**ROAD TRAFFIC ACCIDENT ANALYSIS USING**

**MACHINE LEARING**

**Major Project Report**

***Submitted in partial fulfillment of the Requirements for the award of degree Of***

**Bachelor of Technology (B. Tech)**

**In**

**COMPUTER SCIENCE AND ENGINEERING**

**By**

**E. TEJASRI (18AG1A0515)**

**P. SHIVASINDHU (18AG1A0546)**

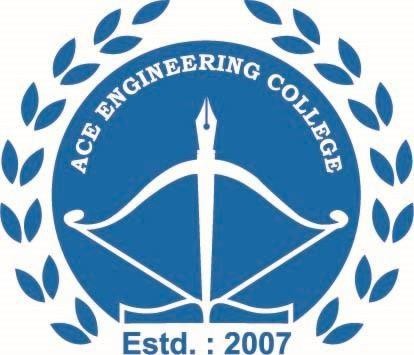
**R. TARUN (18AG1A0548)**

**G. SHIRISHA (18AG1A0516)**

**P. SHIVA SAI GANESH (18AG1A0547)**

**Under the Esteemed Guidance of Mr.V.Maheshwar Reddy**

**Assistant Professor**



**Department of Computer Science and Engineering**

**ACE ENGINEERING COLLEGE**

**An Autonomous Institution**

(NBA ACCREDITED B.TECH COURSES, Accorded NAAC ‘A’ GRADE)

**(Affiliated to Jawaharlal Nehru Technological University, Hyderabad,Telangana)**

**Ghatkesar,Hyderabad-501301**

**JUNE 2022**

** ACE**

**Engineering college**

**An Autonomous Institution**

(NBA ACCREDITED B.TECH COURSES:EEE, ECE, MECH, CIVIL&CSE, ACCORDED NAAC‘A’GRADE)

Ghatkesar, Hyderabad-501301

**(Affiliated to Jawaharlal Nehru Technological University Hyderabad)**

**Website**[**:www.aceec.ac.in**](http://www.aceec.ac.in/) **E-mail**[**:info@aceec.ac.in**](mailto:info@aceec.ac.in)

**CERTIFICATE**

This is to certify that the Major project work entitled “**ROAD TRAFFIC ACCIDENT ANALYSIS USING MACHINE LEARNING TECHNIQUES**” is being submitted by **E.TEJASRI(18AG1A0515),P.SHIVASINDHU(18AG1A0546),R.TARUN(18AG1A0548)**,**P.SIRISHA(18AG1A0516), P.SHIVA SAI GANESH (18AG1A0547)** in partial fulfillment for the award of Degree of **BACHELOR OF TECHNOLOGY** in **COMPUTER SCIENCE AND ENGINEERING** to the Jawaharlal Nehru Technological University, Hyderabad during the academic year 2021-22 is a record of Bonafide work carried out by him/her under our guidance and supervision.

The results embodied in this report have not been submitted by the student to any other University or Institution for the award of any degree or diploma.

### **Internal Guide** **Head of the Department**

**Mr. V. Maheshwar Reddy Dr. M. V. VIJAYA SARADHI**

**Assistant Professor Professor and Head**

**Dept. of CSE Dept. of CSE**

# External Examiner

**ACKNOWLEDGEMENT**

I would like to express my gratitude to all the people behind the screen who have helped me transform an idea into real time application.

I would like to express my heart-felt gratitude to my parents without whom I would not have been privileged to achieve and fulfill my dreams.

A special thanks to our secretary, **Prof. Y. V. GOPALA KRISHNA MURTHY**, for having founded such an esteemed institution. I am also greatful to our beloved principal, **Dr. B. L. RAJU** for permitting us to carry out this project.

I profoundly thank **Dr. M.V. VIJAYA SARADHI**, Head of the Department of Computer Science and Engineering, who has been an excellent guide and also a great source of inspiration to my work.

I extremely thank **Mrs**. **SOPPARI KAVITHA** Assistant Professor, Project coordinator, who helped us in all the way in fulfilling of all aspects in completion of our Major-Project.

I am very thankful to my internal guide **Mr. V. MAHESHWAR REDDY**, Assistant Professor of the Department of Computer Science and Engineering who has been an excellent and given continuous support for the Completion of my project work.

The satisfaction and euphoria that accompany the successful completion of the task would be great, but incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crown all the efforts with success. In this context, I would like to thank all the other staff members, both teaching and non-teaching, who have extended their timely help and eased my task.

**E. TEJASRI (18AG1A0515)**

**P. SHIVA SINDHU (18AG1A0546)**

**R. TARUN (18AG1A0548)**

**G. SHIRISHA (18AG1A0516)**

**P. SHIVA SAI GANESH (18AG1A0547)**

## 

## 

## 

## DECLARATION

I hereby declare that the project work entitled “**Road Traffic Accident Analysis using machine learning techniques**” submitted to the **ACE ENGINEERING COLLEGE** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology (B. Tech)** in Computer Science and Engineering is a record of an original work done by us under the guidance of **Mr. V .Maheshwar Reddy, Assistant Professor** and this project work have not been submitted to any other university for the award of any other degree or diploma.

**E. TEJASRI (18AG1A0515)**

**P. SHIVA SINDHU (18AG1A0546)**

**R. TARUN (18AG1A0548)**

**G. SHIRISHA (18AG1A0516)**

**P. SHIVA SAI GANESH (18AG1A0547)**

## 

# ABSTRACT

Traffic congestion and accidents are two of the most pressing issues we face today. Accidents are unavoidable, despite significant developments in automotive design to add safety features. By constructing a prediction model capable of automatic separation of multiple accident scenarios, we can classify the accident data obtained and detect the accident patterns involved in various circumstances. Machine learning algorithms such as recurrent random forest algorithm in Python can be used to calibrate the models. Clusters are formed through analyzing data. These clusters can be useful in predicting the patterns, preventing the accidents and develop safety measures.

Development of infrastructures is one of government's major investments in developed as well as in developing countries, while road safety is extremely important for passengers. Engineers must analyse all of the parameters that play a critical part in guaranteeing passenger safety throughout the building or maintenance phase of a road. The basic goals of accident data analysis are to identify the major elements that cause traffic accidents. The data are to be analyzed from official sources and have various characteristics. It is difficult to extract and transform such relevant data in order to detect and analyze it in order to generate decision trees that provide additional insights into accident patterns. We used Machine Learning concepts to accomplish this. We used decision trees, classification models, and boosting algorithms to solve our problem. According to the results, the random forest classifier outperformed the Decision tree classifier, naive .

# INDEX

## CONTENTS PAGENO

**CHAPTER 1**

1. Introduction 1-2
2. 1.1Problem Definition 3-4

CHAPTER 2

* 1. Literature Survey 5-7
  2. Existing System 7-8

CHAPTER 3

1. [SOFTWA REREQUIREMENT ANALYSIS](#_TOC_250002)
   1. [Functional Requirements 8-8](#_TOC_250001)

**3.1.1 Advantages of Functional Requirements 8-8**

3.2 [Non-Functional Requirements 8-9](#_TOC_250000)

**3.2.1 Advantages of Non-Functional Requirement 9-9**

## CHAPTER 4

1. **System Analysis**

## 4.1 System Requirements 9-9

**4.2 Software Design 10-10**

## 4.2.1 UML diagrams 10-11

**4.2.2 Class diagram 11-13**

## 4.2. 3 Sequence diagram 14-15

## 4.2.4 Activity diagram 15-17

## 4.2.5 Use case diagram 17-18

**CHAPTER 5**

## 1 Proposed System 19-19

## 5.2 Architecture 19-20

**CHAPTER 6**

## 6.1 Data Preprocessing 21-21

**6.1.1 Steps involved in data preprocessing 21-23**

**6 Modeling**

**6.2 Random Forest Classifier 24-24**

**6.2.1 Code 24-30**

**6.3 Decision Tree 31-31**

**6.3.1 Code 32-36**

**6.4 Naive Bayes 37-37**

**6.4.1 Code 38-43**

**CHAPTER 7**

**7.1 Conclusion 44-44**

**7.2 Future scope 44-44**

**CHAPTER 8**

**8.1 Reference 44-44**

# List of Figures

**Fig No. Descriptions Page No.**

|  |  |  |
| --- | --- | --- |
|  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | |  | | |
| Figure1.1 | No. of accidents according to the time | | 2 |
| Figure4.1 | Class Diagram | | 14 |
| Figure4.2 | Sequence Diagram | | 16 |
| Figure4.3 | Activity Diagram | | 18 |
| Figure4.4 | Use Case Diagram | | 19 |
| Figure5.2 | Architecture | | 21 |
| Figure6.1 | Random Forest Classifier Result 1 | | 30 |
| Figure6.2 | Random Forest Classifier Result 2 | | 31 |
| Figure6.3 | Decision tree Result | | 37 |
| Figure6.4 | Naive Bayes Result | | 44 |
|  |  | |  |

**CHAPTER 1**

**1 INTRODUCTION**

The expense of road fatalities and injuries has a significant impact on society. As a result, traffic accidents must be controlled. It's achievable thanks to data categorization and analysis from past accidents. The difficulty with accident data analysis is that the data contains diverse characteristics. We must view nature as it is, otherwise we will be unable to establish the link between all of the qualities. Accident data that is precise and compressive is the foundation of accident analysis. The accuracy, retention, and exploratory data analysis of accident data may all be used to determine its appropriateness.

The majority of studies have concluded that accident severity may considerably aid in the analysis of accident data. This accident investigation is significant since it is utilized to infer trends and forecast future severity. Using the analyzed data, the main goal is to identify a link between the severity of accident injuries and numerous contributing elements such as geometric design characteristics, traffic circumstances, collision type, and so on. We can determine the patterns of accidents that have occurred using these relationships. Accident clusters are the names given to the patterns.

Accident clusters can help forecast crashes and create an accident-free environment. We consider the collision injury data as a classification issue since it comprises output data as categories such as fatal, grievous, capacitating, non-capacitating, no injury, and so on. Parametric statistical models, such as linear classifiers, Poisson and negative binomial classifiers, and quadratic classifiers, have drawbacks, such as the need to make assumptions about data distributions. We use non-parametric approaches and Artificial Intelligence models like Classification and Regression Tree Models, as well as decision trees, to overcome the restriction.

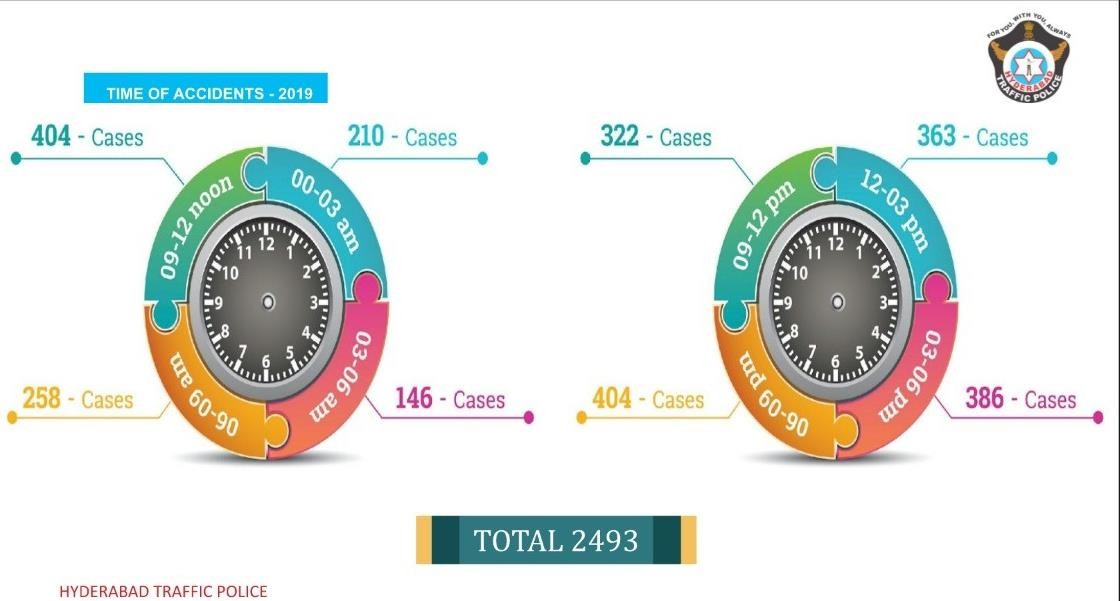


Fig1.2 No. of accidents according to the time

# 1.1 Project Definition

In this project, we focus on the analysis of given data of accidents with factors. The main objective is to identify the relationship between attributes that significantly affecting the severity of accident. This involves in three main steps, those are,

• Data Preprocessing (Data Wrangling, ETL)

. • Exploratory Data Analysis.

• Advanced Data Analysis. (Modeling and Prediction)

This in depth process of analysis can’t be done by typical database or analysis toolkit like Excel. In order to analyze precisely we have used advanced scripting language and its respective working environment, that is, we have used python 3.7 to analyze the data.

Python is an open source scripting language that may be used for both development and statistical research. In the Python documentation, there are a plethora of analytic packages. All of these packages will assist us in making data analysis easier. For the most part, we used the following packages in our project for analysis and modelling.

• Pandas

• Numpy

• Matplotlib, pyplot

• Seaborn (Used to add styles and effects to the plots)

• Statsmodel (Used for modeling and statistical operations)

• Sklearn/Sci-kit learn (Used for all kinds of metrics, advanced analysis and modeling)

All these analysis is done in a python environment, which is Anaconda Jupyter Notebook. Jupyter Notebook is an open source IDE available for three scripting and statistical languages that are Julia, Python, and R. It is an interactive IDE for easy working and understanding of code. The models we have implemented for advanced analysis of given data are,

• Random Forest Classifier

• Decision Tree Classifier

• Naive Bayes

The given data is to be analyzed for severity of accident. Severity type attribute is ordinal data type. Hence, we chose the classifier data mining techniques listed above. The random forest classifier is simply a combination of decision tree classifiers. Decision tree classifier gives the best described severity type for each validation set.

A naive Bayes classifier is an algorithm that uses Bayes' theorem to classify objects. Naive Bayes classifiers assume strong, or naive, independence between attributes of data points. Popular uses of naive Bayes classifiers include spam filters, text analysis and medical diagnosis. These classifiers are widely used for machine learning because they are simple to implement. Naive Bayes is also known as simple Bayes or independence Bayes.

# CHAPTER 2

# 2.1 LITERATURE REVIEWS

**1. Classification of Vehicle collision pattern in road accidents using  Data mining-2011**

**Authors:** S. Shanthi, Dr. R. Geetha Ramani

**Findings:** This paper emphasises the importance of data mining techniques like C-45, C RT, CS-MC4, ID3 decision trees, Random forest classifier, Naive Bayes Classifier. The paper concluded that random forest classifier has better performance when compared to other techniques.

**Drawbacks:** In Naive Bayes, all predictors are assumed to be independent, which is rarely the case in real life.

This method encounters the Zero-Frequency problem, in which it assigns a category variable zero probability.

**2. Road traffic accident simulation modelling- A Kernel approach, Nov 2015**

**Authors:** Dr. Clive Esabel, Dr. Simon Kingham, Alan Nicholson; New Zealand.

**Findings:** This article outlined the development of a method for using Kernel estimation  cluster analysis technique to automatically identify road traffic accident black spots. The study used GIS, Python environments and Kernel estimation with Monte Carlo  simulation technique. The study concluded that accident clusters have identified with Kernel  estimator and Monte Carlo technique has found to be statistically significant. **Drawbacks:** Monte Carlo simulation usually requires several (perhaps many) runs at given  input values. Time consuming to build simulation.

**3. Multivariable Monte Carlo Analysis methods in Traffic accident  reconstruction using Python-2015**

**Authors:** Michael A. Knox, Jasonvill

**Findings**: The research used Monte Carlo multivariable method for analysing multiple, co- dependent and stochastic variables of traffic crash data.

The method provided an excellent tool for calculating the probability ranges of vehicle speed  and multiple stochastic input variables.

**Drawbacks**: It purely depends on our assumptions. If we choose the best set of variables  the algorithm works more efficiently and if our assumption is wrong this algorithm does not  works properly.

**4. Analysing angle crashes at unsignalized intersections using Machine  Learning techniques-2015**

**Authors:** Md. Abdel Aty,Kirolos Haleem

**Findings:** The study used two predominant techniques for research. They are multivariable  adaptive regression splines and negative binomial regression. The study focussed on finding  angle crash frequency at 3 and 4 legged unsignalized intersections and found MARS technique yielded better results than Negative binomial.

**Drawbacks:** Fitting process for MARS regression is done in a stepwise greedy manner  due to this the speed of the algorithm decreases and also takes long time for smaller data sets.

**5.Crash Injury Severity analysis using SVM and OP,Models-2016**

**Authors:** Zhibin Li, Pan Lui, Wei Wang, Chengcheng Xu, Nanjing

**Findings:** The research used Support Vector Machine, Ordered Probit Model. Research found that Support Vector Machine model showed better performance than OP  model. It is important to notice that SVM model shows better performance on small  proportions and it doesn’t work well with multiclassification problems. **Drawbacks:** Key Parameters has to be set correctly.

**6.Traffic accident Data mining using Machine Learning techniques-2016**

**Authors:** Miao Chong, Ajith Abraham,Marcin Paprzycki

**Findings:** The scope of the article is to predict severity of accident injuries. This article used four machine learning paradigms.i.e., neural network, decision trees, support vector machine, Hybrid decision tree neural network.

The article concluded that Hybrid decision tree neural network has better performance of all techniques used in classifying the crash severity data.

**Drawbacks:** This algorithm works mostly on larger data sets. Many other machine learning works more effectively than this algorithm. It also takes more time than any other traditional algorithm for development.

**7.Road accident prediction using Machine Learning paradigms, Mar-2018**

**Authors:** Vipul Rana, Deepak Parmar, Monika Kanojiya

**Findings:** The research objective is to analyse previously occurred accident and to make better predictions based on constraints.

Software like Jupyter, Python, HTML, Jscript; Classification algorithm like logistic regression algorithm was used to classify 70-30 and 80-20 dataset.

The paper concluded that parameters like vehicle speed, road conditions, climatic conditions contributed most in the cause of accidents.

**Drawbacks:** The major limitation is the assumption of linearity between dependent and independent variables. Hence, this cannot be used for solving non-linear problems.

**8. Model Evaluation for Forecasting Traffic Accident Severity Using  Machine Learning Algorithms: Seoul City Study- 2020**

**Authors:** Jonghak Lee,Taekwan yoon,Sangil Kwon,Jongtae Lee

**Findings:** This work used ANN and decision tree algorithms to analyse data from traffic  accidents. In comparison to the other models, the ANN algorithm overestimated the AL. Due  to AL overestimation, the ANN model produced the highest values in the model evaluation  using the MSE and RMSE.

**Drawbacks:** Because this approach cannot handle complex associations without making  assumptions and does not lessen the risk of over-fitting, the data were evaluated using a  machine learning tool.

**9. Traffic Accident Injury and Severity Prediction Using Machine  Learning Algorithms-2020**

**Authors:** Nithin Kashyap,Hari Raksha K. Malali,Koushik S. E,Raju G,T. H. Sreenivas **Findings:** This investigation establishes models for selecting a large number of compelling  variables and developing a model for categorising wound severity. Accident data is used to test  both supervised and unsupervised learning systems. The main goal is to determine if there is a  link between various types of incidents and the types of injuries that may have happened.  Different machine learning algorithms are used to plan these models.

**10.Prediction and Analysis of the Severity and Number of Suburban Accidents Using Logit Model, Factor Analysis and Machine Learning – 2021**

**Authors:** Meisam Ghasedi,Maryam Sarfjoo,Iraj Bargegol

**Findings:** The goal of this study is to look into and figure out what factors influence  automobile and pedestrian accidents in the busiest regions. This strategy may aid in the  development of the most thorough and effective model for identifying the key contributing  component. Excessive speed, rainy weather, and driver age (30–50) all play major roles in the  severity of vehicle accidents, according to the factor analysis and logit model.

**11.Older Pedestrian Traffic Crashes Severity Analysis Based on an  Emerging Machine Learning – 2021**

**Authors:**ManzeGuo,ZhenzhouYuan,BruceJanson,YongxinPengYangYang,WenchengWa

**Findings:** Extreme Gradient Boosting (XGBoost) is utilised in this study to simulate the  classification problem of three different levels of severity of older pedestrian traffic crashes  from Colorado crash data. The interpretation results suggest that the most important elements  impacting the probability of the three severity levels are driver characteristics, older pedestrian  characteristics, and vehicle movement.

# 2.2 EXISTING SYSTEM

# 

# Naive bayes and Decision tree are used in the current system. Naive Bayes assumes that all predictors (or features) are independent, rarely happening in real life.This algorithm faces the 'zero-frequency problem' where it assigns zero probability to a categorical variable whose category in the test data set wasn't available in the training dataset. The first and foremost disadvantage of decision making is that it is too expensive to process. Decision making in organisations involves different peoples for taking proper action. Putting different people together in one requires large efforts.

The present systems are less efficient as a result of the two factors mentioned above. Some of  the existing systems directly implemented the code with dataset without any analysis. But this does  not give us a clear understanding of variables or attributes in the dataset before modeling the code. This investigation establishes models for selecting a large number of compelling variables and  developing a model for categorizing wound severity. Accident data is used to test both  supervised and unsupervised learning systems. The main goal is to determine if there is a link  between various types of incidents and the types of injuries that may have happened.

Different  machine learning algorithms are used to plan these models.

**CHAPTER 3**

# SOFTWARE REQUIREMENT ANALYSIS

**3.1 Functional Requirements**

These are the features that the platform should give in order to meet the expectations of the  end user. As part of the agreement, each of these features must be integrated into the system.  These would be represented or stated as input to the system, function to be carried out, and  expected outcome. While functional user requirements may be high-level statements about what the sy stem  should do, they should also include detailed details about the system's services. You can use functional requirements to express the system's desired behaviour. This activity  might be represented by operations, offerings, or actions that the system must perform.

**3.1.1 Advantages of Functional Requirements**

1. Assists you in determining whether the application has all of the features specified in  the functional requirements.

2. A requirement specification document assists in defining the functionality of a system  or one of its subsystems.

**3.2 Non-Functional Requirements**

A non-functional necessity defines the quality attribute of a software system. They are set of criteria used to assess the specific operation of a system. It's vital to ensure that the entire  software system is functional and useable. Non-functional requirements that aren't met can  result in systems that don't meet the needs of the users.

Non-functional Requirements allow you to constrain or limit the architecture of the system  across many agile backlogs. For example, if there are more than ten thousand concurrent  visitors, the site should load in three seconds. Non-functional requirements must be specified  with the same attention to detail as functional requirements.

**3.2.1 Advantages of Non-Functional Requirements:**

1. They create a positive consumer experience, keep expenses low, and make the product  simple to use.

2. They ensure that the software system is reliable, available, productive, and flexible.

3. They ensure that the software system conforms with all applicable laws and regulations,  as well as defining the software's quality attribute.

4. They contribute to the creation of the software system's defensive.

# CHAPTER 4

**4.1 System Requriements**

# 

**4.1.1 Jupyter Notebook**

Jupyter Notebook App is a web-based server-client utility for editing and running notebook  papers. The Jupyter Notebook App can be used locally on a PC that does not have internet access or remotely on a server that does. It's a fantastic tool for putting together and presenting interactive data science projects.

Jupyter Notebooks is a clone of the IPython project, which used to have its own project called IPython Notebook. Julia, Python, and R are the three main programming languages that Jupyter supports. Jupyter includes the IPython kernel, which allows users to write Python programs, however there are currently over 100 other kernels available.

The notebook was divided into two sections. Users insert computer code or text into cells  on a front-end web page. There have been over 100 Jupyter kernels created, which support  a variety of programming languages. The code is subsequently transferred from the browser  to the backend kernel, which executes the code and returns the results.

The feed in(input) and outputs of a collaborative session are included in the notebook, as  well as explanatory content that is intended to supplement the code but is not run. The  notebook includes useful output from running code, such as HTML, photographs, video,  and charts, making it a complete and self-contained record of a calculation.

**4.1.2 Python**

Python is a powerful and adaptable programming language that can be used to accomplish  a wide range of tasks. It's used in a variety of fields, including web development, data  science, and software prototyping. For newcomers, Python features a basic and easy-to understand syntax. Python is an ideal language for novices to learn to programme in  because of this. Python includes a wide standard library with a variety of useful codes and  functions that we can use in our Python programmes. It features exception handling, which  allows us to write less erroneous code and test a variety of scenarios that could lead to an  error.

Python supports automated memory management, which ensures that memory is cleaned  and removed automatically. It is not necessary for you to clear your memory. Python may  also be used to build charts that represent data analysis and visualization.

## 4.2 SOFTWARE DESIGN

**4.2.1 UML Diagrams**

The Unified Modelling Language (UML) is a tool that aids a designer in communicating  project concepts to a customer and developer. Modeling is crucial in the creation of software.  It's possible that a lousy model will lead to shoddy software development. To aid in the description of systems from diverse perspectives, a UML system incorporates  five different views. Each view consists of a collection of diagrams and components that  display real-time objects.

**a. User Model view**

1. It simulates user behavior in a system setting.

2. All schematics are drawn with the user's response and reaction to a system in  mind.

**b. Structural Model view**

1. For modelling static structures, this view includes a class diagram and an object  diagram.

2. Objects, characteristics, relationships, and operations are all used.

**c. Behavioral Model view**

1. The key components are the sequence diagram, cooperation diagram, status chart  diagram, and activity diagram. They basically reflect the flow of actions between  the system's many components.

2. They're utilized to visualize the system architecture's different dynamic  characteristics.

**d. Implementation Model view**

1. Component diagrams and deployment diagrams are included in this view. This  view is used to represent an organization's static software components. 2. This is where the data files, documentation, executables, and source code are  normally stored.

3. These are the system's physically replaceable components.

**4.2.2 Class Diagram**

**Definition:** A class diagram is a type of static structural diagram that represents the structure

of a system by displaying the system's classes, properties, operations (or methods), and  relationships among objects in the Unified Modeling Language (UML). They are a representation of a static application. Only class diagrams can be directly mapped  with OOP languages because everything in OOPs is modelled in the form of classes and  objects. As a result, these diagrams are commonly used in the building process. This is one of  the most commonly used UML diagrams among designers. A class diagram is essential in  forward and reverse engineering.

1. It serves as a base for the component and deployment diagrams.

2. It mainly describes and defines the basic responsibilities of system’s application.

3. It implements the analysis and design view for a static application.

In a class diagram, each object is modelled as a class.  Each class consists of section or compartments.

1. Class name

2. Attributes of a class

3. Methods or functions

4. Documentation (optional section)

While constructing a class diagram, following points are to be considered: a. In order to represent the framework's characteristic, the class diagram's name must be  significant.

b. Each component, as well as the links between them, must be identified ahead of time. c. Each class has a set of responsibilities (attributes and methods) that need to be declared  properly.

d. The number of properties in each class must be reduced to a bare minimum. Because the  unneeded attributes will lead the diagram to become jumbled.

a. When a component of the diagram needs to be depicted, take notes. Because it is near the  end of the diagram, it must be justified to the designer/coder.

f. Before finalizing the final version, the diagram must be sketched on plain paper and  revised as many times as necessary to make it correct.

**1.Scopes**: The UML diagrams have two different types of scopes for class members: • instance members scope and

• classifier members scope

**2.Classifier members** are “static” members of a class in many programming languages. The  scope is the class itself.

Static attributes are common to all other objects that invoke the class and Static methods are  not instantiated.

**3.Instance members** are nothing but the members that are local toan object.  The main purpose of instance members is to allow the objects to store their states.

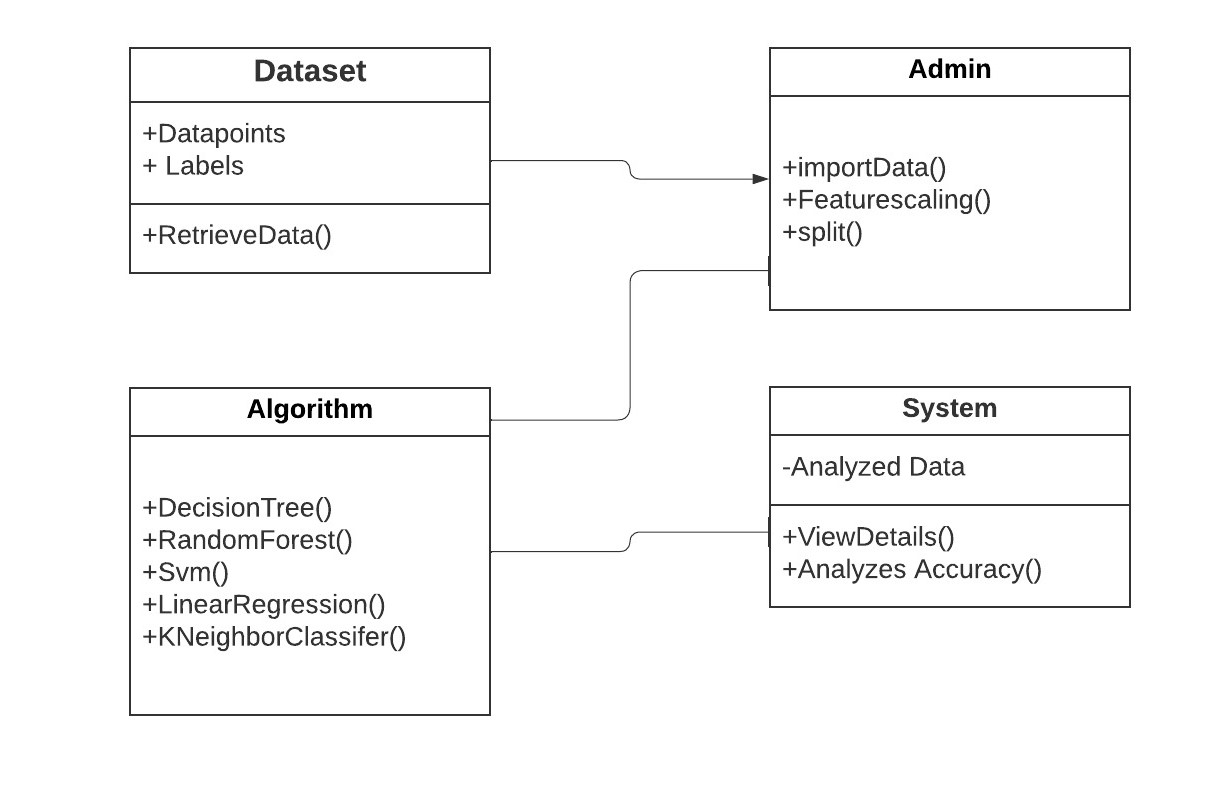


Fig4.1 Class Diagram for Road Traffic Accident Analysis and Prediction Machine Learning

**4.2.3 Sequence Diagram**

**Definition** The Sequence Diagram depicts the interaction of several elements in an  application across time. It denotes the series of messages that objects exchange in order to  fulfil the essential functionality.

Lifelines, which are usually vertical parallel lines, make up the structure. It is made up of  horizontal arrows that point towards the direction of the messages that are sent in a logical  and easy-to-understand manner.

The lifeline of an object represents a role. A synchronous call is represented by a solid arrow  head, while asynchronous contacts are represented by an open arrow head.

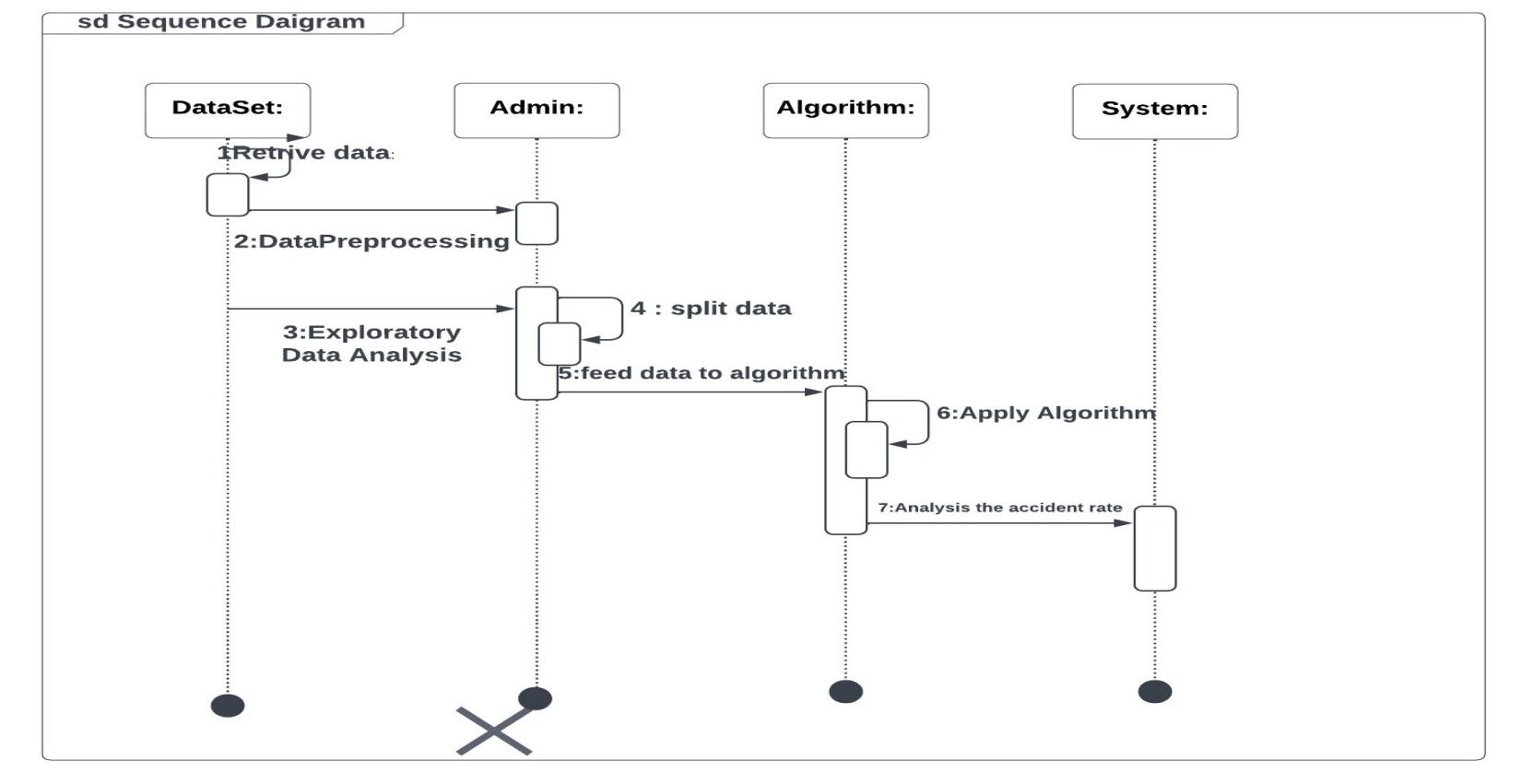
Every object is represented in the order in which it appears in time. The importance of  message timing in sequence diagrams cannot be overstated. An item is killed instantly when  it is utilized in a sequence diagram.

**1. Common properties:** An arrangement graph is a sort of diagram that has some  qualities in common with other diagrams. In any case, in terms of content, it is unlike  any other diagram.

**2. Contents:** Objects can refer to occurrences of other objects like components,  interactions, and nodes, as well as named or unnamed instances of a class. The object  is graphically represented as a rectangle because of its name.

**3. Links:** A link is a semantic linkage between items, i.e., a connection is an object of an  affiliation. A line is used to depict it.

4. **Messages:** A message is a determination of a correspondence between objects that  transmits data in the hopes of triggering an action.

 Fig 4.2 Sequence Diagram for Road Traffic Accident Analysis and Prediction using machine learning

**4.2.4 Activity Diagram:**

**Definition:** The flow of information from one action to the next can be depicted using an  activity diagram, which is a flow chart. It is used to create UML diagrams. It represents all of  the system's objects' dynamic properties.

The control flow from one object to the next is shown, along with the basic operations to be  performed.

The following steps are used to create an activity diagram:

1. Rectangle with rounded corners is used to represent actions.

2. Diamonds are used to represent decisions.

3. Concurrent activities bars are represented using the start or end.

Diagram

Description automatically generated

4. Time event is represented as

Diagram

Description automatically generated

5. Encircled black circle is used to represent final state



The purpose of an activity diagram is the same as that of other UML diagrams. The dynamic behavior of the system is depicted in the activity diagram. Backward and forward engineering mechanisms are used to construct a system.

An activity diagram serves the following purposes:

1) To depict the activity (flow) in a system.

2) The goal is to show how events flow from one activity to the next.

The elements that are used in an Activity diagram are

1. Association relationship

2. Activities

3. Constraints

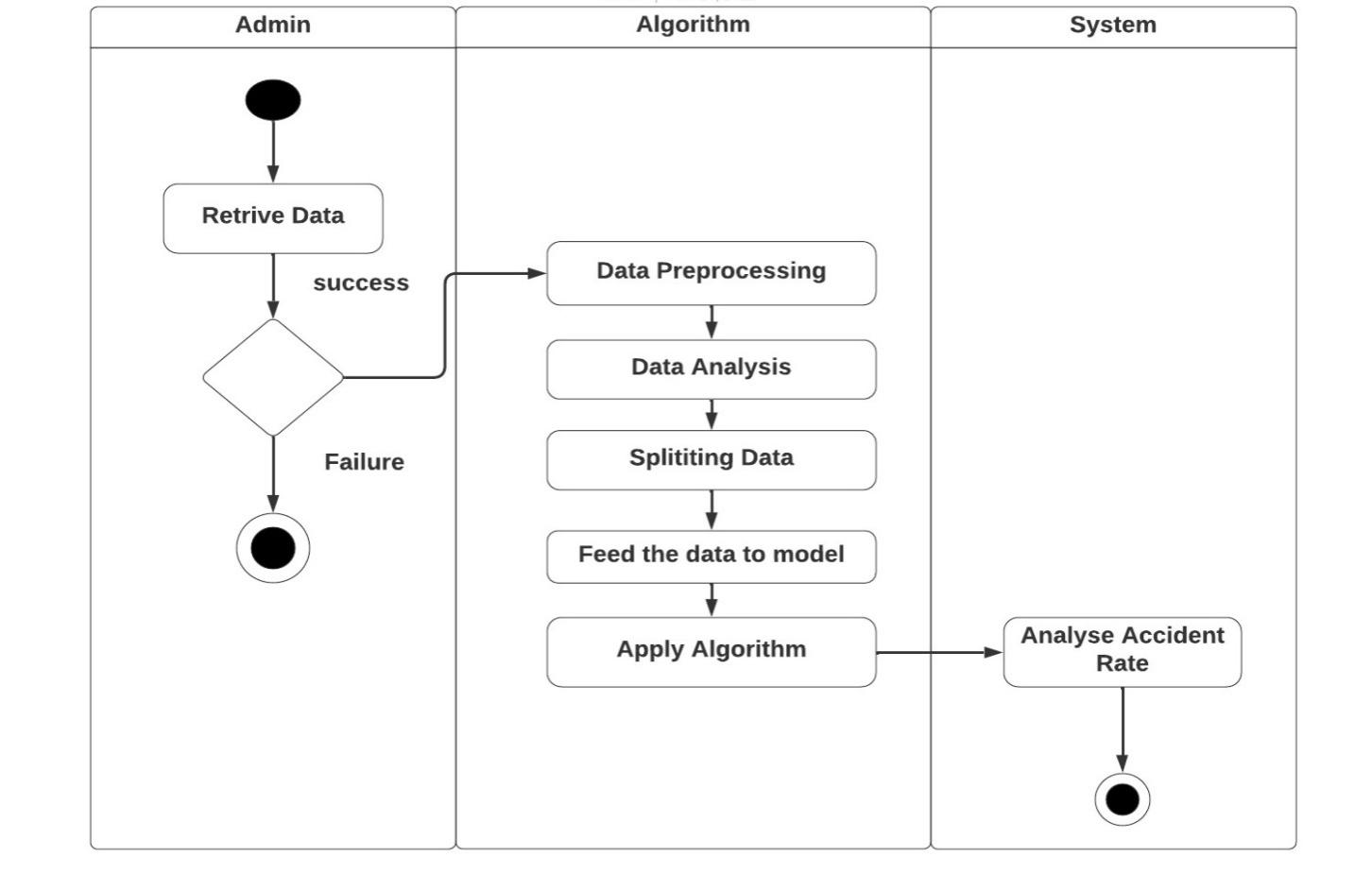


Fig4.3Activity Diagram for Road Traffic Accident Analysis and Prediction using Machine Learning

**4.2.5 Use case Diagram:**

**Definition:** The simplest basic representation of a user's engagement with technology is the use case diagram.

It includes the user's interaction with numerous use cases, as well as the actors involved.  There are numerous links to examine between the use cases and the actors.

The following are some of them:

1. Association relationship

2. Generalization

3. Dependency

4. Realizations

5. Transitions

Components of a use case diagram include the following:

1. Actors

The users that interact with the system.

2. System

A specific sequence of actions and interactions between actors and the system. 3. Goals

The end result of most use cases.

**Use case diagram symbols and notation:**

**Associations:** A line between actors and use cases.

**System boundary boxes:** A box that sets a system scope to use cases.

**Packages:** A UML shape that allows you to put different elements into groups.

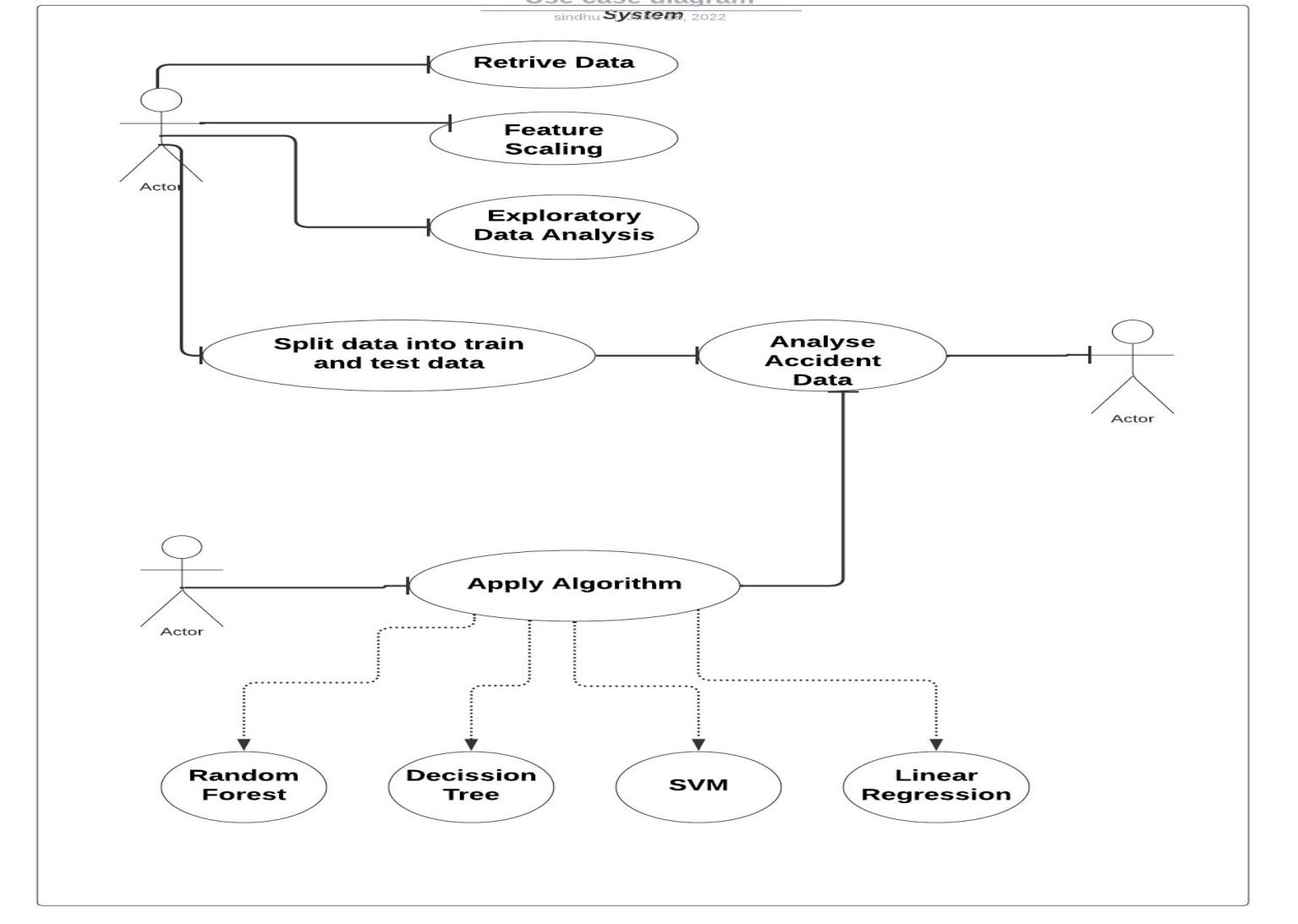


Fig4.4Use case Diagram for Road Traffic Accident Analysis and Prediction using Machine Learning

# CHAPTER 5

**5.1 PROPOSED SYSTEM**

We are utilizing a random forest classifier to alleviate the shortcomings of the current approach.

The system is more efficient than the existing approach because Random Forest is a mixture of several Decision trees.

Each decision tree's gene index and entropy change are used by the Random Forest method. When the entropy change is big and the gene index is low, the system is said to be more  random, or more efficient.

# 5.2 ARCHITECTURE

# Random Forest Classifier, Decision Tree Classifier, Naïve bayes, are the three methods we used. We compare the accuracy and forecasts of such algorithms.

Finally, we identify the algorithm with the best accuracy and ability to accurately forecast the target variable (severity).

Data collection is the first step in the project. Data is such an important part of the project  that there is no way to an analyze it without it.

Preprocessing of Data:

• In Excel: EXCEL is used for basic data adjustments.

• In Python: In-depth adjustments, such as filling null values, deleting duplicates,  defining data types, correcting entering values, and extracting desirable attributes is all  done here. It is simply termed as Extraction, Transformation, Loading.  • The cleaned data is exported to system as ‘.csv’ file.

Clean data is subjected to exploratory data analysis where we draw insights about data.  Furthermore, advance analysis is done on the data.

• The data is split into train and test data.

• Using train data, we train the model we have implemented.

• The fitted model is used to make predictions.

• The accuracy of predicted values is validated by using test data. This is called validation  of model.

**ARCHITECTURE**

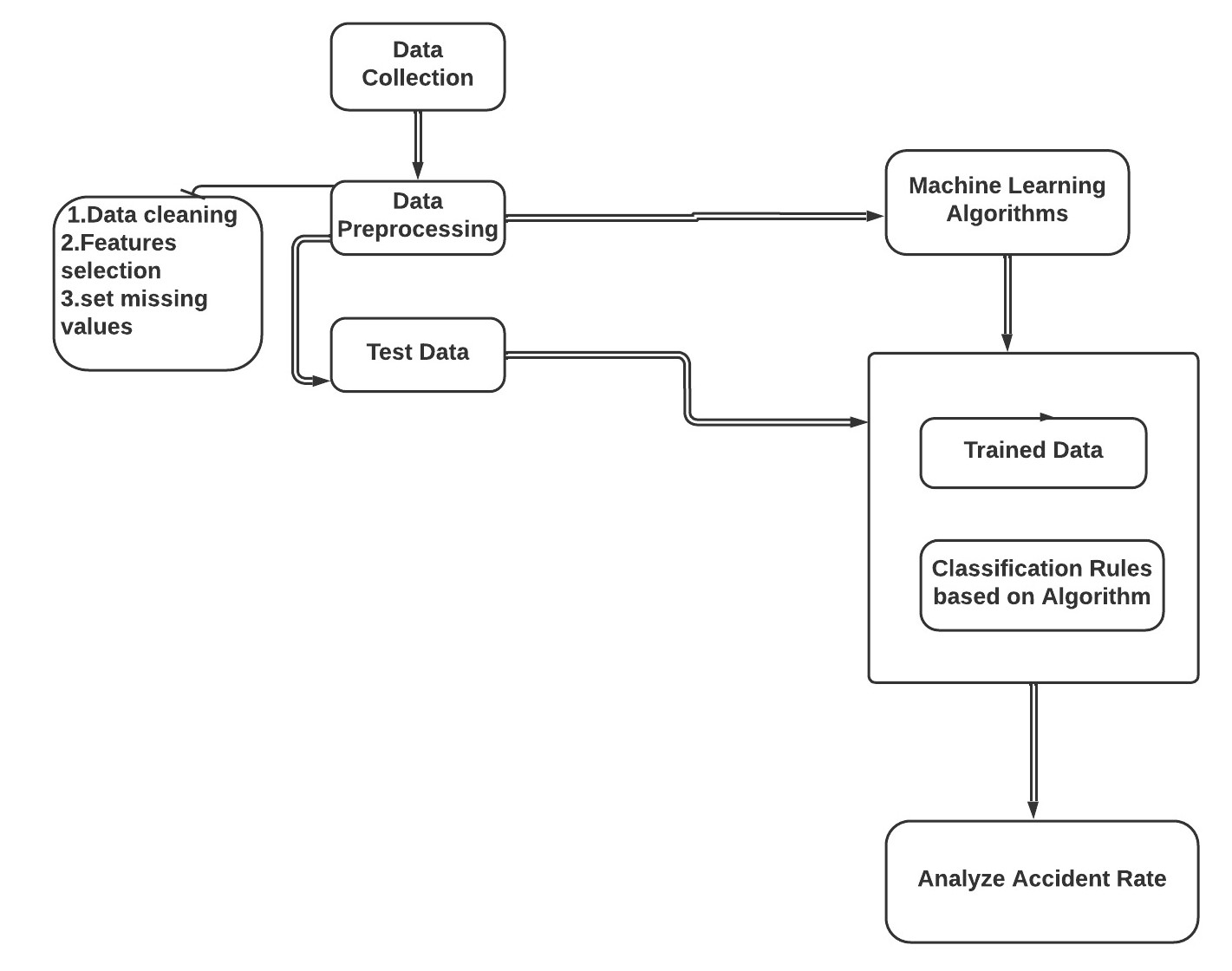


Fig5.1Architecture

# CHAPTER 6

# 6.IMPLEMENTATION

# 6.1 Data Preprocessing

# Data preprocessing is a data mining technique which is used to transform the raw data in a useful and efficient format.

# ****6.1.1 Steps Involved in Data Preprocessing****

**1. Data Cleaning:**   
 The data can have many irrelevant and missing parts. To handle this part, data cleaning is

done. It involves handling of missing data, noisy data etc. 

**(a). Missing Data:**   
 This situation arises when some data is missing in the data. It can be handled in various ways.  
 Some of them are:

**1. Ignore the tuples:**   
 This approach is suitable only when the dataset we have is quite large and multiple values are

missing within a tuple. 

**2.Fill the Missing values:**   
 There are various ways to do this task. You can choose to fill the missing values manually,

by attribute mean or the most probable value. 

**(b). Noisy Data:**   
 Noisy data is a meaningless data that can’t be interpreted by machines. It can be generated

due to faulty data collection, data entry errors etc. It can be handled in following ways:

**1. Binning Method:**   
 This method works on sorted data in order to smooth it. The whole data is divided into segments of equal size and then various methods are performed to complete the task. Each

segmented is handled separately. One can replace all data in a segment by its mean or boundary

values can be used to complete the task.

**2. Regression:**   
 Here data can be made smooth by fitting it to a regression function.The regression used may be linear (having one independent variable) or multiple (having multiple independent variables). 

**3. Clustering:**   
 This approach groups the similar data in a cluster. The outliers may be undetected or it will fall outside the clusters.

**2. Data Transformation:**   
 This step is taken in order to transform the data in appropriate forms suitable for mining

process. This involves following ways:

**1.Normalization:**   
 It is done in order to scale the data values in a specified range (-1.0 to 1.0 or 0.0 to 1.0) 

**2.Attribute Selection:**   
 In this strategy, new attributes are constructed from the given set of attributes to help the

mining process. 

**3.Discretization:**   
 This is done to replace the raw values of numeric attribute by interval levels or conceptual

levels. 

**4.Concept Hierarchy Generation:**   
 Here attributes are converted from lower level to higher level in hierarchy. For Example-The

attribute “city” can be converted to “country”.

**3. Data Reduction:**   
 Since data mining is a technique that is used to handle huge amount of data. While working

with huge volume of data, analysis became harder in such cases. In order to get rid of this, we uses data

reduction technique. It aims to increase the storage efficiency and reduce data storage and analysis

costs.

The various steps to data reduction are:

**1.Data Cube Aggregation:**   
 Aggregation operation is applied to data for the construction of the data cube. 

**2.Attribute Subset Selection:**   
 The highly relevant attributes should be used, rest all can be discarded. For performing

attribute selection, one can use level of significance and p- value of the attribute the attribute having

p- value greater than significance level can be discarded.

**3.Numerosity Reduction:**   
 This enable to store the model of data instead of whole data, for example: Regression Models. 

**4.Dimensionality Reduction:**   
 This reduce the size of data by encoding mechanisms. It can be lossy or lossless. If after

reconstruction from compressed data, original data can be retrieved, such reduction are called lossless

reduction else it is called lossy reduction. The two effective methods of dimensionality reduction

are: Wavelet transforms and PCA (Principal Component Analysis).

# 6.2 Modeling

Modeling involves in,

* + Random Forest Classifier
  + Decision Tree Classifier
  + Naive Bayes

**6.2.1 Random Forest Classifier**

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of **ensemble learning,** which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

As the name suggests, **Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset.** Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

# 6.2.2 CODE

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from sklearn.model\_selection import train\_test\_split

import seaborn as sns

import os

from google.colab import files

uploaded = files.upload()df1=pd.read\_csv('data.csv')

df1.head(100)

A picture containing text, decorated

Description automatically generated

df1.columns

df1.info()

len(df1.columns)

Table

Description automatically generated

Table

Description automatically generated

df1.isnull().sum()

Text

Description automatically generated

df1.isnull().values.any()

->**True**

df1.head()

Graphical user interface

Description automatically generated with low confidence

df=df1.loc[:,df1.columns!='accident\_index']

df2=df.loc[:,df.columns!='local\_authority\_(highway)']

df3=df2.loc[:,df2.columns!='lsoa\_of\_accident\_location']

df4=df3.loc[:,df3.columns!='date']

df5=df4.loc[:,df4.columns!='time']

df6=df5.fillna(0)

Y\_mnl=df6['accident\_severity']#Response

X\_mnl=df6.drop(['accident\_severity'],1) #predictors

X\_train\_mnl ,X\_test\_mnl,y\_train\_mnl,y\_test\_mnl=train\_test\_split(X\_mnl,Y\_mnl,test\_size=0.3,random\_state=0)

Graphical user interface, application

Description automatically generated

from sklearn.ensemble import RandomForestClassifier

model\_RF=RandomForestClassifier(n\_estimators=100)

model\_RF.fit(X\_train\_mnl,y\_train\_mnl)

print(f'Model Accuracy: {model\_RF.score(X\_train\_mnl,y\_train\_mnl)}')

n\_nodes=[]

max\_depth=[]

for ind\_tree in model\_RF.estimators\_:

  n\_nodes.append(ind\_tree.tree\_.node\_count)

  max\_depth.append(ind\_tree.tree\_.max\_depth)

print(f'Average number of nodes {int(np.mean(n\_nodes))}')

print(f'Average maximum depth{int(np.mean(max\_depth))}')

Text

Description automatically generated

y\_test\_mnl1=model\_RF.predict(X\_train\_mnl)

from sklearn import metrics

pred=model\_RF.predict(X\_test\_mnl)

print(pred.shape)

print(y\_test\_mnl1.shape)

**->(85600,)**

**(199732,)**

y\_test\_mnl.shape

**->(85600,)**

print("Accuracy Of The Model: ",metrics.accuracy\_score(y\_test\_mnl,pred))

**->Accuracy Of The Model: 0.940642523364486**

from sklearn import metrics

from sklearn.metrics import confusion\_matrix,ConfusionMatrixDisplay

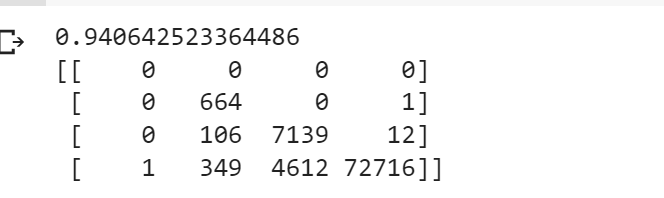
import matplotlib.pyplot as plt

pred=model\_RF.predict(X\_test\_mnl)

print(metrics.accuracy\_score(pred,y\_test\_mnl))

cm=metrics.confusion\_matrix(pred,y\_test\_mnl)

print(cm)



cm=confusion\_matrix(y\_test\_mnl,pred,labels=None)

print(confusion\_matrix)

**-><function confusion\_matrix at 0x7f02886c9950>**

import seaborn

seaborn.heatmap(cm)

plt.show()

**A picture containing chart

Description automatically generated**

Fig 6.1 Random Forest Result

sns.heatmap(df1.corr())

**<matplotlib.axes.\_subplots.AxesSubplot at 0x7f02836f2bd0>**

A picture containing diagram

Description automatically generated

Fig 6.2 Random Forest Result

df1.vehicle\_reference.value\_counts()

**Table

Description automatically generatedTable

Description automatically generated**

df1.pedestrian\_location.value\_counts()

**Table

Description automatically generated**

**6.3 Decision Trees**

A Decision Tree is a decision-making aid that uses a tree-like graph or model of  options and their potential consequences, such as chance, resource, and utility  outcomes. One way to demonstrate an algorithm is to use just conditional control  expressions.

The decision tree is a flow chart structure in which each inner node represents a  test (for example, a coin flip) on an attribute, each branch reflects the test's result,  and each leaf node represents a class label (decision taken after computing all  attributes). The classification rules are represented by the root to leaf routes.

Tree-based learning algorithms are one of the best and most often used supervised  teaching methods. Predictive models with tree-based techniques are more  predictable, stable, and interpretable.

**6.3.1 How Decision Tree Works?**

A tree accuracy is greatly influenced by the decision to make strategic splits. For  classification and regression trees, the criteria for choice are different. Decision trees are dividing a node into two or more sub nodes by using several methods.  The formation of sub-nodes enhances their homogeneity. In other words, with regard to  the goal variable, we may say that the purity of the node rises. The decision tree divides  nodes into all the variables available and picks a split, resulting in the most uniform sub nodes.

# 6.3.1 CODE

Graphical user interface, text, application, email, Teams

Description automatically generated

Table

Description automatically generated

Graphical user interface, table

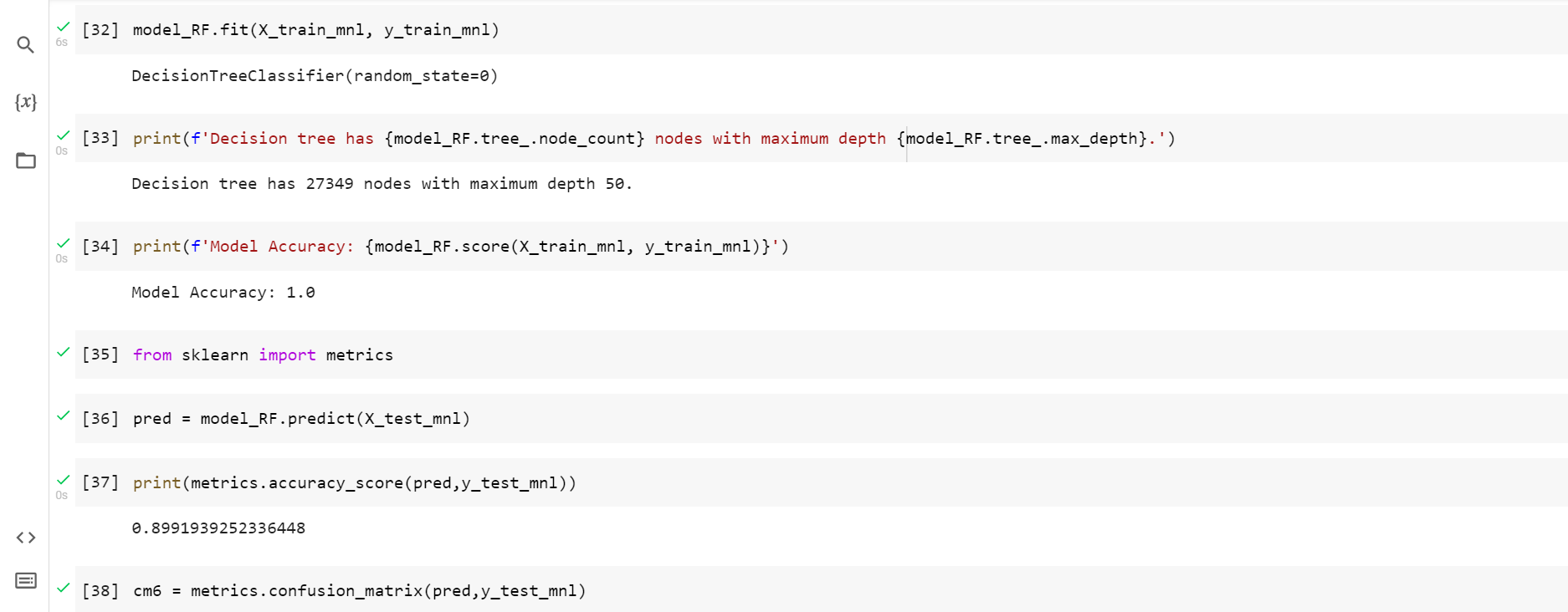
Description automatically generated with medium confidence

Graphical user interface, table

Description automatically generated with medium confidence

Graphical user interface, text, application

Description automatically generated with medium confidence



Table

Description automatically generated

Graphical user interface, application

Description automatically generated

Chart

Description automatically generated

Fig 6.3 Decision Tree Result

**6.4 Naive Bayes**

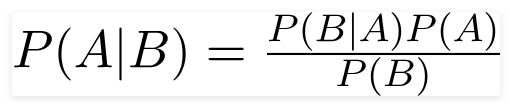
Naive Bayes classifiers are a collection of classification algorithms based on **Bayes’ Theorem**. It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e. every pair of features being classified is independent of each other.

# 6.4.1 Principle of Naive Bayes Classifier

A Naïve Bayes classifier is a probabilistic machine learning model that’s used for classification task.

The crux of the classifier is based on the Bayes Theorem.

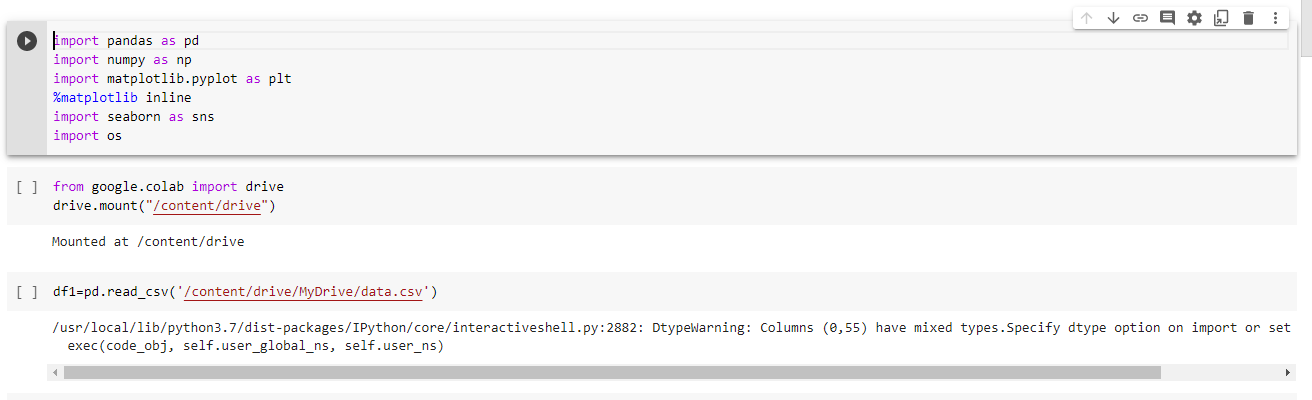
# 6.4.2 Bayes Theorem:

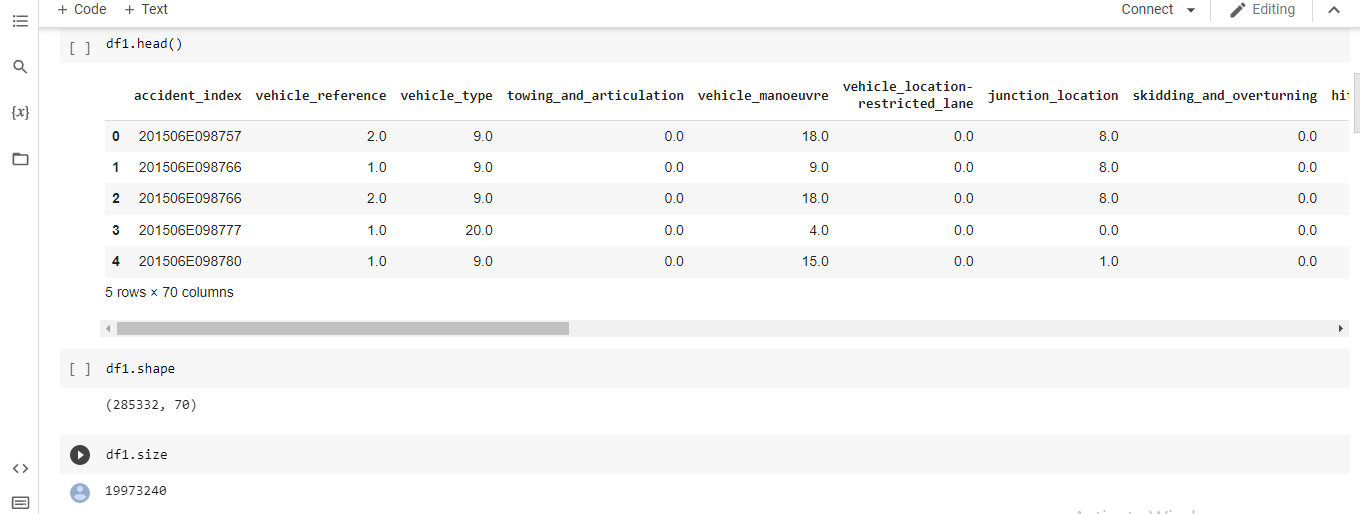


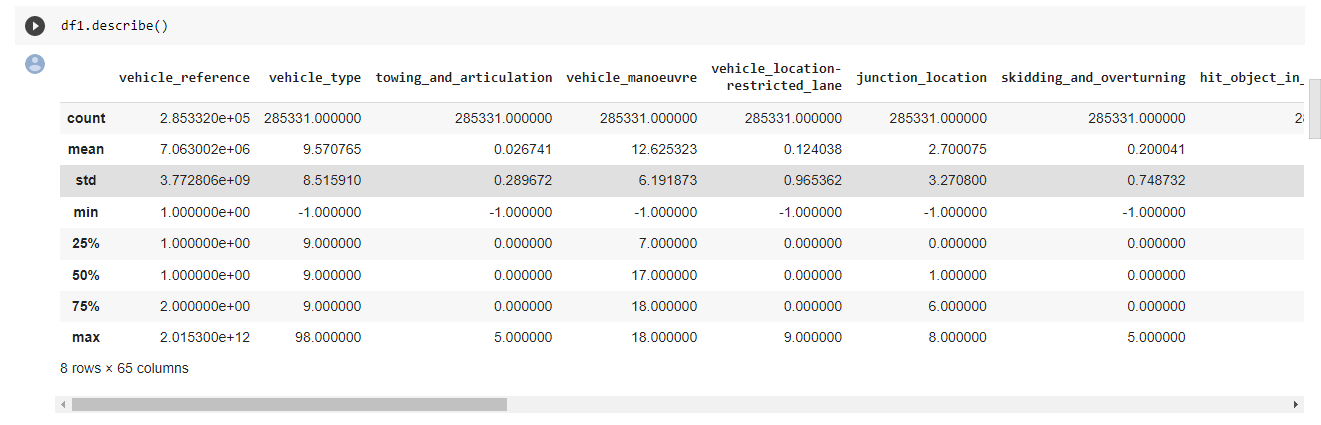
Using Bayes theorem , we can find the probability of A happening. Given that B has occurred. Here , B is the evidence and A is the hypothesis. The assumption made here is that the predicators/features are independent .

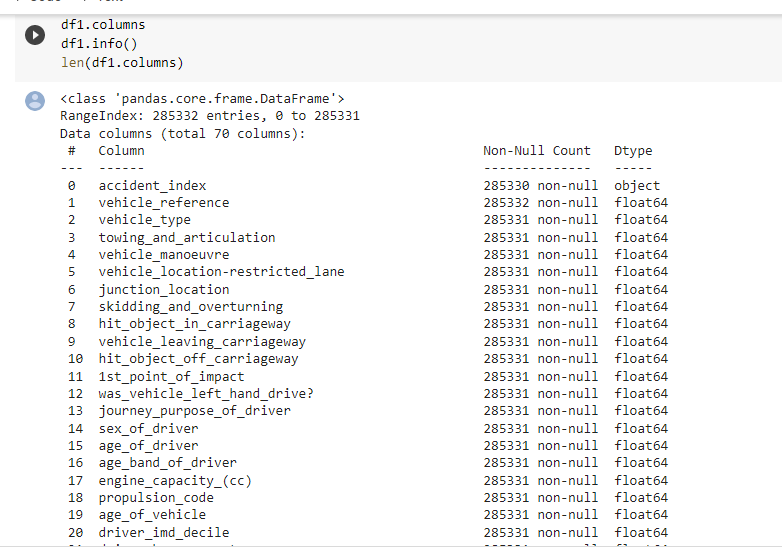
That is presence of one particular feature does not affect the other. Hence it is called naïve.

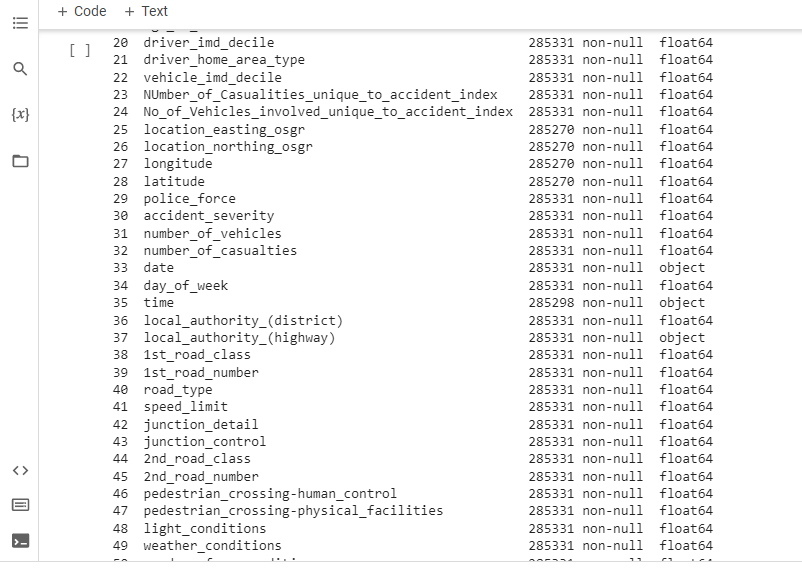
**6.4.3 CODE**

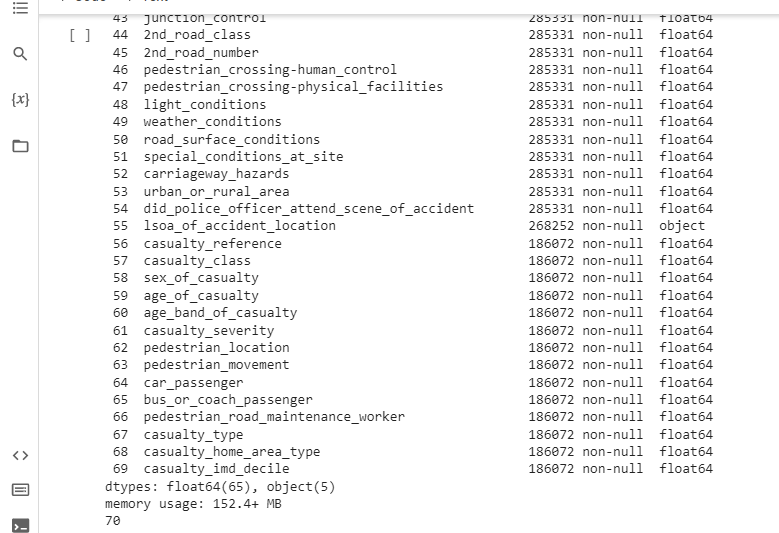


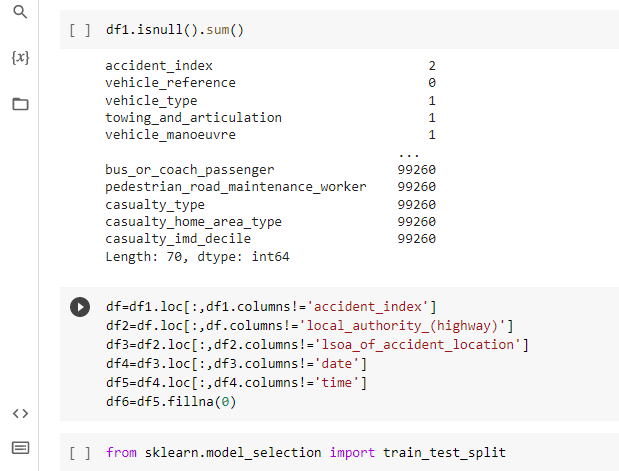


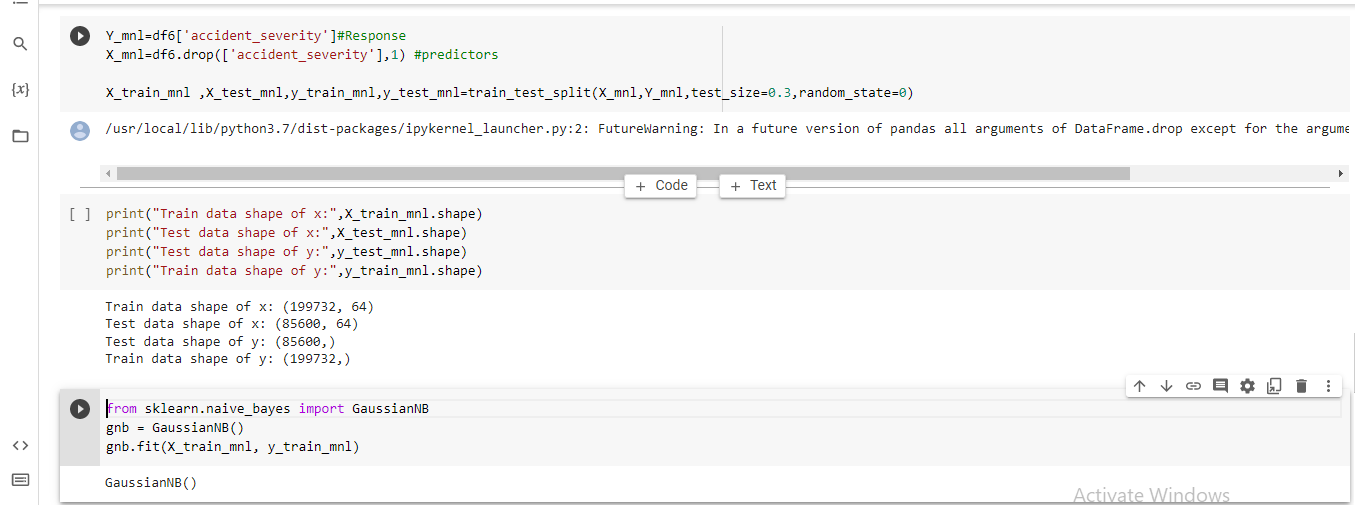














|  |  |
| --- | --- |
|  |  |

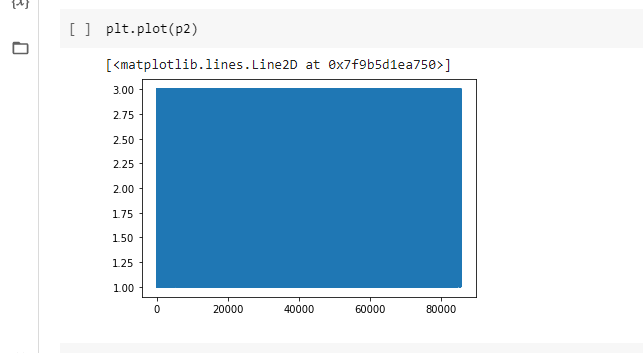


Fig 6.4 Naive Bayes Result

**CHAPTER 7**

**7.1 CONCLUSIPN**

Upon the analysis and tuning, the naive bayes classifier, Decision Tree and Random Forest algorithms, Random Forest have performed with the best accuracy(94.03%)while the other algorithms naïve bayes classifier has performed with the accuracy of 83 % and decision Tree has performed with the accuracy of 90 %.By further tree pruning and gradient boosting it can possibly yield best results of all models for Accident Severity Analysis.

From EDA,

Precautions to be taken for pedestrian safety.

Two wheeler motor cycles should be monitor and made to follow road signs, as more volume of accidents are occurring because of them.

Traffic should be properly regulated on weekends and prime hours (5PM-8PM)

**7.2 FUTURE SCOPE**

• We can further integrate this model with GIS System to get the real time analysis.

• We can find the cluster areas in this system which is used to detect accident prone areas.

• Sharing this with the police departments could be useful so that they can take necessary precautions to avoid accidents.

**CHAPTER 8**

**REFERENCE**

[1] Older Pedestrian Traffic Crashes Severity Analysis Based on an Emerging ML

<https://ideas.repec.org/a/gam/jsusta/v13y2021i2p926-d482269.html>.

[2] Prediction and Analysis of the Severity and Number of Suburban Accidents Using Logit Model, Factor Analysis and Machine Learning

<https://link.springer.com/article/10.1007/s42452-020-04081-3>

[3] Traffic Accident Injury and Severity Prediction Using Machine Learning Algorithms

<https://link.springer.com/chapter/10.1007/978-981-15-7961-5_96>

[4] Model Evaluation for Forecasting Traffic Accident Severity Using Machine Learning Algorithms: Seoul City Study- 2022

https://www.mdpi.com/2076-3417/10/1/129

ww.mdpi.com/2076-3417/10/