savings, reducing medical expenses, vehicle repair costs, and insurance claims. Furthermore, this project plays a pivotal role in promoting awareness about road safety among the general public, fostering a culture of responsible driving behavior and pedestrian awareness. As the project evolves and refines its algorithms, it sets a precedent for continuous improvement in accident prediction and prevention efforts, ensuring ongoing advancements in road safety practices and outcomes.

1.5 Outline of this Report

In chapter 2, The literature survey is given. The proposed method and implementation of proposed method is discussed in detail in chapter 3. In chapter 4, Results of proposed system is discussed, followed by conclusion and future scope in chapter 5.

In this chapter, we introduced the project, outlining its aim, methodology, and significance. We discussed the project's objectives and approach, highlighting its importance in the field.

CHAPTER 2

LITERATURE SURVEY

The field of road safety and accident prediction has witnessed significant advancements with the application of machine learning algorithms. Various studies have explored different methodologies and models to analyze factors influencing road accidents and predict accident severity.

Chen et al. [1], highways emerge as the predominant location for the occurrence of a significant portion of reported accidents.

Williams et al. [2], it was revealed that the age and driving experience of individuals significantly influence the frequency of accidents.

Sarkar et al. [3] conducted a comparative analysis investigating the prevalence of accidents across various road types. In addition to examining the factors contributing to accidents, their study revealed a higher incidence of accidents on highways compared to standard roads similar to [1].

Zheng et al. [4] conducted research focusing on the spectrum of injuries resulting from motor vehicle accidents. Additionally, they investigated the emotional state of the drivers involved, exploring its potential influence on the occurrence of accidents.

In their study, Tessa K. Anderson et al. [5] proposed a method aimed at identifying high-density casualty zones. This strategy involves implementing a clustering procedure to identify clusters where stochastic events are likely to occur. Consequently, it allows for the assessment of their occurrence within these clusters.

The study conducted by [6] took into account several factors including the log-straight model, driver characteristics, pedestrian traits, road traffic, and vehicle classifications. This comprehensive approach provides clear insights into the factors influencing casualties in school zones.

De Ona et al. [7] employed Latent Class Clustering and Bayesian methodologies in their examination of automobile collisions to identify the key determinants of casualty severity. The simultaneous application of these two techniques is particularly noteworthy as it unveils supplementary insights into the data.

Mahendra G et al. [8] devised a tool for detecting reckless driving on roadways and

promptly notifying traffic authorities in the event of any speed infractions.

Jamal Raiyn et al. [9] present a model that identifies traffic incidents by analyzing the speed fluctuations of vehicles located both upstream and downstream of a specific point on the highway.

In a study conducted by X. Gao et al. [10], a novel algorithm termed Weighted Quantitative Random Forest (WQRF) was introduced. This algorithm aims to forecast employee salary turnover within various industries. The predictive model incorporates several key features, including overtime hours, age, monthly income, distance from home, and tenure within the company. These factors collectively contribute to the model's ability to anticipate shifts in employee salary turnover rates.

Our project builds upon these research findings by employing a comparative study of machine learning algorithms including Random Forest, Decision Tree, SVM, KNN, and Naive Bayes. Through rigorous data preprocessing, feature selection, and model training, this project aims to develop a robust predictive model for road accident severity prediction. By synthesizing insights from existing literature and leveraging state-of-the-art machine learning techniques, this project seeks to contribute to the advancement of road safety measures and accident prevention strategies. This project aims to provide valuable insights for stakeholders in traffic management, law enforcement agencies, and policymakers to make informed decisions and improve road safety for all.

We summarized existing research literature relevant to the topic of study in this chapter.

CHAPTER 3

PROPOSED METHOD

3.1 METHODOLOGY

In this project, the methodology adopted for predicting road accidents integrates a thorough approach to data collection, preparation, and model development to ensure robust predictive performance. And initiated the process by sourcing data from diverse public databases and collaborations with local authorities, focusing on key variables such as weather conditions, light conditions, vehicle types, and road surface types. This raw data was then carefully cleaned and preprocessed

To deal with the prediction task, we chose five distinct machine learning algorithms: Random Forest, Decision Tree, Naive Bayes, Support Vector Machine (SVM), and K-Nearest Neighbors (KNN). Each algorithm was selected for its specific strengths in handling different types of data distributions and complexities. The process is implemented step by step in Fig.3.1.

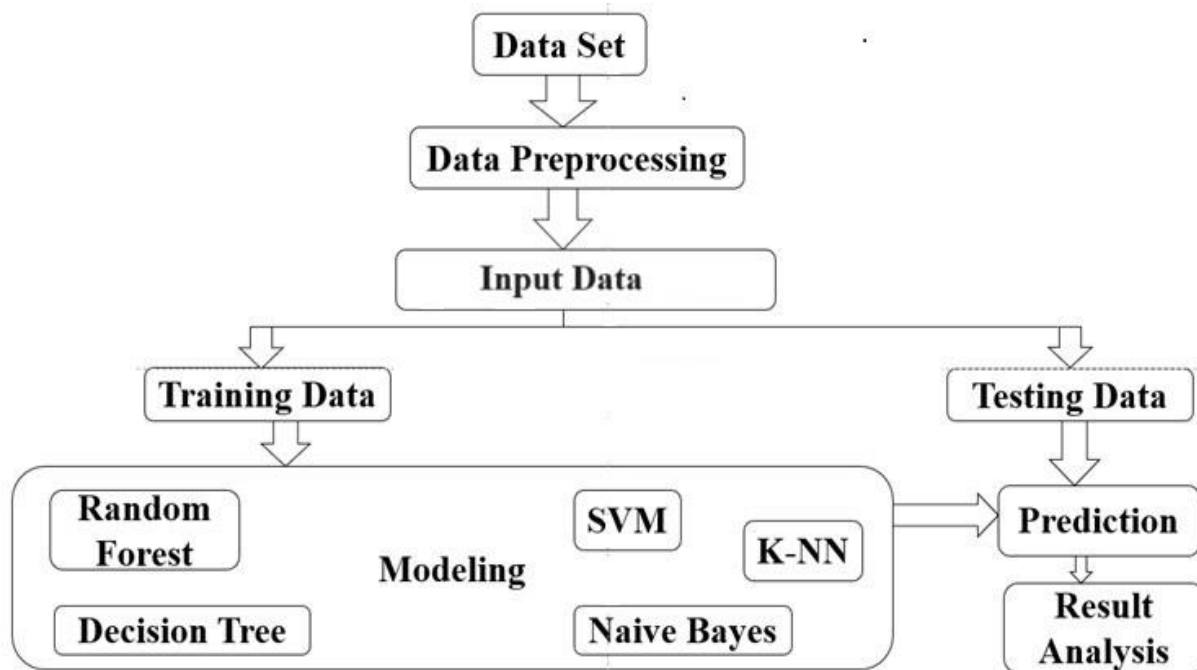


Fig. 3.1 Block Diagram of Proposed Method

3.2 IMPLEMENTATION

3.2.1 Dataset

Dataset Name: RADataset

Source: Kaggle

Description: The dataset on road accidents plays a pivotal role in our project's capacity to estimate accident probability accurately. Through an examination of this dataset, patterns, correlations, and risk factors associated with accidents can be discerned. The data furnishes crucial insights into variables such as weather conditions, road types, vehicle characteristics, and light conditions. By Leveraging machine learning algorithms, this project aims to develop predictive models that can assess the likelihood of accidents. These models hold promise for enhancing road safety measures and contributing to the reduction of accidents. The below Fig .3.2 is the dataset snippet which is used for training the machine learning model.

Type_of_vehicle	Road_surface	Light_conditions	Weather_conditions	Accident_Severity		
Automobile	Asphalt road	Daylight	Normal	Low Probability		
Public (> 4	Asphalt road	Daylight	Normal	Low Probability		
Lorry (41?;	Asphalt road	Daylight	Normal	High Probability		
Public (> 4	Earth road	Darkness	Normal	Low Probability		
	Asphalt road	Darkness	Normal	Low Probability		
		Daylight	Normal	Low Probability		
Automobile		Daylight	Normal	Low Probability		
Automobile	Asphalt road	Daylight	Normal	Low Probability		
Lorry (41?;	Earth road	Daylight	Normal	Low Probability		
Automobile	Asphalt road	Daylight	Normal	High Probability		
Public (13	Asphalt road	Daylight	Normal	High Probability		
Automobile	Earth road	Daylight	Normal	High Probability		
Public (> 4	Asphalt road	Daylight	Normal	Low Probability		
Lorry (41?;	Asphalt road	Daylight	Normal	Low Probability		
Automobile	Asphalt road	Daylight	Normal	High Probability		
Lorry (11?;	Asphalt road	Darkness	Raining	High Probability		
Public (13	Asphalt road	Darkness	Raining	Low Probability		
Public (13	Asphalt road	Darkness	Raining	High Probability		
	Asphalt road	Darkness	Raining	Low Probability		

Fig. 3.2 Snippet of Dataset

3.2.2 Data Preprocessing:

The code starts by importing necessary libraries. pandas (pd) is a powerful tool for data manipulation and analysis in Python. It allows us to load data from various sources (like CSV files in this case) and perform operations like viewing summaries, cleaning, and transforming the data. numpy (np) is another essential library for scientific computing. It provides efficient mathematical functions and data structures that are often used in machine learning algorithms. Finally, the warnings library helps suppress warnings that might clutter the output during data loading.

- i. **Data Acquisition:** Load the road accident information. contain numerous characteristics such as vehicle type, road conditions, and accident severity.
- ii. **Identifying Categorical Features:** Datasets include categorical attributes that indicate non-numerical attributes. In this case, features such as vehicle type, road conditions, daylight, and weather conditions are most likely classified. This allows us to better understand the distribution of categorical data and plan for appropriate encoding methods.
- iii. **Transforming Categorical Data (Label Encoding):** Since machine learning algorithms typically work best with numerical data, the code employs label encoding to transform categorical features. This method replaces the original category labels (e.g., "sunny" and "rainy") with numerical values (e.g., 0, 1). This allows machine learning models to understand the relationships between these features and the target variable more effectively. The steps of preprocessing is shown in Fig .3.3.

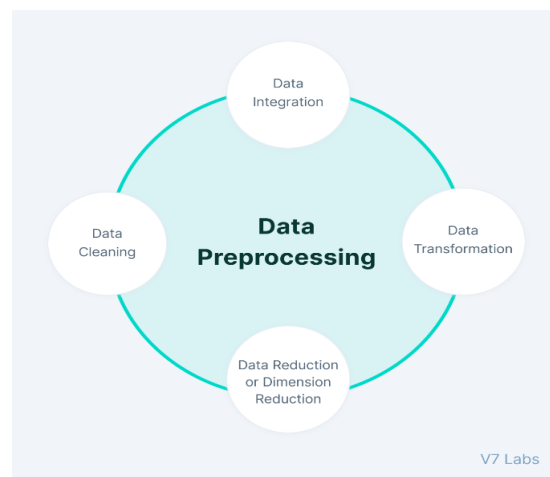


Fig. 3.3 Steps of Data Preprocessing

3.2.3 Training and Testing Data

Training and testing data for machine learning are primarily composed of two types of data. When creating and assessing machine learning models, each has a distinct role to play. Data from datasets is used to train machine learning algorithms. They learn things and find patterns. Make decisions and then assess them. In machine learning, training data is more powerful than testing data and provides the model. Since more data contributes to predicting models that are more accurate. A machine learning algorithm builds a decision-making model by identifying patterns in the data it receives from our records. Using algorithms, a business can make decisions based on its historical performance. It evaluates every instance that has already happened and its outcomes, then builds models to score and forecast how the current cases will turn out using this data. Over time, ML models become more and more trustworthy at making predictions the more data they have access to.

Training Data: Training data is used to train the machine learning model. A model's ability to produce correct predictions increases with the amount of training data it possesses. The model can learn from the training set of data and improve its prediction accuracy. The distribution of the training data ought to resemble the distribution of the real data that the model will utilize.

Testing Data: Testing data is employed to assess the performance of the model. The testing data is not revealed to the model until evaluation. This ensures that the model is unable to provide perfect forecasts by learning the testing data by heart. There are significant differences in the distribution of the testing and real-world data. The model's performance is evaluated by generating predictions on the testing data and comparing them to the real labels. We divided the data in this model into 70% training and 30% testing.

Various evaluation metrics are used to assess the model's performance on the testing data.

Accuracy: The percentage of correct predictions out of total predictions.

Precision: The ratio of true positive predictions to the total positive predictions, indicating the model's ability to avoid false positives.

Recall: The ratio of true positive predictions to the actual positive instances, indicating the model's ability to identify all relevant instances.

F1-Score: The harmonic mean of precision and recall, providing a balanced measure of model performance.

Confusion Matrix: A matrix that shows the number of true positives, true negatives, false positives, and false negatives, providing a detailed view of prediction outcomes.

3.2.4 Predictive Factors

These factors are derived from historical accident data and can include a wide range of variables that influence the occurrence and severity of accidents.

Weather Conditions: Weather conditions are one of the critical factors in predicting road accidents. Various weather elements can significantly affect road surface conditions, visibility, and driver behavior, thereby influencing the likelihood and severity of accidents.

Vehicle type: It is a significant predictive factor in road accident prediction models. It refers to the categorization of vehicles involved in accidents based on their characteristics and features. Different types of vehicles have varying sizes, weights, handling capabilities, and safety features, which can influence their likelihood of being involved in accidents and the severity of those accidents.

Light conditions: Light conditions play a crucial role in road safety and are important predictive factors in road accident prediction models. The level of illumination can significantly affect visibility, driver behavior, and the likelihood of accidents.

Road Type: Different types of roads, such as highways, urban streets, or rural roads, have distinct characteristics that affect accident risk. Factors like speed limits, road curvature, lane width, and road surface conditions can impact accident likelihood.

3.2.5 Technologies Used

Python

Python is a widely used general-purpose, high level programming language. It was initially designed by Guido van Rossum in 1991 and developed by Python Software Foundation. It

was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code.

Python is an interpreted, high-level, general-purpose programming language. Python features adynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural. It also has a comprehensive standard library.

It is the world's fastest growing and most popular programming language used by software engineers, analysts, data scientists, and machine learning engineers alike. It is used by sites like YouTube and Dropbox.

It supports functional and structured programming methods as well as OOP. It can be used as a scripting language or can be compiled to byte-code for building large applications. It provides very high-level dynamic data types and supports dynamic type checking. It supports automatic garbage collection.

It can be easily integrated with C, C++, and Java. Python uses whitespace indentation, rather than curly brackets or keywords, to delimit blocks. An increase in indentation comes after certain statements; a decrease in indentation signifies the end of the current block. Thus, the program's visual structure accurately represents the program's semantic structure.

Jupyter

Jupyter Notebook is an interactive web application widely used for data exploration, analysis, and visualization across various fields such as data science, machine learning, scientific research, and education. It provides a flexible environment where users can create documents, known as notebooks, containing live code, equations, visualizations, and explanatory text. One of its key features is the support for multiple programming languages, with Python being the most commonly used. Users can write and execute code in individual cells, allowing for iterative development and testing. Furthermore, Jupyter Notebook supports Markdown, a lightweight markup language, enabling users to format text, create headings, lists, tables, and mathematical equations within the notebook. This rich combination of code and narrative text makes it an ideal platform for creating reproducible and interactive reports, presentations, and tutorials.

Beyond its support for code execution and text formatting, Jupyter Notebook offers seamless integration with popular Python libraries such as NumPy, Pandas, Matplotlib, and Scikit-learn, enabling users to leverage their functionality for data manipulation, analysis, and visualization. Users can generate rich output, including plots, graphs, images, and HTML content, directly within the notebook, facilitating the creation of dynamic and interactive data visualizations. Moreover, Jupyter Notebook promotes collaboration and sharing by allowing users to export notebooks to various formats, including HTML, PDF, and slides, or publish them directly to platforms like GitHub or JupyterHub. With its active community of users and developers, extensive documentation, and integration with JupyterLab - an advanced integrated development environment (IDE) - Jupyter Notebook continues to be a powerful and versatile tool for data scientists, researchers, educators, and enthusiasts alike, fostering innovation and collaboration in the field of data science and beyond.

3.2.6 Algorithms

Random Forest

Random Forest is a powerful ensemble learning technique for classification and regression tasks. It combines the predictions of multiple decision trees to create a more robust and accurate model.

- i. **Bootstrap Aggregation (Bagging):** Instead of using the entire dataset to train a single decision tree, Random Forest draws multiple random samples (with replacement) from the original data. These samples are called bootstrap replicates. This injects randomness and helps prevent the trees from overfitting the specific training data.
- ii. **Decision Tree Building:** For each bootstrap replicate, a decision tree is constructed. These trees can have different depths and may use different features at each split. This diversity helps the forest to capture a wider range of patterns in the data.
- iii. **Random Feature Selection:** At each node of a decision tree in the forest, instead of considering all features for splitting, a random subset of features is chosen as candidates for the split. This further increases diversity among the trees and reduces the chance of overfitting to irrelevant features.

- iv. **Making Predictions:** In classification tasks (like predicting accident severity) the most frequent class predicted by the trees becomes the final prediction for the forest. In regression tasks, the average of the individual tree predictions is used as the final prediction.

The precision, recall, f1-score, and accuracy of the Random Forest are shown in Fig . 3.4.

```

-----
Random Forest Classifier:
accuracy : 0.85
confusion matrix : [[ 5 550]
 [ 14 3126]]
classification report:

```

			precision	recall	f1-score	support
	0	0.26	0.01	0.02		555
	1	0.85	1.00	0.92		3140
	accuracy			0.85		3695
	macro avg	0.56	0.50	0.47		3695
	weighted avg	0.76	0.85	0.78		3695

```

accuracy : 0.85
precision : 0.76
recall : 0.85
f1 : 0.78

```

Fig. 3.4 Random Forest

Decision Tree

A decision tree is a supervised learning algorithm that resembles a tree structure for classification and regression tasks. It works by recursively splitting the data based on features that best distinguish between different classes or predict a continuous value.

- i. **Building the Tree:** The algorithm starts with the entire dataset as the root node. It chooses the most informative feature (the one that best separates the data) to split the node into child nodes. This selection can be based on various measures like information gain (classification) or variance reduction (regression). The splitting process continues recursively on each child node, using the best remaining feature for further separation. The process stops when a stopping criterion is met, such as reaching a certain depth in the tree, having pure classes (classification) or minimal variance (regression) in a node, or having no more informative features to split on.

- ii. **Making Predictions:** Once the tree is built, a new data point (whose class or value needs to be predicted) is passed through the tree. At each node, the value of the corresponding feature in the data point is compared to the splitting threshold. The data point is directed to the left or right child node based on the comparison. This process continues until the data point reaches a leaf node, which represents the predicted class (classification) or predicted continuous value (regression).
- iii. **Information Gain (Classification):** This measure calculates how much "purity" (separation between classes) is gained by splitting on a particular feature. Higher information gain indicates a more informative split.

$$\text{Information}(\text{Feature A}) = \text{Entropy}(\text{Parent}) - \sum [\text{Entropy}(\text{Child}) * \text{Proportion}(\text{Child})] \quad (1)$$

- iv. **Gini Impurity (Classification):** Another common measure for calculating impurity, particularly for imbalanced datasets.

$$\text{Impurity}(\text{Parent}) = 1 - \sum [\text{Proportion}(\text{Class } i)^2] \quad (2)$$

The precision, recall, f1-score, and accuracy of the Decision Tree are shown in Fig . 3.5.

```

-----
Decision Tree Classifier:
accuracy : 0.84
confusion matrix : [[ 8 547]
 [ 37 3103]]
classification report:

```

			precision	recall	f1-score	support
	0	0.18	0.01	0.03		555
	1	0.85	0.99	0.91		3140
accuracy			0.84			3695
macro avg	0.51	0.50	0.47			3695
weighted avg	0.75	0.84	0.78			3695

```

accuracy : 0.84
precision : 0.75
recall : 0.84
f1 : 0.78

```

Fig. 3.5 Decision Tree

K-Nearest Neighbors (KNN)

K-Nearest Neighbors (KNN) is a non-parametric, supervised machine learning algorithm used for both classification and regression tasks.

It works by classifying a data point based on the similarity of its k-nearest neighbors in the training data.

- i. **Data Representation:** Each data point is represented as a feature vector, containing values for all features (e.g., vehicle type, speed, weather).
- ii. **Distance Metric:** A distance metric is chosen to measure the similarity between data points. Common choices include Euclidean distance, Manhattan distance, or Mahala Nobis distance.
- iii. **K Selection:** K, the number of nearest neighbors to consider, is a crucial parameter. A higher k value leads to smoother decision boundaries but might be more susceptible to noise. A lower k value can capture local variations but might be prone to overfitting.
- iv. **Classification (for accident severity prediction):** Calculate the distance between the new data point and all points in the training data. Identify the k nearest neighbors based on the chosen distance metric. Determine the most frequent accident severity class (e.g., high, low) among these k neighbors. Assign the new data point to the most frequent class, predicting its accident severity.
- v. **Euclidean Distance:**

$$\text{distance}(x_1, x_2) = \sqrt{\sum ((x_{1_i} - x_{2_i})^2 \text{ for } i \text{ in features})} \quad (3)$$

The precision, recall, f1-score, and accuracy of the K-NN are shown in Fig . 3.6.

```

-----
KNN Classifier:
accuracy : 0.84
confusion matrix : [[ 10  545]
 [ 64 3076]]
classification report:

```

			precision	recall	f1-score	support
	0	0.14	0.02	0.03		555
	1	0.85	0.98	0.91		3140
	accuracy			0.84		3695
	macro avg	0.49	0.50	0.47		3695
	weighted avg	0.74	0.84	0.78		3695

```

accuracy : 0.84
precision : 0.74
recall : 0.84
f1 : 0.78

```

Fig. 3.6 K-Nearest Neighbor(KNN)

Naive Bayes Classifier

The Naive Bayes classifier is a popular supervised learning algorithm for classification tasks. It works based on Bayes' theorem, a fundamental concept in probability theory that allows us to calculate the conditional probability of an event (accident severity in this case) occurring given the presence of certain features (vehicle type, road conditions, etc.).

The core formula for Naive Bayes classification is:

$$P(\text{Class} | \text{Features}) = (P(\text{Features} | \text{Class}) * P(\text{Class})) / P(\text{Features}) \quad (4)$$

The precision, recall, f1-score, and accuracy of the Naive Bayes are shown in Fig . 3.7.

```

-----
Naive Bayes Classifier:
accuracy : 0.84
confusion matrix : [[ 1 554]
 [ 45 3095]]
classification report:

```

			precision	recall	f1-score	support
	0	0.02	0.00	0.00		555
	1	0.85	0.99	0.91		3140
	accuracy			0.84		3695
	macro avg	0.43	0.49	0.46		3695
	weighted avg	0.72	0.84	0.78		3695

```

accuracy : 0.84
precision : 0.72
recall : 0.84
f1 : 0.78

```

Fig. 3.7 Naive Bayes

Support Vector Machine (SVM)

SVM is a powerful machine-learning algorithm for classification tasks. It aims to find a hyperplane in the feature space that best separates the data points belonging to different classes. Here's a breakdown of the algorithm with formulas and its application in your road accident severity prediction project.

- i. **Hyperplane Equation:** A hyperplane in n-dimensional space can be represented by the equation:

$$w^T * x + b = 0 \quad (5)$$

- ii. **Margin:** The margin between the hyperplane and a support vector is calculated as:

$$|w^T \cdot x_s + b| / \|w\| \quad (6)$$

The objective of SVM is to maximize this margin, which intuitively translates to finding the best separation between the classes. In summary, SVM aims to find the optimal hyperplane $w^T \cdot x + b = 0$ that separates the data points with the maximum margin while penalizing misclassifications based on the regularization parameter C . The use of kernel functions allows SVM to handle non-linearly separable data by mapping it to a higher-dimensional space where linear separation is possible.

The precision, recall, f1-score, and accuracy of the SVM are shown in Fig . 3.8.

```

-----
SVM Classifier:
accuracy : 0.85
confusion matrix : [[ 0 555]
 [ 0 3140]]
classification report:

```

				precision	recall	f1-score	support
	0	0.00	0.00	0.00	555		
	1	0.85	1.00	0.92	3140		
	accuracy			0.85	3695		
	macro avg	0.42	0.50	0.46	3695		
	weighted avg	0.72	0.85	0.78	3695		

```

accuracy : 0.85
precision : 0.72
recall : 0.85
f1 : 0.78

```

Fig. 3.8 Support Vector Machine(SVM)

In this chapter, we discussed the methodology of the project, the steps involved in the prediction, Data Preprocessing, Technologies used, and algorithms used for road accident prediction.

CHAPTER 4

RESULTS

In our road accident prediction project utilizing machine learning, we employed five algorithms: Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Naive Bayes, Decision Tree, and Random Forest. Through rigorous evaluation, we observed varying performances across these algorithms. SVM and Random Forest emerged as the top performers, exhibiting high accuracy, precision, recall, and F1 scores. They effectively predicted the probability of accidents based on input features such as weather conditions, vehicle type, road type, and light conditions.

Table. 4.1: Evaluation Metrics of Classification Algorithms

Algorithm	Accuracy	Precision	f1-score	Recall
Random Forest	0.85	0.76	0.78	0.85
Decision Tree	0.84	0.75	0.78	0.84
SVM	0.85	0.72	0.78	0.85
KNN	0.84	0.74	0.78	0.84
Naive Bayes	0.84	0.72	0.78	0.84

Table 4.1. represents the evaluation metrics of classification algorithms. The Random Forest algorithm emerged as the top-performing model in our road accident prediction project, showcasing the highest accuracy among the algorithms tested. So, in our project, the Random Forest algorithm is used in web development to predict the accident probability.

4.1 Web Development

A web page is designed to improve the user experience in this project. Offering a user-friendly platform, this web interface allows users to quickly and efficiently obtain predictive outputs without complexity. Users can easily input parameters such as light conditions, vehicle type, weather conditions, and road type to receive accurate predictions on the likelihood of road accidents occurring.

4.1.1 Running Flask Web Application

The project directory consists of several folders and files organized for the purpose of predicting of road accidents using machine learning techniques. Within the “static” directory are static assets such as CSS stylesheets with a subdirectory named “js” containing CSS files. In the “templates” directory, HTML templates used for rendering web pages for displaying the main interface.

The “app.py” file serves as the main Python script for running the web application. The “test.pkl” file stores a trained machine-learning model for predicting road accidents. Additionally, “RoadAccident.ipynb” is the Jupyter Notebook containing code for predicting road accidents.

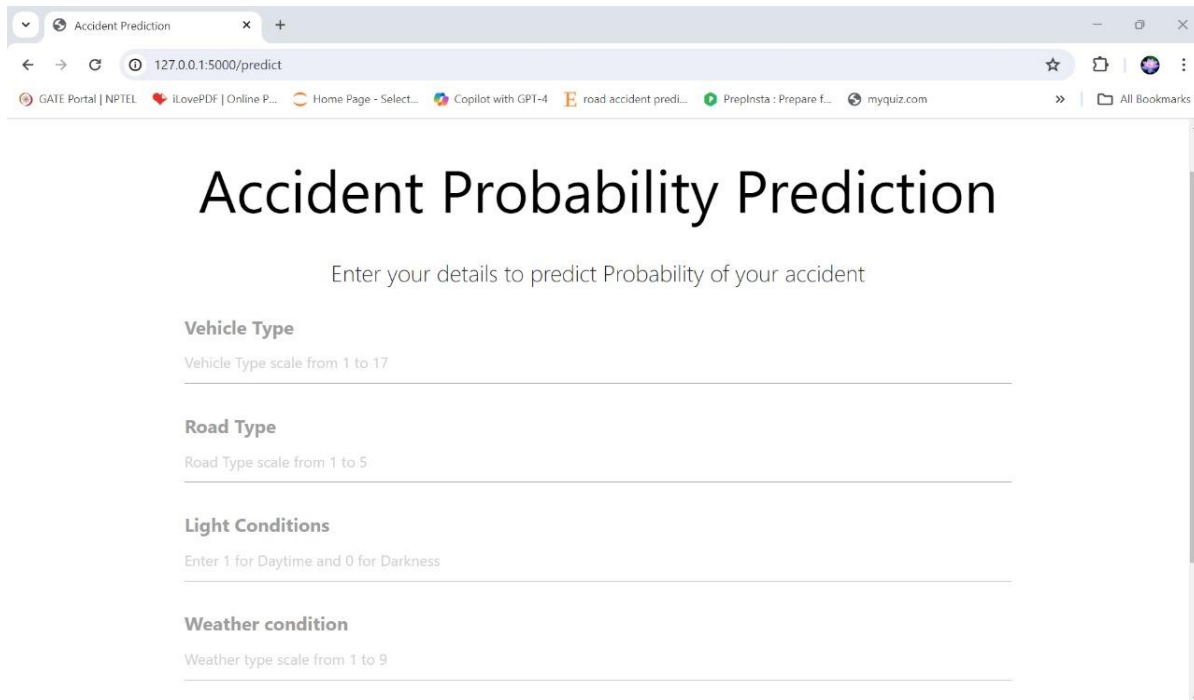
In order to start the Flask Application, we need to run the app.py file. After running the app.py we will get the URL `http://127.0.0.1:5000/` we can run this URL in any browser as shown in Fig. 4.1.

```
Python 3.6.7 (v3.6.7:6ec5cf24b7, Oct 20 2018, 13:35:33) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:\Users\Reshma Priya\OneDrive\Desktop\Accident\MAIN PROJECT\app.py
* Serving Flask app 'app' (lazy loading)
* Environment: production
[[31m  WARNING: This is a development server. Do not use it in a production deployment.
[[0m
[[2m  Use a production WSGI server instead.
[[0m
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

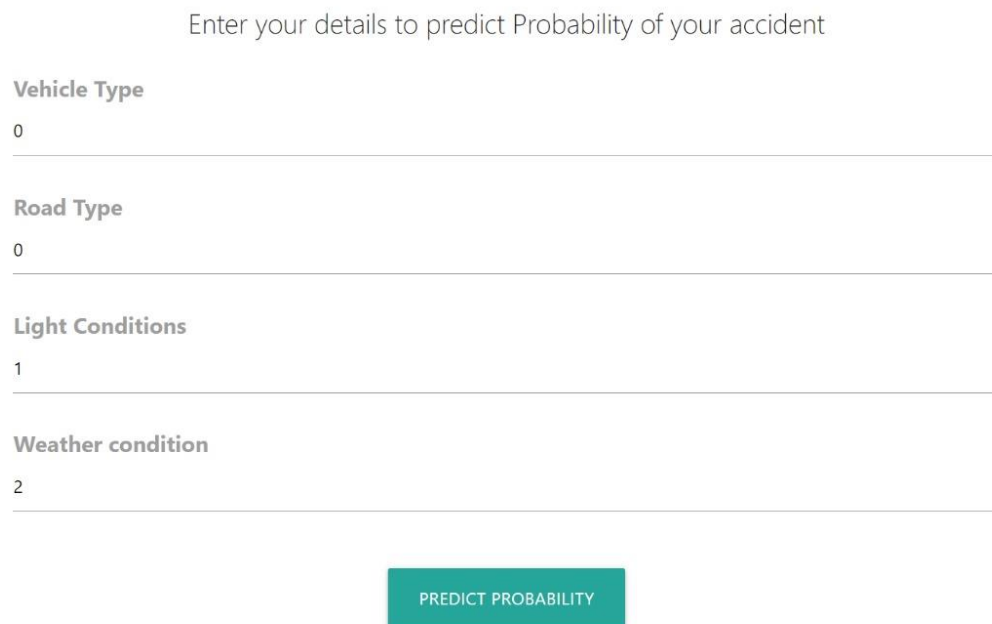
Fig. 4.1 Running the Application

4.1.2 Web Interface Results

Users can easily input the features for prediction. After inputting the features, the user has to click the “PREDICT PROBABILITY” button to generate the results as shown in Fig. 4.2.



The screenshot shows a web browser window with the title 'Accident Prediction'. The address bar shows the URL '127.0.0.1:5000/predict'. The page has a heading 'Accident Probability Prediction' and a subheading 'Enter your details to predict Probability of your accident'. There are four input fields: 'Vehicle Type' (scale 1 to 17), 'Road Type' (scale 1 to 5), 'Light Conditions' (1 for Daytime, 0 for Darkness), and 'Weather condition' (scale 1 to 9). All fields are currently empty.

Fig. 4.2 Web Page

The screenshot shows the same web page as Fig. 4.2, but with input values entered: 'Vehicle Type' is 0, 'Road Type' is 0, 'Light Conditions' is 1, and 'Weather condition' is 2. A green button labeled 'PREDICT PROBABILITY' is visible at the bottom.

Probability of accident is : Low

Fig. 4.3 Output for Low Probability of Accident

Fig. 4.3 consists of parameters Vehicle type – Automobile, Road Type – Asphalt roads, Light Conditions – Daylight, Weather conditions – Normal.

Enter your details to predict Probability of your accident

Vehicle Type
8

Road Type
2

Light Conditions
0

Weather condition
6

PREDICT PROBABILITY

Probability of accident is : High

Fig. 4.4 Output for High Probability of Accident

Fig. 4.4 consists of parameters Vehicle type – Public (> 45 seats), Road Type – Earth roads, Light Conditions – Darkness, Weather conditions – Snow.

Road accidents are more likely to occur under adverse conditions such as darkness, snow, and on less-maintained earth roads, especially when larger public vehicles are involved. Conversely, accidents are less probable under favorable conditions like daylight, normal weather, and well-maintained asphalt roads, particularly for smaller and more agile automobiles.

In this chapter, we discussed the result of the project and Web Development. Images along with a few test cases are fed to the web, and outputs are evaluated in this chapter.

CHAPTER 5

CONCLUSION AND FUTURE SCOPE

Road accidents have a profound impact on individuals and communities, highlighting the importance of proactive measures to reduce their occurrence. While it is each individual's responsibility to adopt safe driving practices, the role of road development authorities and automobile industries in creating safer infrastructure and vehicles cannot be understated. However, addressing the multifaceted nature of accidents requires a comprehensive approach that includes predictive modeling based on historical data and regulatory compliance. Our project successfully developed an application capable of efficiently predicting road accidents by leveraging machine learning algorithms. By analyzing factors such as vehicle types, light and weather conditions, and road types, our model can assess the risk probability of accidents in different areas with a high degree of accuracy.

The application of machine learning algorithms, including Random Forest, Decision Tree, SVM, KNN, and Naive Bayes, over a carefully curated dataset enabled us to create a robust predictive model. Moving forward, the integration of such predictive models into road safety initiatives can significantly contribute to accident reduction efforts. By utilizing data-driven approaches, authorities can prioritize resources and interventions effectively, leading to safer roads and reduced casualties. In conclusion, our project showcases the potential of machine learning in enhancing road safety and accident prediction.

The future of this project is, with more resources, continuous prediction, and alerts can be sent to the police for every location at regular intervals of time to take preventive measures. The web app can be incorporated with Google Maps which can be live-tracked by the police. A fully-fledged web app for user and police interaction can be published for use in real time. It can be used for Indian states or cities if proper data on accidents is provided by the Indian Government.

REFERENCES

- [1] Chen ZY, Chen CC. (2015). Identifying the stances of topic persons using a model-based expectationmaximization method. J. Inf. Sci. Eng 31(2): 573-595.
<http://dx.doi.org/10.1504/IJASM.2015.068609>.
- [2] Williams T, Betak J, Findley B. (2016). Text mining analysis of railroad accident investigation reports. In 2016 Joint Rail Conference. American Society of Mechanical Engineers V001T06A009 V001T06A009. <http://dx.doi.org/10.14299/ijser.2013.01>.
- [3] Sarkar S, Pateshwari V, Maiti J. (2017). Predictive model for incident occurrences in steel plant in India. In ICCCNT 2017, IEEE, pp. 1-5. <http://dx.doi.org/10.14299/ijser.2013.01>.
- [4] Zheng CT, Liu C, Wong HS. (2018). Corpus based topic diffusion for short clustering. Neurocomputing <http://dx.doi.org/10.1504/IJIT.2018.090859>.
- [5] Tessa KA. Kernel density estimation and K-means clustering to profile road accident hotspots. Elsevier. Accident Analysis and Prevention. 2009; 41:359–364.
- [6] A. Briz-Redon, F. Martinez-Ruiz, and F. Montes. Estimating the occurrence of traffic accidents near school locations: A case study from Valencia (Spain) including several approaches. Elsevier. Accident Analysis & Prevention. 2019;132.
- [7] De Ona. J, Lopez. G, Mujalli. R, Calvo.F. J. Analysis of traffic accidents on rural highways using latent class clustering and bayesian networks. Elsevier. Accident Analysis and Prevention. 2013;51:1-10.
- [8] Mahendra G, Dayananda R B. Vehicle rash drive control system. International Journal for Research in Engineering Application and Management. 2018;04(03):676-681.
- [9] Jamal Raiyn, Tomer Toledo. Real-time road traffic anomaly detection. Journal of Transportation Technologies. 2014; 4(3):256-266.
- [10] X. Gao, J. Wen, C Zhang. An improved random forest algorithm for predicting employee turnover. Hindawi. 2019; 4140707:12 p.
- [11] World Health Organization. Road Traffic Injuries. Available online: <https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries> (accessed on 20 June 2021).

PROJECT OUTCOMES MAPPED WITH PROGRAMME SPECIFIC OUTCOMES (PSOs) AND PROGRAMME OUTCOMES (POs)

	Application	Product	Research	Review
Classification of Project	✓		✓	

Note: Tick Appropriate category.

Project Outcomes

Outcome 1: Employ a systematic approach to design and develop engineering solutions for complex problems.

Outcome 2: Using modern technologies like Machine Learning.

Outcome 3: Evaluate ethical, environmental, legal, and security concerns throughout the project implementation process.

PROGRAMME SPECIFIC OUTCOMES (PSOs)

The ECE Graduates will be able to

PSO1: designing electronics and communication systems in the domains of VLSI, embedded systems, signal processing and RF communications, and applying modern tools.

PSO2: applying the contextual knowledge of Electronics and Communication Engineering to design, develop, analyze and test systems containing hardware and software components taking into societal, environmental, health, safety, legal, cultural, ethical and economical considerations.

PROGRAMME OUTCOMES (POs)

1. Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

3. Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Mapping Table

Project Outcomes	Programme Outcomes (POs)												PSOs	
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
Outcome 1	3	3	3	3	3	3	2	2	3	3	3	3	3	3
Outcome 2	3	3	3	3	3	3	2	2	3	3	3	3	3	3
Outcome 3	3	3	3	3	3	3	2	2	3	3	3	3	3	3

Note: Map each project outcomes with POs and PSOs with either 1 or 2 or 3 based on level of mapping as follows:

- 1-Slightly (Low) mapped
- 2-Moderately (Medium) mapped
- 3-Substantially (High) mapped

ROAD ACCIDENT PREDICTION USING MACHINE LEARNING

Viswanadhuni Reshma Priya¹, Shaik BB Ayesha², Vennapusa Venkata Vamsi³,

Ms. Ramya Palli⁴

Department of Electronics and Communication Engineering,

SR Gudlavalleru Engineering College, Gudlavalleru, Andhra Pradesh, India.

ABSTRACT:

Due to the escalating number of vehicles on roads, the incidence of daily traffic accidents is rising dramatically. This surge in accidents and fatalities underscores the urgency of accurately forecasting accident rates over time to facilitate informed decisions by transportation authorities. Analyzing accident occurrences can provide valuable insights for developing effective strategies to mitigate accidents. Although accidents often exhibit inherent uncertainty, patterns of regularity emerge over time in specific areas. Leveraging these patterns can enable us to develop robust accident prediction models. Our study focuses on examining the correlations between road accidents, road conditions, and environmental factors to enhance accident prediction accuracy.

Utilizing machine learning techniques, including the K-Nearest Neighbors (K-NN) Algorithm, Support Vector Machines (SVM), Random Forest Algorithm, Naive Bayes Algorithm, and Decision Tree Algorithm, this paper aims to construct a predictive model for accident occurrences. By harnessing the power of these algorithms, this paper improves understanding of accident dynamics and contributes to the development of proactive measures to reduce accidents on roads.

Keywords: Road Accident, Machine Learning (ML), Dataset.

INTRODUCTION:

Road accidents are a major hazard to public safety, causing injuries, deaths, and economic losses around the world. In recent times, advances in machine learning have created new opportunities for accident prevention and reduction. This paper use data-driven methodologies to anticipate the likelihood of traffic accidents grounded on a variety of contributing factors. Our exploration focuses on examining historical accident data, meteorological conditions, road infrastructure, and vehicle characteristics. Machine learning methods are employed in this project to construct accurate models capable of estimating accident probabilities. These predictive technologies hold the potential to assist traffic authorities, emergency services, and policymakers in devising visionary strategies to reduce accident rates.

This paper investigates five established machine learning techniques: Random Forest,

Decision Tree, Naive Bayes, K-nearest neighbors (KNN), and Support Vector Machine. The aim is to discern the most effective accident prediction algorithm through trial and review. The findings of this paper contribute to the enhancement of road safety initiatives, ultimately fostering a safer transportation environment for all.

LITERATURE REVIEW:

The field of road safety and accident prediction has witnessed significant advancements with the application of machine learning algorithms. Various studies have explored different methodologies and models to analyze factors influencing road accidents and predict accident severity.

Chen et al. [1], highways emerge as the predominant location for the occurrence of a significant portion of reported accidents.

Williams et al. [2], revealed that the age and driving experience of individuals significantly influence the frequency of accidents.

Sarkar et al. [3] conducted a comparative analysis investigating the prevalence of accidents across various road types. In addition to examining the factors contributing to accidents, their study revealed a higher incidence of accidents on highways compared to standard roads similar to [1].

Zheng et al. [4] conducted research focusing on the spectrum of injuries resulting from motor vehicle accidents. Additionally, they investigated the emotional state of the drivers involved, exploring its potential influence on the occurrence of accidents.

Tessa K. Anderson et al. [5] proposed a method aimed at identifying high-density casualty zones. This strategy involves implementing a clustering procedure to identify clusters where stochastic events are likely to occur. Consequently, it allows for the assessment of their occurrence within these clusters.

Briz-Redon et al. [6] took into account several factors including the log-straight model, driver characteristics, pedestrian traits, road traffic, and vehicle classifications. This comprehensive approach provides clear insights into the factors influencing casualties in school zones.

De Ona et al. [7] employed Latent Class Clustering and Bayesian methodologies in their examination of automobile collisions to identify the key determinants of casualty severity. The simultaneous application of these two techniques is particularly noteworthy as it unveils supplementary insights into the data.

Mahendra G et al. [8] devised a tool for detecting reckless driving on roadways and promptly notifying traffic authorities in the event of any speed infractions.

Jamal Raiyn et al. [9] present a model that identifies traffic incidents by analyzing the speed fluctuations of vehicles located both upstream and downstream of a specific point on the highway.

X. Gao et al. [10], a novel algorithm termed Weighted Quantitative Random Forest (WQRF) was introduced. This algorithm aims to forecast employee salary turnover within various industries. The predictive model incorporates several key features, including overtime hours, age, monthly

income, distance from home, and tenure within the company. These factors collectively contribute to the model's ability to anticipate shifts in employee salary turnover rates.

Our paper builds upon these research findings by employing a comparative study of machine learning algorithms including Random Forest, Decision Tree, SVM, KNN, and Naive Bayes. Through rigorous data preprocessing, feature selection, and model training, this project aims to develop a robust predictive model for road accident severity prediction. By synthesizing insights from existing literature and leveraging state-of-the-art machine learning techniques, this project seeks to contribute to the advancement of road safety measures and accident prevention strategies. This paper aims to provide valuable insights for stakeholders in traffic management, law enforcement agencies, and policymakers to make informed decisions and improve road safety for all.

PROPOSED METHODOLOGY:

In the context of road accident prediction, the process involves several key steps. First, relevant accident data is collected and pre-processed, addressing issues like missing values and outliers. Feature engineering follows, where meaningful features are extracted from the data. The dataset is then split into training and testing subsets. Next, machine learning models (including Random Forest, Decision Tree, Naive Bayes, KNN, and SVM) are selected and trained using the training data. Model evaluation metrics, such as accuracy and precision, guide the choice of the best-performing model. Applying the chosen model(s) to the testing data yields accident probability predictions. Finally, result analysis provides insights for decision-makers and safety improvements. Overall, this process aims to enhance road safety and prevent accidents. The architecture of accident prediction is shown in Fig.1.

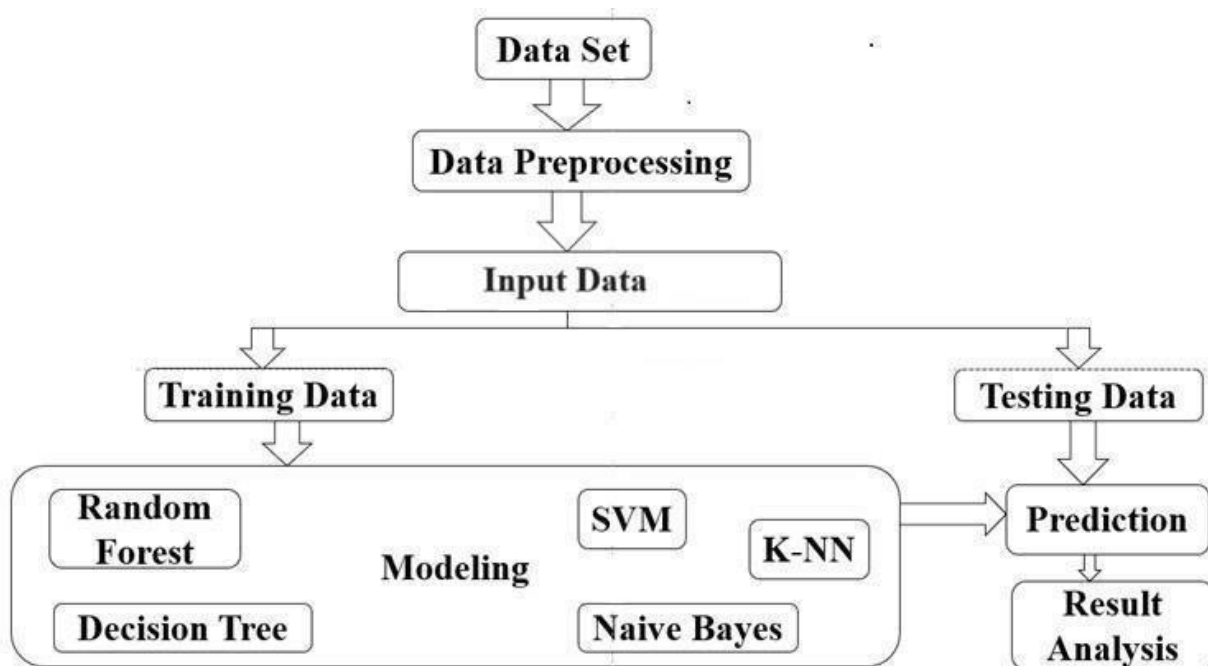


Fig.1: Accident Prediction Architecture

Dataset: The dataset on road accidents plays a pivotal role in our paper's capacity to estimate accident probability accurately. Through an examination of this dataset, patterns, correlations, and risk factors associated with accidents can be discerned. The data furnishes crucial insights into variables such as weather conditions, road types, vehicle characteristics, and light conditions. By Leveraging machine learning algorithms, this paper aims to develop predictive models that can assessthe likelihood of accidents. These models hold promise for enhancing road safety measures and contributing to the reduction of accidents.

Data Preprocessing: The code starts by importing necessary libraries. pandas(pd) is a powerful tool for data manipulation and analysis in Python. It allows us to load data from various sources (like CSV files in this case) and perform operations like viewing summaries, cleaning, and transforming the data. NumPy (np) is another essential library for scientific computing. It provides efficient mathematical functions and data structures that are often used in machine learning algorithms. Finally, the warnings library helps suppress warnings that might clutter the output during data loading.

- i. **Data Acquisition:** Load the road accident information. contain numerous characteristics such as vehicle type, road conditions, and accident severity.
- ii. **Identifying Categorical Features:** Datasets include categorical attributes that indicate non-numerical attributes. In this case, features such as vehicle type, road conditions, daylight, and weather conditions are most likely classified. This allows us to better understand the distribution of categorical data and plan for appropriate encoding methods.
- iii. **Transforming Categorical Data (Label Encoding):** Since machine learning algorithms typically work best with numerical data, the code employs label encoding to transform categorical features. This method replaces the original category labels (e.g., "sunny" and "rainy") with numerical values (e.g., 0, 1). This allows machine learning models to understand the relationships between these features and the target variable more effectively.

TRAINING AND TESTING DATA:

Splitting the data into two sets: training and testing. The training set (typically 70% of the data) is used to train the machine learning models, while the testing set (the remaining 30%) is used to evaluate their performance on unseen data. This helps prevent overfitting, where the models become too good at fitting the training data but perform poorly on new data.

Random Forest:

Random Forest is a powerful ensemble learning technique for classification and regression tasks. It combines the predictions of multiple decision trees to create a more robust and accurate

model.

- i. **Bootstrap Aggregation (Bagging):** Instead of using the entire dataset to train a single decision tree, Random Forest draws multiple random samples (with replacement) from the original data. These samples are called bootstrap replicates. This injects randomness and helps prevent the trees from overfitting the specific training data.
- ii. **Decision Tree Building:** For each bootstrap replicate, a decision tree is constructed. These trees can have different depths and may use different features at each split. This diversity helps the forest to capture a wider range of patterns in the data.
- iii. **Random Feature Selection:** At each node of a decision tree in the forest, instead of considering all features for splitting, a random subset of features is chosen as candidates for the split. This further increases diversity among the trees and reduces the chance of overfitting to irrelevant features.
- iv. **Making Predictions:** In classification tasks (like predicting accident severity) the most frequent class predicted by the trees becomes the final prediction for the forest. In regression tasks, the average of the individual tree predictions is used as the final prediction.

Decision Tree:

A decision tree is a supervised learning algorithm that resembles a tree structure for classification and regression tasks. It works by recursively splitting the data based on features that best distinguish between different classes or predict a continuous value.

- i. **Building the Tree:** The algorithm starts with the entire dataset as the root node. It chooses the most informative feature (the one that best separates the data) to split the node into child nodes. This selection can be based on various measures like information gain (classification) or variance reduction (regression). The splitting process continues recursively on each child node, using the best remaining feature for further separation. The process stops when a stopping criterion is met, such as reaching a certain depth in the tree, having pure classes (classification) or minimal variance (regression) in a node, or having no more informative features to split on.
- ii. **Making Predictions:** Once the tree is built, a new data point (whose class or value needs to be predicted) is passed through the tree. At each node, the value of the corresponding feature in the data point is compared to the splitting threshold. The data point is directed to the left or right child node based on the comparison. This process continues until the data point reaches a leaf node, which represents the predicted class (classification) or predicted continuous value (regression).
- iii. **Information Gain (Classification):** This measure calculates how much "purity"

(separation between classes) is gained by splitting on a particular feature. Higher information gain indicates a more informative split.

$$\text{Information (Feature A)} = \text{Entropy (Parent)} - \sum [\text{Entropy (Child)} * \text{Proportion (Child)}] \quad (1)$$

- iv. **Gini Impurity (Classification):** Another common measure for calculating impurity, particularly for imbalanced datasets.

$$\text{Impurity (Parent)} = 1 - \sum [\text{Proportion (Class } i) ^2] \quad (2)$$

K-Nearest Neighbors (KNN):

K-Nearest Neighbors (KNN) is a non-parametric, supervised machine learning algorithm used for both classification and regression tasks. It works by classifying a data point based on the similarity of its k-nearest neighbors in the training data.

- i. **Data Representation:** Each data point is represented as a feature vector, containing values for all features (e.g., vehicle type, speed, weather).
- ii. **Distance Metric:** A distance metric is chosen to measure the similarity between data points. Common choices include Euclidean distance, Manhattan distance, or Mahala Nobis distance.
- iii. **K Selection:** K, the number of nearest neighbors to consider, is a crucial parameter. A higher k value leads to smoother decision boundaries but might be more susceptible to noise. A lower k value can capture local variations but might be prone to overfitting.
- iv. **Classification (for accident severity prediction):** Calculate the distance between the new data point and all points in the training data. Identify the k nearest neighbors based on the chosen distance metric. Determine the most frequent accident severity class (e.g., high, low) among these k neighbors. Assign the new data point to the most frequent class, predicting its accident severity.
- v. **Euclidean Distance:**

$$\text{distance (x1, x2)} = \sqrt{\sum ((x1_i - x2_i)^2 \text{ for } i \text{ in features})} \quad (3)$$

Naive Bayes Classifier: A Probabilistic Approach

The Naive Bayes classifier is a popular supervised learning algorithm for classification tasks. It works based on Bayes' theorem, a fundamental concept in probability theory that allows us to calculate the conditional probability of an event (accident severity in this case) occurring given the presence of certain features (vehicle type, road conditions, etc.).

The core formula for Naive Bayes classification is:

$$P(\text{Class} | \text{Features}) = (P(\text{Features} | \text{Class}) * P(\text{Class})) / P(\text{Features}) \quad (4)$$

Support Vector Machine (SVM):

SVM is a powerful machine-learning algorithm for classification tasks. It aims to find a hyperplane in the feature space that best separates the data points belonging to different classes. Here's a breakdown of the algorithm with formulas and its application in your road accident severity prediction project.

- i. **Hyperplane Equation:** A hyperplane in n-dimensional space can be represented by the equation:

$$w^T \cdot x + b = 0 \quad (5)$$

- ii. **Margin:** The margin between the hyperplane and a support vector is calculated as:

$$|w^T \cdot x_s + b| / \|w\| \quad (6)$$

The objective of SVM is to maximize this margin, which intuitively translates to finding the best separation between the classes. In summary, SVM aims to find the optimal hyperplane $w^T \cdot x + b = 0$ that separates the data points with the maximum margin while penalizing misclassifications based on the regularization parameter C. The use of kernel functions allows SVM to handle non-linearly separable data by mapping it to a higher-dimensional space where linear separation is possible.

RESULTS:

Table 1: Evaluation Metrics of Classification Algorithms

Algorithm	Accuracy	Precision	f1-score	Recall
Random Forest	0.85	0.76	0.78	0.85
Decision Tree	0.84	0.75	0.78	0.84
SVM	0.85	0.72	0.78	0.85
KNN	0.84	0.74	0.78	0.84
Naive Bayes	0.84	0.72	0.78	0.84

Table 1. represents the evaluation metrics of classification algorithms. The Random Forest algorithm emerged as the top-performing model in our road accident prediction project, showcasing the highest accuracy among the algorithms tested. So, in our project, the Random Forest algorithm is used in web development to predict the accident probability.

Web Development

A web page is designed to improve the user experience in this project. Offering a user- friendly platform, this web interface allows users to quickly and efficiently obtain predictive outputs without complexity. Users can easily input parameters such as light conditions, vehicle type, weather

conditions, and road type to receive accurate predictions on the likelihood of road accidents occurring. we will get the URL <http://127.0.0.1:5000/> for the web page and we can run this URL in any browser. The web page is shown in Fig. 2.

Accident Probability Prediction

Enter your details to predict Probability of your accident

Vehicle Type
Vehicle Type scale from 1 to 17

Road Type
Road Type scale from 1 to 5

Light Conditions
Enter 1 for Daytime and 0 for Darkness

Weather condition
Weather type scale from 1 to 9

PREDICT PROBABILITY

Fig.2: web page

Enter your details to predict Probability of your accident

Vehicle Type
0

Road Type
0

Light Conditions
1

Weather condition
2

PREDICT PROBABILITY

Probability of accident is : Low

Fig.3: Output for Low Probability of Accident

Fig. 3 consists of parameters Vehicle type – Automobile, Road Type – Asphalt roads, Light Conditions – Daylight, and Weather conditions – Normal.

Enter your details to predict Probability of your accident

Vehicle Type
8

Road Type
2

Light Conditions
0

Weather condition
6

PREDICT PROBABILITY

Probability of accident is : High

Fig.4: Output for High Probability of Accident

Fig.4 consists of parameters Vehicle type – Public (> 45 seats), Road Type – Earth roads, Light Conditions – Darkness, and Weather conditions – Snow.

CONCLUSION

Road accidents have a profound impact on individuals and communities, highlighting the importance of proactive measures to reduce their occurrence. While it is each individual's responsibility to adopt safe driving practices, the role of road development authorities and automobile industries in creating safer infrastructure and vehicles cannot be understated. However, addressing the multifaceted nature of accidents requires a comprehensive approach that includes predictive modeling based on historical data and regulatory compliance. Our project successfully developed an application capable of efficiently predicting road accidents by leveraging machine learning algorithms. By analyzing factors such as vehicle types, light and weather conditions, and road types, our model can assess the risk probability of accidents in different areas with a high degree of accuracy.

The application of machine learning algorithms, including Random Forest, Decision Tree, SVM, KNN, and Naive Bayes, over a carefully curated dataset enabled us to create a robust predictive model. Moving forward, the integration of such predictive models into road safety initiatives can significantly contribute to accident reduction efforts. By utilizing data-driven

approaches, authorities can prioritize resources and interventions effectively, leading to safer roads and reduced casualties. In conclusion, our project showcases the potential of machine learning in enhancing road safety and accident prediction.

REFERENCES

- [1] Chen ZY, Chen CC. (2015). Identifying the stances of topic persons using a model-based expectationmaximization method. *J. Inf. Sci. Eng* 31(2): 573-595.
<http://dx.doi.org/10.1504/IJASM.2015.068609>.
- [2] Williams T, Betak J, Findley B. (2016). Text mining analysis of railroad accident investigation reports. In 2016 Joint Rail Conference. American Society of Mechanical Engineers V001T06A009 V001T06A009. <http://dx.doi.org/10.14299/ijser.2013.01>.
- [3] Sarkar S, Pateshwari V, Maiti J. (2017). Predictive model for incident occurrences in steel plant in India. In ICCCNT 2017, IEEE, pp. 1-5. <http://dx.doi.org/10.14299/ijser.2013.01>.
- [4] Zheng CT, Liu C, Wong HS. (2018). Corpus based topic diffusion for short clustering. *Neurocomputing* <http://dx.doi.org/10.1504/IJIT.2018.090859>.
- [5] Tessa KA. Kernel density estimation and K-means clustering to profile road accident hotspots. *Elsevier. Accident Analysis and Prevention*. 2009; 41:359–364.
- [6] A. Briz-Redon, F. Martinez-Ruiz, and F. Montes. Estimating the occurrence of traffic accidents near school locations: A case study from Valencia (Spain) including several approaches. *Elsevier. Accident Analysis & Prevention*. 2019;132.
- [7] De Ona. J, Lopez. G, Mujalli. R, Calvo.F. J. Analysis of traffic accidents on rural highways using latent class clustering and bayesian networks. *Elsevier. Accident Analysis and Prevention*. 2013;51:1-10.
- [8] Mahendra G, Dayananda R B. Vehicle rash drive control system. *International Journal for Research in Engineering Application and Management*. 2018;04(03):676-681.
- [9] Jamal Raiyn, Tomer Toledo. Real-time road traffic anomaly detection. *Journal of Transportation Technologies*. 2014; 4(3):256-266.
- [10] X. Gao, J. Wen, C Zhang. An improved random forest algorithm for predicting employee turnover. *Hindawi*. 2019; 4140707:12 p.
- [11] World Health Organization. Road Traffic Injuries. Available online: <https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries> (accessed on 20 June 2021).