

ABSTRACT

Road accidents are one of the most regrettable hazards in this hectic world. Road accidents lead to numerous casualties, injuries, and fatalities each year, as well as significant economic losses. There are many factors that contribute to road accidents, especially those related to the environment, vehicles and the travelers. Despite many precautionary measures and safety measures to reduce road accidents it remains one of the uncontrollable. By analyzing the severity of the road accidents that happened in the past, and the factors that caused it, it is possible to take precautionary measures to reduce the road accidents rate significantly in the future.

This project implements developing a machine learning model which can classify the severity of the accident, based on the accident influential factors. Various machine learning classifiers such as Decision tree (DT), K- Nearest Neighbour (KNN), Random Forest (RF) and Gradient Boosting Classifiers(GBC) used to develop a predictive model. However, results show that a Gradient boosting algorithm is capable of 90% accurately detecting the road accident severity. The results of the comparative study suggest the Gradient Boosting Classifier could be a useful tool for forecasting accident severity and the hotspots. Further, the model used in the web portal to develop an intelligent system for the accident severity prediction.

LIST OF CONTENTS

S. NO	TITLE	PG. No
1	INTRODUCTION	6
	1.1.1 Problem Definition	7
	1.2 Objective	7
2	LITERATURE SURVEY	8
2.1	Related Works	8
3	METHODOLOGY	11
3.1	Proposed System	11
	3.1.1 DataSet	12
	3.1.2 Data Preprocessing	12
	3.1.3 Building Machine Learning Model	15
	3.1.4 Training and Testing	15
	3.1.5 Result Analysis	15
3.2	Graphical User Interface	15
	3.2.1 Save The Machine Learning Model	15
3.3	Tools Used	16
	3.3.1 Python	16
	3.3.2 Jupyter Notebook	18
	3.3.2 Flask	19
	3.3.4 HTML	20
	3.3.5 CSS	20
	3.3.6 JavaScript	20
	3.3.7 MySql	21
	3.3.8 Spyder	21

3.4	Algorithm	22
	3.4.1 Decision Tree Learning	22
	3.4.2 K-Nearest Neighbor	24
	3.4.3 Random Forest	25
	3.4.4 Gradient Boosting	26
4	RESULTS AND DISCUSSIONS	28
4.1	Data Preprocessing Result	28
4.2	Performance Metrics	29
	4.2.1 Confusion Matrix	29
	4.2.2 Accuracy	30
	4.2.3 Error Rate	30
	4.2.4 Precision	30
	4.2.5 Recall	30
	4.2.6 Specificity	31
	4.2.7 ROC Curve	31
	4.2.8 AUC	31
	4.2.9 Performance of the classifiers	31
	4.2.10 Comparative Analysis	33
	4.2.11 Severity Prediction	34
4.3	Graphical User Interface	34
5	CONCLUSION	39
5.1	5.1 Advantages	39
6	FUTURE SCOPE	40
7	REFERENCES	41

LIST OF FIGURES

S. NO	TITLE
3.1	Block diagram of the proposed system
3.3	Block diagram of web portal
3.3	Dataset
3.4	Label Encoding
3.5	Decision Tree
3.6	K-nearest Neighbor
3.7	Random Forest Classifier
3.8	Gradient Boosting Classifier
4.1	Dataset before preprocessing
4.2	Dataset after preprocessing
4.3	Confusion Matrix
4.4	Performance of the DT
4.5	Performance of the KNN
4.6	Performance of the RF
4.7	Performance of the GB
4.8	Severity Prediction
4.9	User Registration
4.10	User Login
4.11	User Input
4.12	Download The Report
4.13	Accident Factors
4.14	Previous Predictions
4.15	Send Report Via Mail

LIST OF ABBREVIATION

DT	Decision Tree
RF	Random Forest
KNN	K- Nearest Neighbour
GB	Gradient Boosting
SVM	Support Vector Machine
ANN	Artificial Neural Network
CNN	Convolutional Neural Network
HTML	Hyper Text Markup Language
CSS	Cascading Style Sheet
DOM	Document Object Model
RDBMS	Relational Database Management System
TPR	True Positive Rate
FPR	False Positive Rate

Chapter 1

INTRODUCTION

Road Traffic Accident (RTA) is an unexpected event that unintentionally occurs on the road which involves vehicle and/or other road users that causes casualty or loss of property. Over 90% of the world's fatalities on roads occur in low and middle income countries which account for only 48% of world's registered vehicles. The financial loss, which is about US\$518 billion, is more than the development assistance allocated for these countries. While developed rich nations have stable or declining road traffic death rates through coordinated correcting efforts from various sectors, developing countries are still losing 13% of their gross national product (GNP) due to the endemic of traffic casualties. World Health Organization (WHO) fears, unless immediate action is taken, road crashes will rise to the fifth leading cause of death by 2030, resulting in an estimated 2.4 million fatalities per year.

Accidental severity analysis involves three factors that are number of injuries, number of casualties, and destruction of property. Authors take severity level independently and consider four options like light injury, severe injury, fatal, and property damage. The accidental severity level is defined as injury, possible injury, and property damage. In the last two decades, accidental severity is one of the popular research areas. Researchers were applying different statistical approaches for road accident classification. These techniques help in analyzing the cause of road accidents.

To uncover the underlying relationship between a road accident and the contributing factors, ML based models have been studied in recent days. The primary focus of this study is to analyze a set of widely used ML models in terms of their prediction accuracy with the variation of contributing factors. Therefore, this research

aims to study different ML algorithms and compare the performance of these algorithms which can be considered to predict road accidents and its severity.

1.1 Problem Definition

There are many factors that contribute to road accidents, especially those related to the environment, vehicles and the travelers. Despite many precautionary measures to reduce road accidents, it remains one of the uncontrollable. Many factors are associated with traffic accidents, some of which are more significant than others in determining the severity of accidents. Identifying the accident severity and its factors can be given a solution for reducing the risk of future road accidents. Successful development of a countermeasure requires a clear understanding of where it can potentially break the chain of events leading to traumatic injury on the road. The use of big traffic data and artificial intelligence may help develop a promising solution to predict or reduce the risk of road accidents. Machine learning techniques are playing a vital role in predicting the future events based on the historical data. Developing the machine learning classifier for the road accident severity prediction can help to identify the accident severity and developing countermeasures and alerting systems for the future preventions.

1.2 Objective

The project's major purpose is to:

- Identify the group of factors lead to the road accident
- Develop a system to identify the road accident severity
- Deploying a model with higher accuracy and lower error rate
- Developing countermeasures for accident injuries
- Developing E-mail based road accident tracing system

Chapter 2

LITERATURE SURVEY

A literature review is an overview of the previously published works on a specific topic. The term can refer to a full scholarly paper or a section of a scholarly work such as a book, or an article. It is a comprehensive summary of previous research on a topic. The literature review surveys scholarly articles, books, and other sources relevant to a particular area of research. The review should enumerate, describe, summarize, objectively evaluate and clarify this previous research. The literature review acknowledges the work of previous researchers, and in so doing, assures the reader that your work has been well conceived. It is assumed that by mentioning a previous work in the field of study, that the author has read, evaluated, and assimilated that work into the work at hand. A literature review creates a "landscape" for the reader, giving her or him a full understanding of the developments in the field. This landscape informs the reader that the author has indeed assimilated all (or the vast majority of) previous, significant works in the field into her or his research.

2. 1 Related Works

Mubariz et al. [1] analyzed the severity of the accidents based on decision level fusion of machines and deep learning models. Top features affecting accidental severity include distance, temperature, wind_Chill, humidity, visibility, and wind direction. This study presents an ensemble of machine learning and deep learning models by combining Random Forest and Convolutional Neural Network called RFCNN for the prediction of road accident severity. The performance of the model is compared with several base learner classifiers.

Sachin et al. [2] conducted accident data analysis to identify the main factors associated with a road and traffic accident. The author used K-modes clustering technique as a preliminary task for segmentation of 11,574 road accidents on the road network of Dehradun (India) between 2009 and 2014. Association rule mining is used to identify the various circumstances that are associated with the occurrence of an accident for both the entire data set (EDS) and the clusters identified by K-modes clustering algorithm. The findings of cluster based analysis and entire data set analysis are then compared. The results reveal that the combination of k mode clustering and association rule mining is very inspiring as it produces important information that would remain hidden if no segmentation has been performed prior to generating association rules.

Shakil et al. [3] developed a machine learning model for identifying the severity of the accident by considering the human factors such as alcohol, drug, age, and gender are often ignored when determining accident severity. In this work single and ensemble mode machine learning (ML) methods compared their performance in terms of prediction accuracy, precision, recall, F1 score, area under the receiver operator characteristic (AUROC).

Max Cameron [4] analyzed the accident data to develop target groups for developing the accident countermeasures. The High risk group, high accident severity groups and accident involved clusters are identified and used to develop the accident countermeasures. The author identified each category such as vehicle, driver, passenger, environmental factors and the leading accident factors among them. All the high risk factors are clustered to create a countermeasure for each category.

Mohamed et al. [5] performed the data analytics on accident injury data in order to identify the high risk factors that lead to the severe accidents. The author applied advanced data analytics methods to predict injury severity levels and evaluate their performance. The results of this work identified that tree based techniques such as XGBoost outperform regression based ones, such as ANN.

Mehdizadeh et al. [6] presented a comprehensive review on data analytic methods in road safety. Analytics models can be grouped into two categories: predictive or explanatory models that attempt to understand and quantify crash risk and (b) optimization techniques that focus on minimizing crash risk through route/path selection and rest-break scheduling. Their work presented publicly available data sources and descriptive analytic techniques (data summarization, visualization, and dimension reduction) that can be used to achieve safer-routing and provide code to facilitate data collection/exploration by practitioners/researchers.

Sharma et al. [7] applied support vector machines with different Gaussian kernel functions for crash to extract important features related to accident occurrence. The paper compared neural networks with support vector machines. The paper reported that SVMs are superior in accuracy. However, the SVMs method has the same disadvantages of ANN in traffic accident severity prediction.

Ma et al. [8] proposed the XGBoost based framework which analyzed the relationship between collision, time and environmental and spatial factors and fatality rate. Results show that the proposed method has the best modeling performance compared with other machine learning algorithms. The paper identified eight factors that have an impact on traffic fatality.

Chapter 3

METHODOLOGY

3. 1 Proposed System

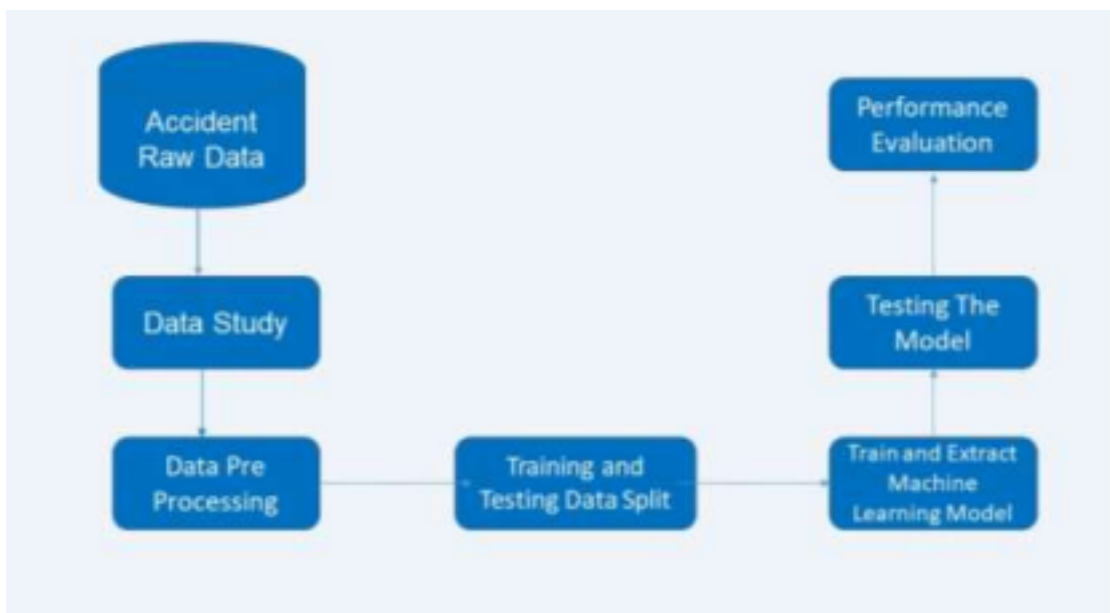


Fig 3.1 : Block diagram of the proposed system

This project classifies the road accident as slight or serious. The flow of the project is shown in figure 3.1. This project consists of five main phases.

1. Conducting descriptive study on the accident data
2. Pre-processing the data using grouping and label encoding
3. Building the machine learning classification model

4. Performance evaluation

5. Developing intelligent web portal for accident severity classification 6



Fig 3.2 : Block diagram of web portal

The steps involved in the intelligence web portal are described in the figure 3.2.

3.1.1 Dataset

This project was implemented using the Accident Severity dataset from the kaggle. The dataset contains 3057 accident samples. Each sample contains 14 predictive attributes and 1 target attribute. The dataset contains the accident data from 01-01-2009 to 31-12-2009. There are no null values present in the dataset. The target attribute contains two values: slight accident and serious accident. The dataset contains 321 serious data and 2736 slight accident data.

The figure 3.3 describes the predictive attributes that are available in the accident data and its description.

Variables	Description	Data Type	Scale	Null Value
Reference No	Accident identity number	Integer	Serial number	No
Easting	Easting point	Integer	Map point	No
Northing	Northing point	Integer	Map point	No
Number of Vehicles	Number of vehicle in the spot	Integer	Vehicle count	No
Accident Date	Date of accident happened	Date	Date	No
Time (24hr)	Time of accident happened	Time	Time	No
1st Road Class	Road type	Varchar	Category	No
Road Surface	Surface type of the road	Varchar	Category	No
Lighting Conditions	Lighting condition in the spot	Varchar	Category	No
Weather Conditions	Weather condition in the spot	Varchar	Category	No
Casualty Class	Casualty type (Driver..ect)	Varchar	Category	No
Sex of Casualty	Sex of the casualty	Varchar	Category	No
Age of Casualty	Age of the casualty	Integer	Age in years	No
Type of Vehicle	Type of the vehicle	Varchar	Category	No
Severity	Accident severity	Varchar	Category	No

Fig 3.3 Dataset

3.1.2 Data Preprocessing

Data preprocessing is a data mining technique which is used to transform the raw data in a useful and efficient format. Data preprocessing transforms the data into a format that is more easily and effectively processed in data mining, machine learning and other data science tasks. The techniques are generally used at the earliest stages of the machine learning and AI development pipeline to ensure accurate results.

Three data preprocessing tasks were conducted in the accident data

1. Data reduction
2. Encoding the categorical data
3. Dropping the unwanted columns

Some attribute values can be combined into a single variable. By grouping those values into the categories will help to reduce the number of attributes. The same thing is applicable to reduce the number of categories in an attribute.

Consider the ‘Type of vehicle’ column in the accident data [M/cycle 50cc and under', 'Motorcycle over 125cc and up to 500cc', 'Motorcycle over 50cc and up to 125cc', 'Motorcycle over 500cc] these are all the different kinds of two wheeler motorcycles. Hence, all the data can be grouped into a single category as ‘Motorcycle’.

ii. Data Encoding

Encoding categorical data is a process of converting categorical data into integer format so that the data with converted categorical values can be provided to the different models. In the field of data science, before going for modeling, data preparation is a mandatory task.

Label Encoding is a popular encoding technique for handling categorical variables. In this technique, each label is assigned a unique integer based on alphabetical ordering.

The figure 3.3 describes how the label encoding performed on the accident data.

Road Surace	Weather Conditions	Casualty Class
0-Dry	0-Fine without high winds	0-Driver
1-Frost / Ice-	1-Fog or mist – if hazard	1-Passenger
2-Flood	2-Fine with high winds	2-Pedestrian
3-Snow	3-Other	
4-Wet / Damp	4-Raining without high winds	
	5-Raining with high winds	
	6-Snowing without high winds	
	7-Snowing with high winds	
	8-Unknown	

Fig 3.4 : Label Encoding

iii. Drop Unwanted Columns

Some of the attributes are usually stored for reference purposes. Those kinds of attributes are not used for the prediction. That data might reduce the accuracy of the classifier. Hence, those columns are removed from the attribute list.

3.1.3 Build machine learning model

Four machine learning classifiers such as Decision tree learning, K-Nearest neighbor, Random forest classifier and the Gradient boosting classifiers are used to develop machine learning models. These models are imported using the Sci-kit learn machine learning library.

3.1.4 Training and Testing

The pre-processed data splitted into two sets. 70% of the data considered as the training data and the rest 30% used to test the model.

3.1.5 Result Analysis

The machine learning models are evaluated using the standard metrics such as accuracy, precision, recall and ROC curve. Based on the metrics of each classifier, the comparative study was conducted over the models. The result shows that the ensembler can give the best accuracy among the traditional single machine learning classifiers.

3.2 Graphical User Interface

To deploy the model in the web browser the user interface was designed using HTML, CSS and Javascript. Flask framework used to provide the interface between the Mysql database and the web browser.

3.2.1 Save the machine learning model

To save the model, we simply need to pass the model object to Pickle's dump() function. This will serialize the object and convert it into a "byte stream" that can be

stored in the model.pkl file. From the result analysis the Gradient Boosting Classifier provides better accuracy.

3. 3 Tools Used

The tools used for the project:

- Python
- Jupyter Notebook
- Flask
- HTML
- CSS
- JavaScript
- MySql
- Spyder

3. 3. 1 Python

Python is a high-level, interpreted, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation.

Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library.

Guido van Rossum began working on Python in the late 1980s as a successor to the ABC programming language and first released it in 1991 as Python 0.9.0.[35] Python 2.0 was released in 2000 and introduced new features such as list comprehensions, cycle-detecting garbage collection, reference counting, and Unicode support. Python 3.0, released in 2008, was a major revision that is not completely

backward-compatible with earlier versions.

i. Pandas

pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three-clause BSD license.

ii. Scikit Learn

Scikit-learn is a free machine learning library for Python. It features various algorithms like support vector machine, random forests, and k-neighbors, and it also supports Python numerical and scientific libraries like NumPy and SciPy.

iii. Collections

Collections in Python are containers that are used to store collections of data, for example, list, dict, set, tuple etc. These are built-in collections. The collections module provides alternatives to built-in container data types such as list, tuple and dict. Several modules have been developed that provide additional data structures to store collections of data

iv. Matplotlib

It is an amazing visualization library in Python for 2D plots of arrays. It is a multiplatform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack. One of the greatest benefits of visualization is that it allows us visual access to huge amounts of data in easily digestible visuals. Matplotlib consists of several plots like line, bar, scatter, histogram etc.

v. NumPy

NumPy is a general-purpose array-processing package. It provides a high performance multidimensional array object, and tools for working with these arrays.

It is the fundamental package for scientific computing with Python.

3.3.2 Jupyter Notebook

The Jupyter Notebook is an open-source web application for creating and sharing documents with live code, equations, visualizations, and text. Jupyter Notebook is administered by the Project Jupyter team. The Jupyter Notebooks project is a spin-off of the IPython project, which previously had its own IPython Notebook project. Jupyter derives its name from the three primary programming languages it supports: Julia, Python, and R. Jupyter ships with the IPython kernel, which enables you to write your programmes in Python, but you can also use more than 100 other kernels. Jupyter notebooks are intended to provide a more user-friendly interface for code used in digitally-supported research or education.. Jupyter, an open-source environment compatible with a variety of programming languages, has gained traction in a variety of fields. It is useful for documenting code, teaching programming languages, and providing students with a space to easily experiment with provided examples. Jupyter Notebooks can be executed in two major environments: Jupyter Notebook and the more recent JupyterLab. Jupyter Notebook is widely used and well-documented; it provides a simple file browser and an environment for creating, editing, and executing notebooks. Jupyter Lab is more complex and resembles an Integrated Development Environment in its user interface.

The Jupyter Notebook is not only useful for teaching and learning programming languages like Python, but also for sharing data.

Google and Microsoft each offer their own version of the Notebook, which can be used to create and share documents at Google Colaboratory and Microsoft Azure Notebooks, respectively.

JupyterLab incorporates the Jupyter Notebook into a browser-based Integrated Development type Editor. JupyterLab can be viewed as an advanced version of Jupyter Notebook. In addition to Note, JupyterLab allows you to run terminals, text editors, and code consoles in your browser.

3.3.3 Flask

Flask is an API of Python that allows us to build up web-applications. It was developed by Armin Ronacher. Flask's framework is more explicit than Django's framework and is also easier to learn because it has less base code to implement a simple web Application. A Web-Application Framework or Web Framework is the collection of modules and libraries that helps the developer to write applications without writing the low-level codes such as protocols, thread management, etc. Flask is based on the WSGI(Web Server Gateway Interface) toolkit and Jinja2 template engine.

i. HTTP Methods

GET - This is used to send the data in and without encryption of the form to the server.

POST - Sends the form data to the server. Data received by POST method is not cached by the server.

ii. Routing

Nowadays, the web frameworks provide routing techniques so that users can remember the URLs. It is useful to access the web page directly without navigating from the Home page. It is done through the following *route()* decorator, to bind the URL to a function.

iii. Handling Static Files

A web application often requires a static file such as javascript or a CSS file to render the display of the web page in the browser. Usually, the web server is configured to set them, but during development, these files are served as static folders in your package or next to the module.

3.3.4 HTML

HTML is the standard markup language for documents intended for display in a web browser. It can be aided by Cascading Style Sheets (CSS) and scripting languages such as JavaScript. Web browsers receive HTML files from a web server or local storage and convert them into multimedia web pages. HTML describes the semantic structure of a web page and originally included hints for the document's appearance.

HTML elements are the fundamental constituents of HTML pages. Images and other objects, such as interactive forms, can be embedded in the rendered page using HTML constructs. HTML enables the creation of structured documents by assigning structural semantics to text elements such as headings, paragraphs, lists, links, and other elements.

3.3.5 CSS

CSS is a style sheet language that is used to describe the presentation of a document written in a markup language such as HTML. In addition to HTML and JavaScript, CSS is a fundamental technology for the World Wide Web. CSS is designed to separate presentation from content, including layout, colors, and fonts. This separation can improve content accessibility; provide greater flexibility and control in the specification of presentation characteristics; enable multiple web pages to share formatting by specifying the relevant CSS in a separate.css file, which reduces complexity and repetition in the structural content; and enable the.css file to be cached to improve page load speed for pages that share the file and its formatting.

3.3.6 JavaScript

JavaScript, also known as JS, is a programming language that, along with HTML and CSS, is one of the core technologies of the World Wide Web. Over 97 percent of websites use JavaScript for client-side web page behavior, with third-party libraries frequently incorporated. All major web browsers include a dedicated JavaScript

engine for executing code on user devices. JavaScript is a high-level, often just-in-time, ECMAScript-compliant, compiled programming language. It features dynamic typing, prototype-based object orientation, and functions with first-class status. Event-driven, functional, and imperative programming styles are supported. APIs for working with text, dates, regular expressions, standard data structures, and the Document Object Model are available (DOM). Initially, JavaScript engines were only used in web browsers, but they are now integral components of many servers and applications.

3.3.7 MySQL

MySQL is an open-source relational database management system (RDBMS). Its name is a combination of "My", the name of co-founder Michael Widenius's daughter and "SQL", the abbreviation for Structured Query Language. A relational database organizes data into one or more data tables in which data may be related to each other; these relations help structure the data.

SQL is a language programmers use to create, modify and extract data from the relational database, as well as control user access to the database. In addition to relational databases and SQL, an RDBMS like MySQL works with an operating system to implement a relational database in a computer's storage system, manages users, allows for network access and facilitates testing database integrity and creation of backups.

3.3.8 Spyder

Spyder is an open-source cross-platform integrated development environment (IDE) for scientific programming in the Python language. Spyder integrates with a number of prominent packages in the scientific Python stack, including NumPy, SciPy, Matplotlib, pandas, IPython, SymPy and Cython, as well as other open-source software.

3. 4 Algorithms

3.4.1 Decision Tree Learning

Decision Tree algorithm belongs to the family of supervised learning algorithms. Unlike other supervised learning algorithms, the decision tree algorithm can be used for solving regression and classification problems too. The goal of using a Decision Tree is to create a training model that can be used to predict the class or value of the target variable by learning simple decision rules inferred from prior data(training data).

In Decision Trees, for predicting a class label for a record we start from the root of the tree. We compare the values of the root attribute with the record's attribute. On the basis of comparison, we follow the branch corresponding to that value and jump to the next node.

Terminologies Of The Decision Trees

Root Node: It represents the entire population or sample and this further gets divided into two or more homogeneous sets.

Splitting: It is a process of dividing a node into two or more sub-nodes.

Decision Node: When a sub-node splits into further sub-nodes, then it is called the decision node.

Leaf / Terminal Node: Nodes that do not split are called Leaf or Terminal nodes.

Pruning: When we remove sub-nodes of a decision node, this process is called pruning. You can say the opposite process of splitting.

Branch / Sub-Tree: A subsection of the entire tree is called branch or sub-tree.

Parent and Child Node: A node, which is divided into sub-nodes is called a parent node of sub-nodes whereas sub-nodes are the child of a parent node.

The figure 3.5 describes the general structure of the decision tree.

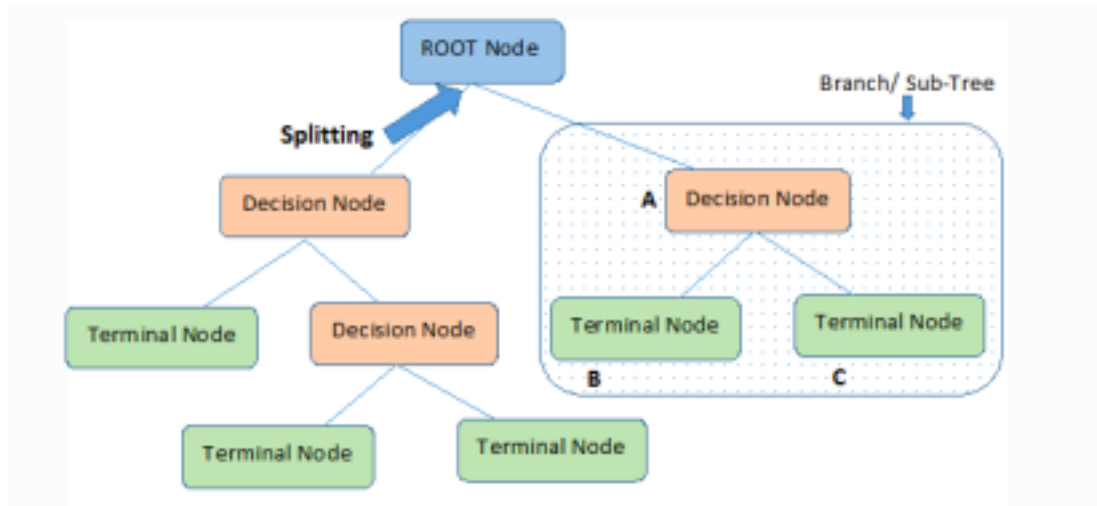


Fig 3.5 : Decision Tree

Steps :

- It begins with the original set S as the root node.
- On each iteration of the algorithm, it iterates through the very unused attribute of the set S and calculates Entropy(H) and Information gain(IG) of this attribute.
- It then selects the attribute which has the smallest Entropy or Largest Information gain.
- The set S is then split by the selected attribute to produce a subset of the data.
- The algorithm continues to recur on each subset, considering only attributes never selected before.

Entropy

Entropy is a measure of the randomness in the information being processed. The higher the entropy, the harder it is to draw any conclusions from that information. Flipping a coin is an example of an action that provides information that is random.

$$E(S) = \sum_{i=1}^c -p_i \log_2 p_i \quad \dots(1)$$

Where, P_i = Probability of randomly selecting an example in class I

Information Gain

Information gain or IG is a statistical property that measures how well a given attribute separates the training examples according to their target classification. Constructing a decision tree is all about finding an attribute that returns the highest information gain and the smallest entropy.

$$\text{Information Gain} = 1 - \text{Entropy} \dots (2)$$

3.4.2 K-Nearest Neighbor

K-Nearest Neighbor is one of the simplest Machine Learning algorithms that utilizes the Supervised Learning technique. It assumes the similarity between the new case/data and existing cases and places the new case in the category that is the most similar to the existing categories. The K-NN algorithm stores all available data and classifies a new data point on the basis of similarity. This implies that when new data becomes available, the K-NN algorithm can easily classify it into a suitable category. It can be used for both Regression and Classification, although Classification is its primary application.

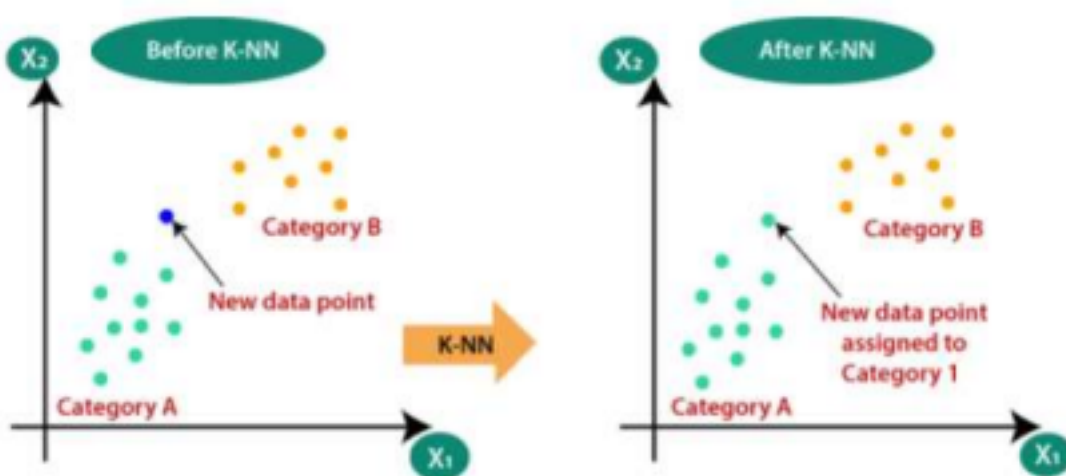


Fig 3.6 : K-nearest Neighbor

K-NN is a non-parametric algorithm, meaning it makes no assumptions about the

data it is analysing. It is also referred to as a lazy learner algorithm because it does not immediately learn from the training set. Rather, it stores the dataset and, at the time of classification, performs an action on the dataset. During the training phase, the system simply stores the dataset, and when it receives new data, it classifies it into a category that is highly similar to the original category.

3.4.3 Random Forest Classifier

Popular machine learning algorithm that belongs to the supervised learning technique is Random Forest. It is applicable to both Classification and Regression problems in Machine Learning. It is based on ensemble learning, which is the process of combining multiple classifiers to solve a complex problem and improve the model's performance.

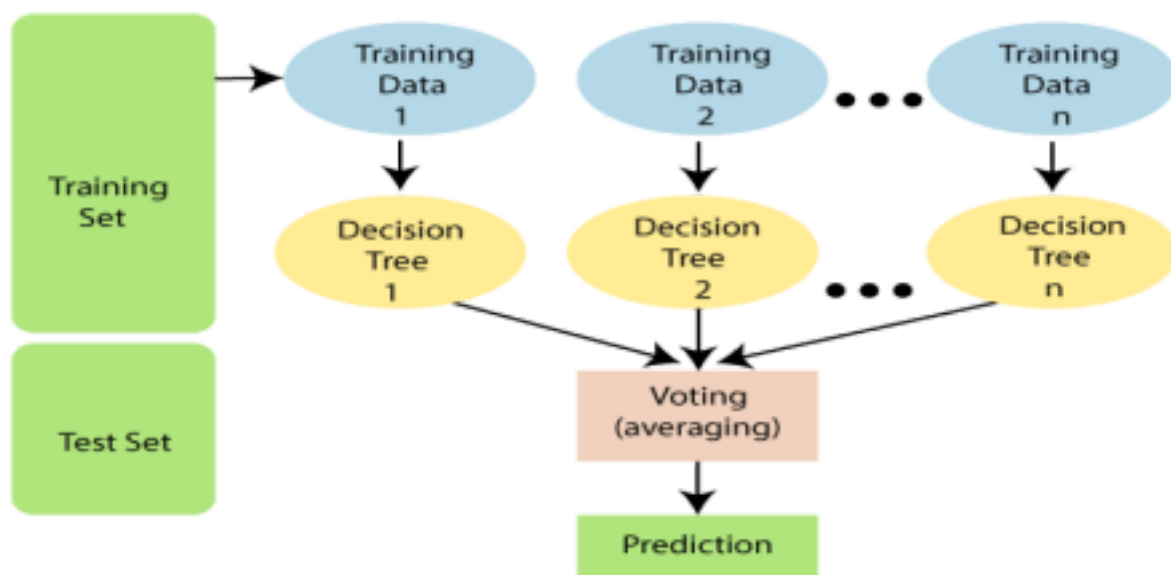


Fig 3.7 : Random Forest Classifier

Since the random forest combines multiple trees to predict the class of the dataset, it is possible that some decision trees may predict the correct output, while others may not. But together, all the trees predict the correct output. Therefore, below are two assumptions for a better Random forest classifier:

There should be some actual values in the feature variable of the dataset so that the classifier can predict accurate results rather than a guessed result.

The predictions from each tree must have very low correlations.

Steps :

Random Forest works in two-phase first is to create the random forest by combining N decision tree, and second is to make predictions for each tree created in the first phase. The Working process can be explained in the below steps and diagram:

Step 1: Select random K data points from the training set.

Step 2: Build the decision trees associated with the selected data points (Subsets).

Step 3: Choose the number N for decision trees that you want to build.

Step 4: Repeat Step 1 & 2.

Step 5: For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

3.4.4 Gradient Boosting Classifier

Gradient boosting algorithm is one of the most powerful algorithms in the field of machine learning. As we know that the errors in machine learning algorithms are broadly classified into two categories i.e. Bias Error and Variance Error. As gradient boosting is one of the boosting algorithms it is used to minimize bias error of the model.

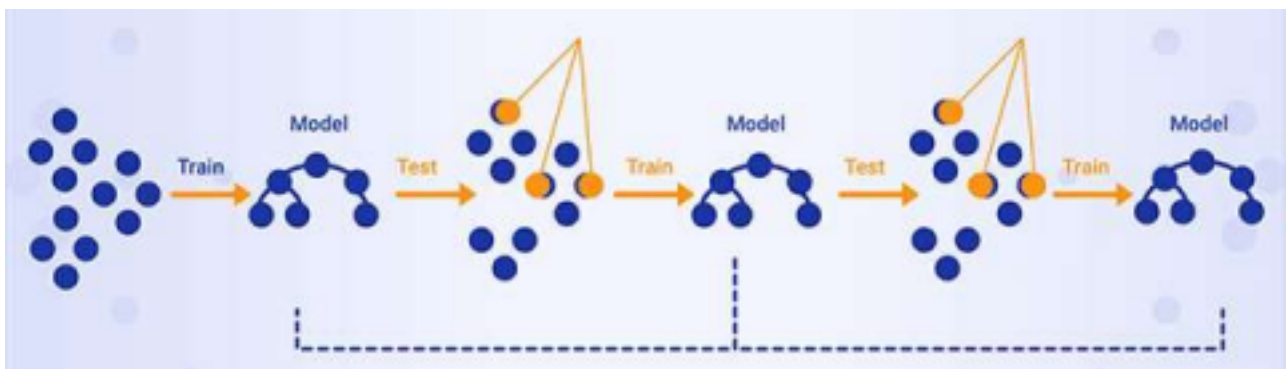


Fig 3.8 : Gradient Boosting Classifier

Gradient boosting algorithms can be used for predicting not only continuous target variables (as a Regressor) but also categorical target variables (as a Classifier). When

it is used as a regressor, the cost function is Mean Square Error (MSE) and when it is used as a classifier then the cost function is Log loss.

Steps:

Step 1: Calculate the average of the target label

Step 2: Calculate the residuals : *residual = actual value – predicted value*

Step 3: Construct a decision tree

Step 4: Predict the target label using all of the trees within the ensemble

Step 5: Compute the new residuals

Step 6: Repeat steps 3 to 5 until the number of iterations matches the number specified by the hyperparameter (i.e. number of estimators)

Step 7: Once trained, use all of the trees in the ensemble to make a final prediction as to the value of the target variable

Chapter 4

RESULTS AND DISCUSSIONS

4.1 Data Preprocessing Result

Figure 4.1 describes about the dataset structure before preprocessing

```
accidentData.head(5)
```

	Reference Number	Easting	Northing	Number of Vehicles	Accident Date	Time (24hr)	1st Road Class	Road Surface	Lighting Conditions	Weather Conditions	Casualty Class	Casualty Severity	Sex of Casualty	Age of Casualty	Type of Vehicle
0	3309	429093	436258	1	01-Jan-09	55	Unclassified	Dry	Darkness: street lights present and lit	Fine without high winds	Pedestrian	Slight	Male	44	Car
1	2609	434723	435534	1	02-Jan-09	2335	Unclassified	Dry	Darkness: street lights present and lit	Fine without high winds	Driver	Serious	Female	23	Car
2	2809	441173	433047	1	02-Jan-09	1645	Unclassified	Dry	Darkness: street lights present and lit	Fine without high winds	Pedestrian	Slight	Female	12	Car
3	3809	428487	431364	1	02-Jan-09	1723	A	Dry	Darkness: street lights present and lit	Fine without high winds	Pedestrian	Slight	Male	15	Car
4	3909	425928	435480	2	02-Jan-09	1350	Unclassified	Dry	Daylight: street lights present	Fine without high winds	Driver	Slight	Female	34	Car

Fig 4.1 : Dataset before preprocessing

Figure 4.2 describes the dataset after performing the pre-processing such as data reduction, label encoding and dropping unwanted attributes.

```
accidentData.head(5)
```

	Number of Vehicles	1st Road Class	Road Surface	Lighting Conditions	Weather Conditions	Casualty Class	Casualty Severity	Sex of Casualty	Age of Casualty	Type of Vehicle
0	1	4	0	2	1	2	0	1	44	6
1	1	4	0	2	1	0	1	0	23	6
2	1	4	0	2	1	2	0	0	12	6
3	1	0	0	2	1	2	0	1	15	6
4	2	4	0	4	1	0	0	0	34	6

Fig 4.2 : Dataset after preprocessing

4.2 Performance Metrics

The following metrics are used to evaluate the performance of the classification.

4.2.1 Confusion Matrix

A confusion matrix, also known as an error matrix, is a summarized table used to assess the performance of a classification model. The number of correct and incorrect predictions are summarized with count values and broken down by each class. The confusion matrix is as shown in figure 4.3.

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Fig 4.3 Confusion matrix

Positive (P): Observation is positive

Negative (N): Observation is not positive.

True Positive (TP): Outcome where the model correctly predicts the positive class.

True Negative (TN): Outcome where the model correctly predicts the negative class.

False Positive (FP): Also called a type 1 error, an outcome where the model incorrectly predicts the positive class when it is actually negative.

False Negative (FN): Also called a type 2 error, an outcome where the model

incorrectly predicts the negative class when it is actually positive.

4.2.2 Accuracy

Accuracy is defined as the ratio of correctly predicted examples by the total examples. Accuracy of the classification is calculated using Eq. (3)

$$Accuracy = TP + TN / (TP + TN + FP + FN) \dots(3)$$

4.2.3 Error Rate : $1 - Accuracy \dots(4)$

4.2.4 Precision

Precision is also called Positive predictive value. Precision is also known as positive predictive value and is the proportion of relevant instances among the retrieved instances. The ratio of correct positive predictions to the total predicted positives. Precision of the classification is calculated using Eq. (5)

$$Precision = TP / (TP + FP) \dots(5)$$

4.2.5 Recall

Recall also called Sensitivity, Probability of Detection, True Positive Rate. The ratio of correct positive predictions to the total positive examples. Recall of the classification is calculated using Eq. (6)

$$Recall = TP / (TP + FN) \dots(6)$$

4.2.6 Specificity

Specificity is defined as the proportion of actual negatives, which got predicted as the negative. Specificity of the classification is calculated using Eq. (7)

$$Specificity = TN / (TN + FP) \dots(7)$$

4.2.7 ROC Curve

A ROC curve (receiver operating characteristic curve) graph shows the performance of a classification model at all classification thresholds. It plots two Parameters namely True Positive (TPR) rate and False positive rate (FPR).

TPR(Eq. 8) and FPR(Eq. 9) of the classification is calculated as follows

$$\text{True Positive Rate} = TP / (FP + TN) \dots (8)$$

$$\text{False positive Rate} = FP / (FP + TN) \dots (9)$$

4.2.8 AUC

AUC stands for Area under the ROC Curve. A perfect classifier would have an AUC of 1. Usually, if your model behaves well, you obtain a good classifier by selecting the value of the threshold that gives TPR close to 1 while keeping FPR near 0.

4.2.9 Performance of the classifiers

The figure 4.4 describes the performance of the Decision Tree classifier

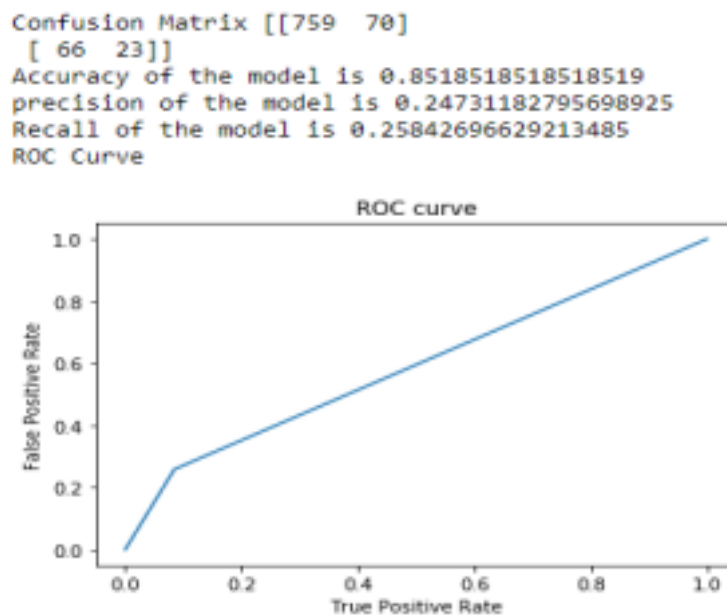


Fig 4.4 Performance of the DT

The figure 4.5 describes the performance of the KNN classifier

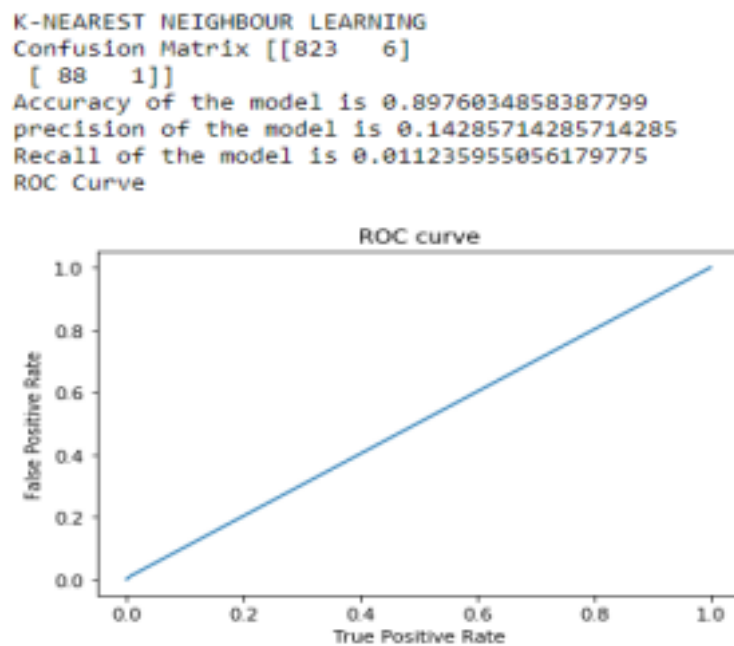


Fig 4.5 Performance of the KNN

The figure 4.6 describes the performance of the Random Forest classifier

```
Random Forest Classifier
Confusion Matrix [[792  37]
 [ 74  15]]
Accuracy of the model is 0.8790849673202614
precision of the model is 0.28846153846153844
Recall of the model is 0.16853932584269662
ROC Curve
```

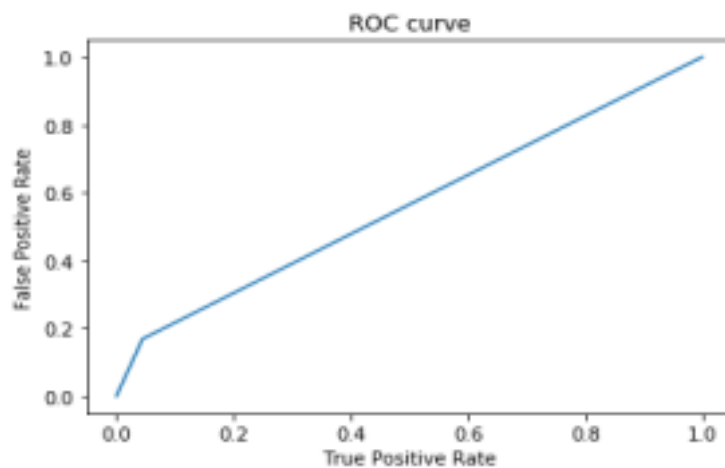


Fig 4.6 Performance of the RF

The figure 4.7 describes the performance of the Gradient Boosting classifier

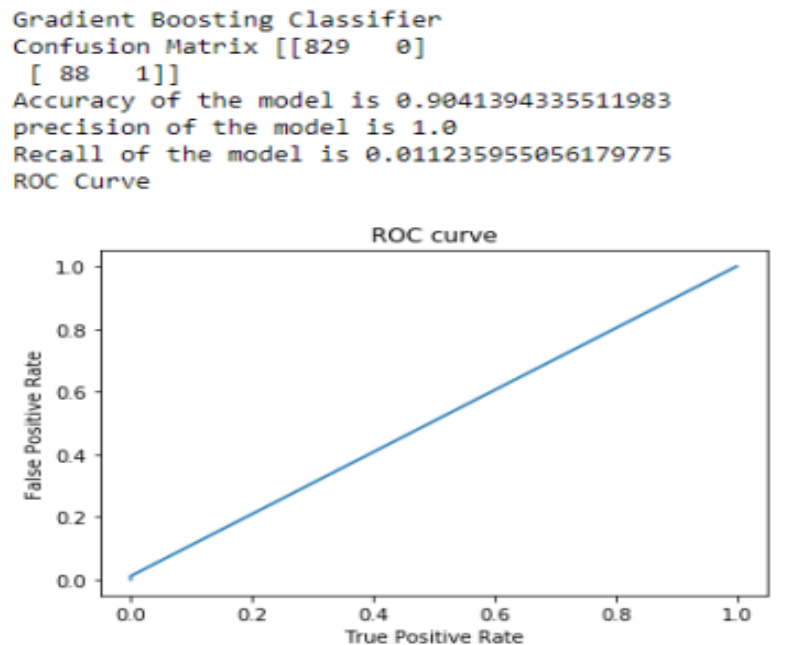


Fig 4.7 Performance of the GB

4.2.10 Comparative Analysis

Each classifier is given best accuracy more than 85%. However the results show that gradient boosting classifiers are given 90% accuracy when comparing all the results of the considered classifiers.

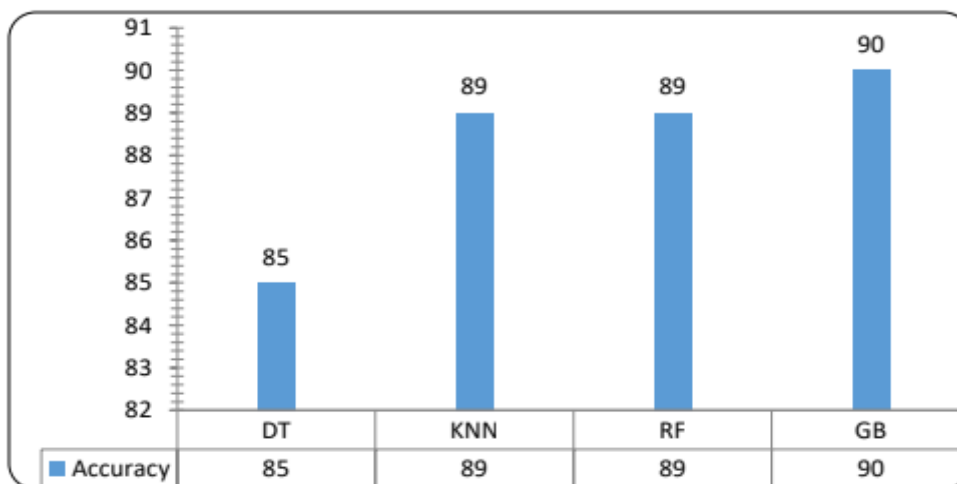
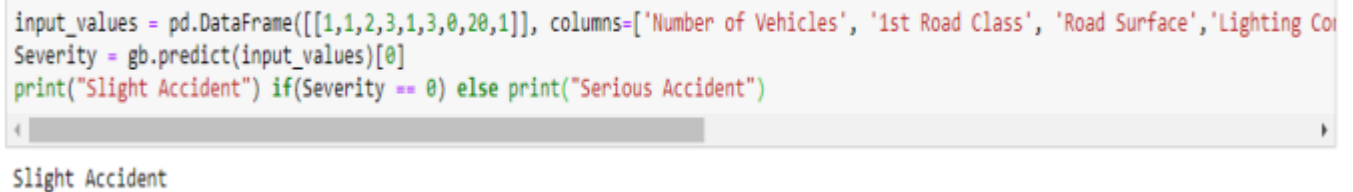


Fig 4.8 Accuracy score

4.2.11 Severity Prediction

The gradient boosting model tested with the different feature values as shown in the figure 4.9.



```
input_values = pd.DataFrame([[1,1,2,3,1,3,0,20,1]], columns=['Number of Vehicles', '1st Road Class', 'Road Surface', 'Lighting Con
Severity = gb.predict(input_values)[0]
print("Slight Accident") if(Severity == 0) else print("Serious Accident")
```

Slight Accident

Fig 4.9 Severity Prediction

4.3 Graphical User Interface

The main objective of developing a user interface is to deploy the extracted machine learning model in a website. By this implementation any end users without any technical knowledge about machine learning also can use the model to predict the accident severity.

The application consists of three phases.

- Front End
- Programming Interface
- Database Server

i. Front End

The front end of this project is developed using HTML, CSS and JavaScript. **ii.**

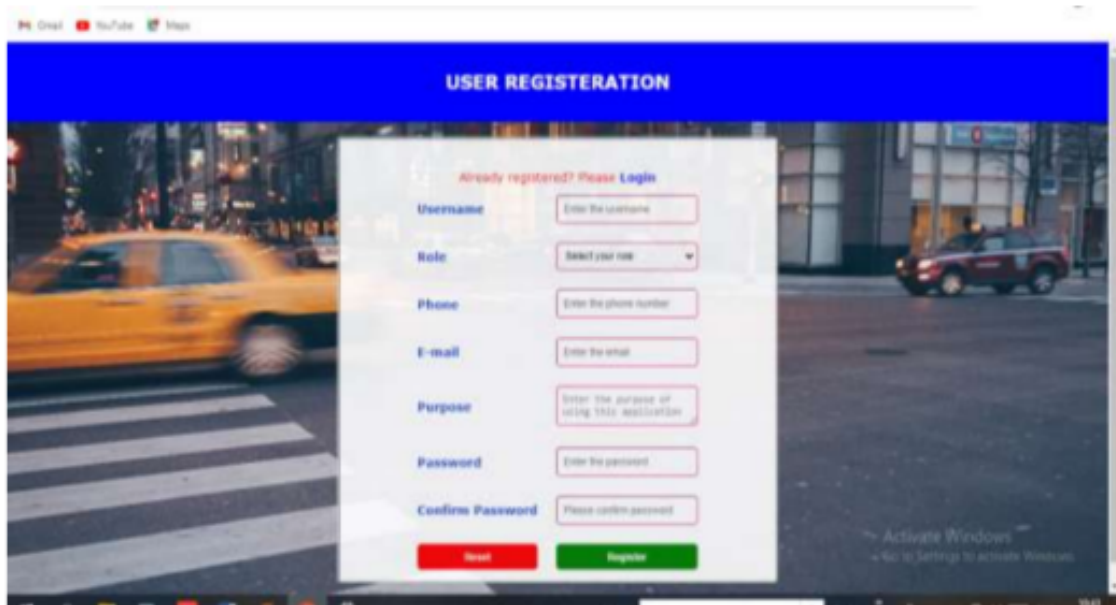
Programming Interface

Flask is used as a backend . It is used to provide the interface between the web browser and the MySql database.

iii. Database Server

Xampp server is used to provide a server based mechanism for the MySQL Database.

The below figures describe the steps to be carried out in the web portal.

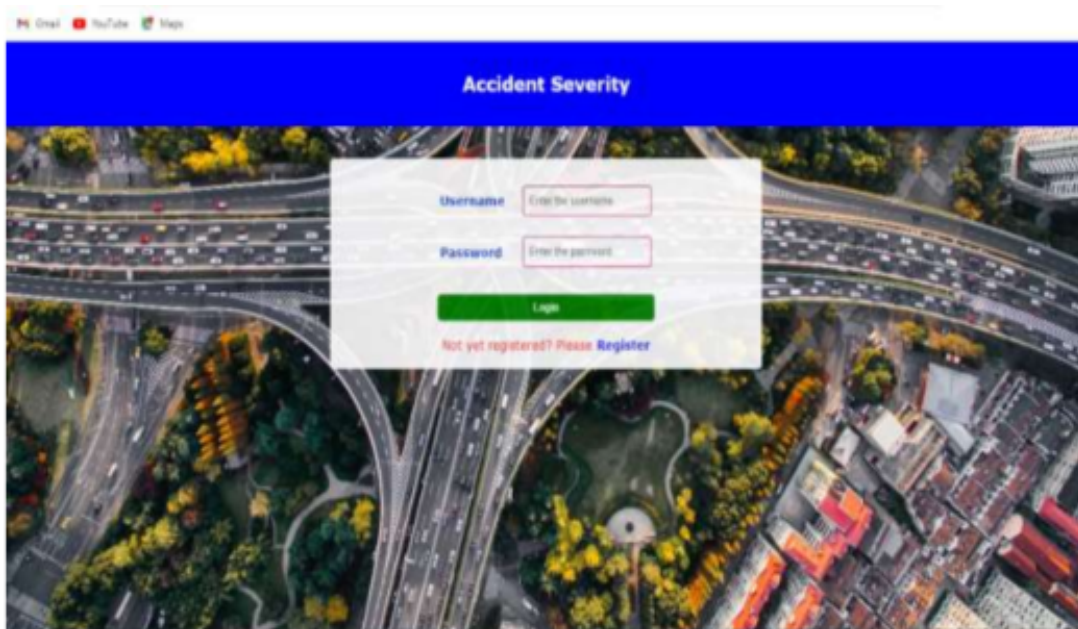


The screenshot shows a web browser window with a blue header bar containing the text "USER REGISTRATION". Below the header is a background image of a city street with a yellow taxi. A white registration form is centered on the page. The form includes the following fields and options:

- Username:** A text input field with the placeholder "Enter the username".
- Role:** A dropdown menu with the placeholder "Select your role".
- Phone:** A text input field with the placeholder "Enter the phone number".
- E-mail:** A text input field with the placeholder "Enter the email".
- Purpose:** A text input field with the placeholder "Enter the purpose of using this application".
- Password:** A text input field with the placeholder "Enter the password".
- Confirm Password:** A text input field with the placeholder "Please confirm password".

At the bottom of the form are two buttons: a red "Reset" button and a green "Register" button. Above the form, there is a link that says "Already registered? Please Login".

Fig 4.10 User Registration



The screenshot shows a web browser window with a blue header bar containing the text "Accident Severity". Below the header is a background image of a busy highway interchange. A white login form is centered on the page. The form includes the following fields and options:

- Username:** A text input field with the placeholder "Enter the username".
- Password:** A text input field with the placeholder "Enter the password".

At the bottom of the form is a green "Login" button. Below the button is a link that says "Not yet registered? Please Register".

Fig 4.11 User Login

Road Accident Analysis

Enter the below input fields to find the accident severities!

Number of Vehicle:

Road Type:

Road Surface:

Light Condition:

Weather:

Casualty Class:

Casualty Sex:

Casualty Age:

Vehicle Type:

Place:

Fig 4.12 User Input

Road Accident Analysis

Result Here: **Slight Accident**

Print: 1 page

Destination:

Pages:

Layout:

More settings:

Fig 4.13 Download the report

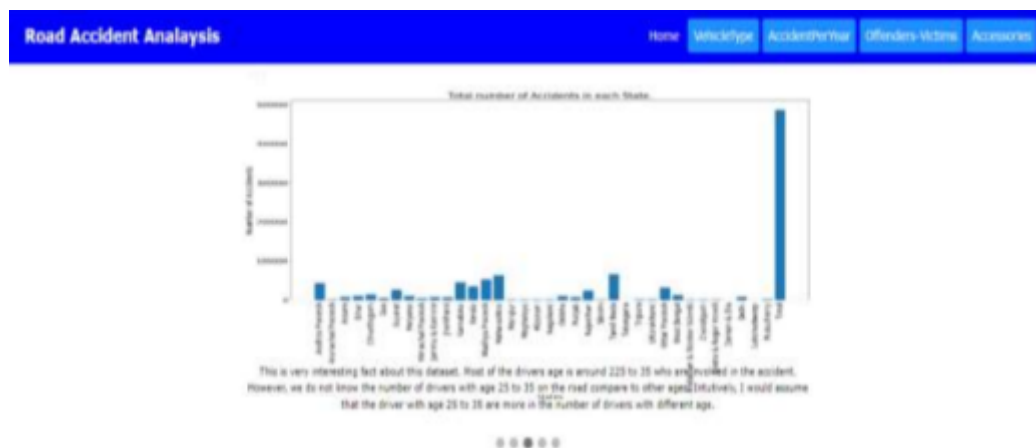


Fig 4.14 Accident factors

Road Accident Analysis						Home	Charts Details
Reference Number	Severity	Investigator	Place	Accident Date	Send Email		
1	Sight	testuser	Tiruchy	2022-05-13 21:38:20	Share via Email		
2	Sight	prithviraj	Tiruchappalli	2022-07-02 09:44:02	Share via Email		
3	Sight	prithviraj	Chennai	2022-07-02 09:45:14	Share via Email		
4	Serious	prithviraj	23	2022-07-02 09:45:42	Share via Email		
5	Sight	prithviraj	Sulthan	2022-07-02 09:51:19	Share via Email		
6	Sight	prithviraj	Namakkal	2022-07-02 09:53:00	Share via Email		
7	Sight	prithviraj	Chennai	2022-07-03 10:45:01	Share via Email		
8	Sight	prithviraj	Chennai	2022-07-03 10:46:11	Share via Email		

Fig 4.15 Previous predictions

Format
Insert
Draw
Options
Discard
Send

B
I
U
Heading 1
Undo
Redo

From: prithivirajk2503@gmail.com

To: chennaipolice@gmail.com;

Cc: cc@example.com;

Bcc: |

Accident Severity Report , Reference Number 8

Accident Reference Number: 8
Place: Chennai
Severity: Slight Accident

Number Of Vehicle(s): 5
Road Type: A - Type
Road Surface: Flood
Lighting Condition: No street lighting
Weather Condition: Street lights present and lit
Casualty Class: Driver
Casualty Age: 23
Casualty Sex: Male
vehicle Type: Motorcycle

With Regards,
prithiviraj

Activate Windows
Go to Settings to activate Windows.

Fig 4.16 Send report via mail

Chapter 5

CONCLUSION

Road accidents are one of the most regrettable hazards in this hectic world. Road accidents lead to numerous casualties, injuries, and fatalities each year, as well as significant economic losses. Predicting the accident severity is one of the major tasks. The proposed model could achieve an accuracy of 90 percent. Number of Vehicles, Road Class, Road Surface, Lighting Conditions, Spot Weather Conditions, Casualty Class, Sex of Casualty, Age of Casualty, Type of Vehicle are used to predict the accident severity. This is extraordinarily beneficial for the highway authorities, police departments and for journalists.

5.1 Advantages

The key advantages of this model is:

- Early accident severity prediction
- No expert knowledge required
- Can be access the model anytime and anywhere
- Can be access the previous predictions
- Can send mail immediately to the respective authority.

Chapter 6

FUTURE SCOPE

The methodology can be used for pre prediction also. Whenever the driver, traveler or passenger starts a journey in a particular area they can predict the accidents happening in that area and the severity of the accident. This can be further used to identify the risk factors, countermeasures. The previous predicted data also can be used to predict the future accident severities and to improve the efficiency of the model.

REFERENCES

- [1] Mubariz mansoor, Muhammad umar , Saima sadiq , Abid isaq, Saleem ullah, Hamza, and Carmen, "RFCNN: Traffic Accident Severity Prediction Based on Decision Level Fusion of Machine and Deep Learning Model", Digital Object Identifier 10.1109/ACCESS.2021.3112546.
- [2] Sachin Kumar and Durga Toshniwal, "A data mining framework to analyze road accident data", DOI 10.1186/s40537-015-0035-y
- [3] Shakil Ahmed, Md Akbar Hossain, Md Mafijul Islam Bhuiyan, Sayan Kumar Ray, "A Comparative Study of Machine Learning Algorithms to Predict Road Accident Severity", 978-1-6654-6667-7/21/\$31.00 ©2021 IEEE DOI 10.1109/IUCC-CIT-DSCI-SmartCNS55181.2021.00069
- [4] Accident Data Analysis to Develop Target Groups For Countermeasures, Max Cameron
- [5] Mohamed K Nour, Atif Naseer, Basem Alkazemi, Muhammad, "Road Traffic Accidents Injury Data Analytics", (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 11, No. 12, 2020
- [6] A. Mehdizadeh, M. Cai, Q. Hu, M. A. A. Yazdi, N. Mohabbati-Kalejahi, A. Vinel, S. E. Rigdon, K. C. Davis, and F. M. Megahed, "A review of data analytic based applications in road traffic safety. Part 1: Descriptive and predictive modeling," Sensors (Switzerland), vol. 20, no. 4, pp. 1–24, 2020.
- [7] Q. Hu, M. Cai, N. Mohabbati-Kalejahi, A. Mehdizadeh, M. A. A. Yazdi, A. Vinel, S. E. Rigdon, K. C. Davis, and F. M. Megahed, "A review of data analytic applications in road traffic safety. Part 2: Prescriptive modeling," Sensors (Switzerland), vol. 20, no. 4, pp. 1–19, 2020.
- [8] J. Ma, Y. Ding, J. C. Cheng, Y. Tan, V. J. Gan, and J. Zhang, "Analyzing the Leading Causes of Traffic Fatalities Using XGBoost and Grid-Based Analysis: A City Management Perspective," IEEE Access, vol. 7, pp. 148 059–148 072, 2019.

- [9] N. Zagorodnikh, A. Novikov, and A. Yastrebkov, "Algorithm and software for identifying accident-prone road sections," *Transp. Res. Procedia*, vol. 36, pp. 817–825, 2018. [Online]. Available: <https://doi.org/10.1016/j.trpro.2018.12.074>.
- [10] L. G. Cuenca, E. Puertas, N. Aliane, and J. F. Andres, "Traffic Accidents Classification and Injury Severity Prediction," in *2018 3rd IEEE Int. Conf. Intell. Transp. Eng. ICITE 2018*, 2018, pp. 52–57