CHAPTER 1

INTRODUCTION

1.1 Overview

Road accidents are unquestionably the most frequent cause of damage. It's one of the most significant causes of the fatalities. The reasons for this are the extremely dense road traffic and the relatively great freedom of movement given to drivers. Accidents that involve heavy goods vehicles (like Lorries, trucks) and even the commercial vehicles with the public transportations like buses are one of the most fatal kinds of accidents that occur, claiming the lives of innocent people.

Highways are always a soft spot for these accidents with injuries and deaths. Various weather conditions like rain, fog etc play a role in catalysing the risk of accidents. Having a proper estimate of accidents and knowing the hotspots of accidents and its factors will help to reduce them. Providing timely emergency support even when the casualties have occurred is needed, and to do that a keen study on accidents is required.

In spite of having set regulations and the highway codes, negligence of people towards the speed of the vehicle, the vehicle condition and their own negligence of not wearing helmets has caused a lot of accidents. These accidents wouldn't have turned fatal, and claimed innocent lives if people had governed by the rules. Prevention of road accidents is significantly important and will be fortified by strict laws, by technical and police controls, tougher training for drivers to issue the driving licence and creating a sense of awareness among people as to how important it is to take these rules seriously by imposing penalties and legalities for people responsible.

Motorization has enhanced the lives of many individuals and societies, but the benefits have come with a price. Although the number of lives lost in road accidents in high-income countries indicate a downward trend in recent decades, for most of the world's population, the burden of road-traffic injury—in terms of societal and economic costs—is rising substantially.[1] Injury and deaths due to road traffic accidents (RTA) are a major public health problem in developing countries where more than 85% of all deaths and 90% of disability-adjusted life years were lost from road traffic injuries.[2].As a developing country, India is no exception. Not a day passes without RTA happening in the roads in India in which countless number of people are killed or disabled. Often members of the whole family are wiped out. Those who are affected or killed are mostly people in their prime productive age.

Road traffic accidents (RTAs) have emerged as an important public health issue which needs to be tackled by a multi-disciplinary approach. The trend in RTA injuries and death is becoming alarming in countries like India. The number of fatal and disabling road accident happening is increasing day by day and is a real public health challenge for all the concerned agencies to prevent it. The approach to implement the rules and regulations available to prevent road accidents is often ineffective and half-hearted. Risk estimation of accident based on various factors so that the accidents can be minimized is the need of the hour to prevent this public health catastrophe.

1.2 Objectives

The main objective of the project is to develop a model that mines valuable information from the given data that can be used to solve problems so that it can help to curb accidents

- Association Rule Mining - Association rules are if-then statements that help to show the probability of relationships between data items within large data sets in various types of databases. It also helps to find important hidden patterns or relationships within the data that can be useful.
- Clustering - The clustering technique will cluster the road accident data into two clusters that estimates the risk associated with various areas of Bengaluru so that alerts can be raised by the traffic department based on the risk predicted.
- Reporting of anonymous accidents: This can help people who are often scared to inform about the accidents occurring due to public legalities and this data will help in further data mining.

Overall a data mining model will help data analysts of RTO, emergency sectors, to perform the rule mining, clustering and analytics with ease by varying the data and pre-processing it according to their need.

1.3 Problem Statement

The main objective of the project is to develop a model that mines valuable information from the given data that can be used to solve problems so that it can help to curb accidents. The previous historical data is used to solve these problems. We develop a model that performs the data mining on the data. Performing clustering to predict risk on

various areas of Bengaluru and association rule mining for finding interesting patterns from the data and other analytics in order to help the data analyst of RTO and other sectors.

1.4 Limitations

- Data Acquisition: Machine Learning requires massive data sets to train on, and these should be inclusive/unbiased, and of good quality. There can also be times where they must wait for new data to be generated.

- Time and Resource: ML needs enough time to let the algorithms learn and develop enough to fulfil their purpose with a considerable amount of accuracy and relevancy. It also needs massive resources to function. This can mean additional requirements of computer power.

1.5 Organization of the Report

This report is organized as 6 chapters, namely, introduction, analysis, design, implementation, testing and lastly conclusion and future enhancements.

Chapter 1 gives a brief introduction about the need for road safety, problem statement, objectives, limitations and the literature survey in this field is mentioned.
Chapter 2 deals with the analysis part of development. Details of the existing system, its drawbacks, the proposed system, its advantages, the functional and non- functional as well as the hardware and software requirements are specified.
Chapter 3 specifies the design details. Design is the process of establishing a system that will satisfy the previously identified functional and non- functional requirements. It has 2 parts, the system design and the detailed design. It contains a mention of the system block diagram or the architecture, and various diagrams like the use case diagram and activity diagram.
Chapter 4 includes the implementation part. Implementation is the process of converting the system design into an operational one. This phase starts after the completion of the development phase and must be carefully planned and controlled as it is a key stage. It includes a list of main packages, some of the user- defined functions and some sample code.
Chapter 5 includes the Testing part which is an investigation conducted to provide stakeholders with information of the quality of the product or service under test. It also gives a business an opportunity to understand the risks of software implementation. Test

techniques include, but are not limited to the process of executing a program or application with the intent of finding software bugs.

Chapter 6 mentions the conclusion and future enhancements for the project. Also are mentioned the glossary, acronyms, a brief description of the language python and the bibliography.

CHAPTER 2

LITERATURE SURVEY

Ayushi Jain, Garima Ahuja, Anuranjana, Deepti Mehrotra[3]were presented analysis of road accidents based on Data Mining technique. The objective of this paper is to have data mining to come to aid to create a model that not only smoothes out the heterogeneity of the data by grouping similar objects together to find the accident prone areas in the country with respect to different accident-factors but also helps determine the association between these factors and casualties. As per the report by WHO, the vulnerable road users (motorcyclists, cyclists and pedestrians) account for half of the world's accident led fatalities; with two-wheeler occupants accounting for about 31% of deaths. It goes on to state that adults account for 59% of the total fatalities globally [4].In this paper, we study the states and the union territories of India against the contributing causes and the educational background of the driver to draw efficient conclusions in order to facilitate road safety in the country. Author has made use of dataset which is selected from data.gov.in. The data includes from all the union territories and states and analysed for 58 attributes like total number of accidents, number of people killed and number of people injured due to various factors like alcohol, speeding, driver's fault, type of vehicles etc. [5]. This paper makes use of K-Means Clustering to group similar object off of the heterogeneous data (four clusters are formed by writers to analyse the road accident).From this paper we can substantiate that cluster analysis helps us to determine the accident prone states and territories of India and further these clusters can be labelled to be classified with the help of decision tree to conclude the dominant factor, backing the accidents.

Ms. Gagandeep Kaur predicted accident prone locations on roads by using Data Mining Techniques [6]. This paper sheds light on predicting the probability of accidents on roads with special emphasis on State Highways and Ordinary District Roads by estimating the severity of accidents based on the type of accident, type of spot using the R tool. It is important to find structure in unlabelled data, so the author has made use of clustering to find the patterns in dataset, in particular Self Organizing Map clustering algorithms are applied on accidental data. Simulation is performed by using R-Studio which is an Integrated Development Environment (IDE) for R tool. Correlation analysis and exploratory visualization techniques has been applied on various parameters of accidental data of roads to analyse and predict the useful results which help to minimize the frequency

of accidents and determine the road conditions. Exploratory visualization is an open process where the user has no set goal and/or is looking for no particular outcome their intention is to understand their data better and perhaps to satisfy their curiosity [7]. Correlation is bivariate analysis that measures the strength of association between two variables [6]. From this paper we can substantiate that Correlation analysis examines the road conditions that help to derive the relation between two numerical variables i.e. length and progress of road which is negative that shows the inverse relation. From the study of Exploratory visualization techniques, we can justify that accident on State Highways occur on Straight roads and on Ordinary District Roads the accidents can occur on other type of spots such as Cross-intersection, R-intersection and Straight road but majorly on Cross- intersection. Mainly the type of accident that occurs is Head on collision type on both Roads.

Authors Irina Makarova KseniaShubenkova, Eduard and Mukhametdinov were highlighted the importance of road quality and infrastructure [8]. According to World Health Organization (WHO) [9] mortality in road accidents directly depends on the level of development of the country. According to the WHO report, in Europe, the richest countries have the lowest mortality rate in accidents. Some data of mortality in the Third World countries: in Mexico, the chance to get into a fatal accident is 12.3%; in Pakistan, you can die in an accident with a probability of 14.2%; Albania – 15.4%; Afghanistan - 15.5%;China, Tajikistan, Russia – 18.8%; Armenia – 18.9%; African countries – 26.6%. This paper reveals that the most common causes of death in road traffic accidents in 2015 were: head-on or lateral collision; poor quality of the road pavement; hitting a pedestrian; vehicle roll-over; faulty state of the vehicle other factors that may contribute to road accidents are Aggressive and careless driving, Distracting circumstances, for example, making a phone call, Alcoholic intoxication, Disregard safety rules: Many drivers consciously do not want to use the seatbelts and are ready to pay fines. The article provides an analysis of the global trends in the field of city transport systems' safety. It is shown that the decrease in the safety of traffic is one of the consequences of the growth of motorization. The efficiency of measures to prevent traffic accidents is analysed from the viewpoint of their role in the process to ensure safety and sustainability of the urban transportation system. In this article authors have illustrated the application of Haddon matrix to improve Road Safety. The Haddon Matrix is the most commonly used paradigm in the injury prevention field. Developed by William Haddon in, the matrix looks at factors related to personal attributes, vector or agent attributes and environmental attributes; before, during

and after an injury or death. By utilizing this framework, one can then think about evaluating the relative importance of different factors and design interventions[10].

Yannis T.H. was presented A Review of The Effect of Traffic and Weather Characteristics on Road Safety [11]. Despite the existence of generally mixed evidence on the effect of traffic parameters, a few patterns can be observed. For instance, traffic flow seems to have a non-linear relationship with accident rates, even though some studies suggest linear relationship with accidents. Regarding weather effects, the effect of precipitation is quite consistent and leads generally to increased accident frequency but does not seem to have a consistent effect on severity. The impact of other weather parameters on safety, such as visibility, wind speed and temperature is not found straightforward so far. The increasing use of real-time data not only makes easier to identify the safety impact of traffic and weather characteristics, but most importantly makes possible the identification of their combined effect. The more systematic use of these real-time data may address several of the research gaps identified in this research.

CHAPTER 3

SYSTEM ANALYSIS

Feasibility Study

The feasibility of the project is analysed in this phase and business proposal is put forth with a very general plan for the project. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the end user. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- Economical Feasibility
- Technical Feasibility
- Social Feasibility

Economical Feasibility

This study is carried out to check the economic impact that the system will have on the organization or the user.

Technical Feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

Social Feasibility

The aspect of study is to check the level of acceptance of the system by the user. The user must not feel threatened by the system, instead must accept it as a necessity. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

3.1 Existing System

The existing system is described as below.

3.1.1 Description

The first step in the existing system is collecting the data. Later this data is converted from raw to understandable parameters. From this the required data set is obtained. Next comes the application of unified data mining processes which is classification, clustering and visualization. Various studies-pivoting on the conventional statistical methods-have been carried out in order to analyse road accidents. The statistical approach used to create a crash prediction model fails to consider the uncertainty factor associated with it. Road and traffic accidents are uncertain and unpredictable incidents and their analysis requires the knowledge of the factors affecting them. Road and traffic accidents are defined by a set of variables which are mostly of discrete nature. The major problem in the analysis of accident data is its heterogeneous nature. The existing system had very less measures to resolve the heterogeneity of the data. Thus heterogeneity must be considered during analysis of the data otherwise, some relationship between the data may remain hidden. Therefore we must find a way to filter the data. Another issue of the existing system is that it is static dynamic entry or retrieval of the data does not occur. Proper resource for trend analysis of accidents is also lacking in this system.

3.1.2 Drawbacks

- Model has not focused on identifying the accident frequency.
- There is no focus on exposing key trends and hidden patterns.
- The model does not estimate the probability of risk value.

3.2 Proposed System

The proposed system is described as below.

3.2.1 Description

We propose a system that is based on the cluster analysis using K means algorithm and association rule mining using Apriori algorithm. Using cluster analysis as a preliminary task can group the data into different homogeneous segments. Association rule mining is further applied on the entire data set to generate association rules. In the best of our knowledge, it is the first time that both the approaches have been used together for analysis of road accident data. The result of the analysis proves that using cluster analysis as a preliminary task can help in removing heterogeneity to some extent in the road accident

data. The basic plan layout of the system is that first the data set is obtained later this obtained data set is cleaned via preprocessing. After preprocessing the required data set is obtained which is later subjected to clustering and association rule mining. After all this the last stage is the risk estimation. Clustering is basically forming clusters based on similar groupings.. In Association rule mining we find interesting patterns and relationships between various attributes. The risk estimation model that will give details about the risk at various locations of Bengaluru based on the most important parameters and it is depicted in the form of low and high risk. Rule mining model will help to identify the most important attributes within a factor that is responsible for accidents based on the given data set in the form of rules. Anonymous entry for new accidents to be stored and used for further data mining.

3.2.2 Advantages

- The accident frequency for given area is identified.
- Proper trend analysis is carried out and displayed.
- Risk estimation such as high or low is provided.

.3.3 Requirements Specification

System implementation is the stage of the project where the theoretical design is tuned into a working system. If the implementation system stage is not carefully controlled and planned, it can cause confusion. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the users a confidence that the system will work and be effective.

- Risk Estimation
- Analysis
- Reporting Anonymous Accidents
- Association Rule Mining

3.3.1 Functional requirements

Risk Estimation Module

- In the risk estimation module the model is first trained and once the training is completed the system replies with a finished clustering message and the submit button appears on screen.
- Once all this procedure is completed the user can select the desired area and time categories to know the risk of accidents i.e High or Low.

Analytics Module

- This module mainly provides the trend analysis of the specific selected area.
- There are also other sections such as causes, condition, time of day, hospital and road condition for which analysis is carried out and graphs are provided along with the graph summary.

Reporting Anonymous Accidents

- In this module public can report any accident anonymously without having to provide their identity.
- A form will be displayed where he will have to enter the fields correctly failing which an error message will be generated.
- There are 7 fields in the form namely date, time, location, type of vehicle, type of accident, number of casualties and vehicle number. The user needs to submit these details to report an accident.

Association Rule Mining

- This module majorly concentrates on providing rules for the given support and confidence value.
- Once the user clicks the mining button a window opens where there is option for the user to enter the support and confidence value of his desire based on which rules are mined and displayed.
- In case of any missing field an error message occur.

3.3.2 Non-Functional Requirements

- Availability – The data required to perform data mining is obtained from various sources and it is an integral part in order to perform data mining hence availability of data is the key requirement in our project.
- Maintainability– The datasets can vary from thousands of records to millions of records hence maintaining these records and prevention of any data corruption or discrepancy is required for the maintainability of the datasets.
- Good data analyst– This is also of significant importance because data analysts are the ones that perform the data mining of the data. They control a big fraction of the entire program and they are the ones who completely understand the attributes of the system. A good data analyst should perform these things with ease and hence quality data analysts are required.

3.3.3 Hardware Requirements

- Processor : i3 generation and above
- Speed : 2.6 GHz and above
- RAM : 3GB
- Hard Disk : 20 GB

3.3.4 Software Requirements

- Operating System : Windows 7 and above
- Application : Anaconda distribution python 3.6 and above
- Front End : Tkinter module for python
- Script : Python.
- Development environment : Spyder IDE
- Datasets : MS EXCEL less than 2016

CHAPTER 4

DESIGN

4.1 System Design

The system design is described as below.

• System

A system is an orderly group of interdependent components linked together according to a plan to achieve a specific objective. Its main characteristics are organization, interaction, interdependence, integration and a central objective.

• System Analysis

System analysis and design are the application of the system approach to problem solving generally using computers. To reconstruct a system the analyst must consider its elements output and inputs, processors, controls, feedback and environment.

• Analysis

Analysis is a detailed study of the various operations performed by a system and their relationships within and outside of the system. One aspect of analysis is defining the boundaries of the system and determining whether or not a candidate system should consider other related systems. During analysis data are collected on the available files decision points and transactions handled by the present system. This involves gathering information and using structured tools for analysis.

4.1.1 Design Overview

The architecture used is data mining architecture. Data mining Architecture system contains too many components. That is a data source, data warehouse server, data mining engine, and knowledge base.

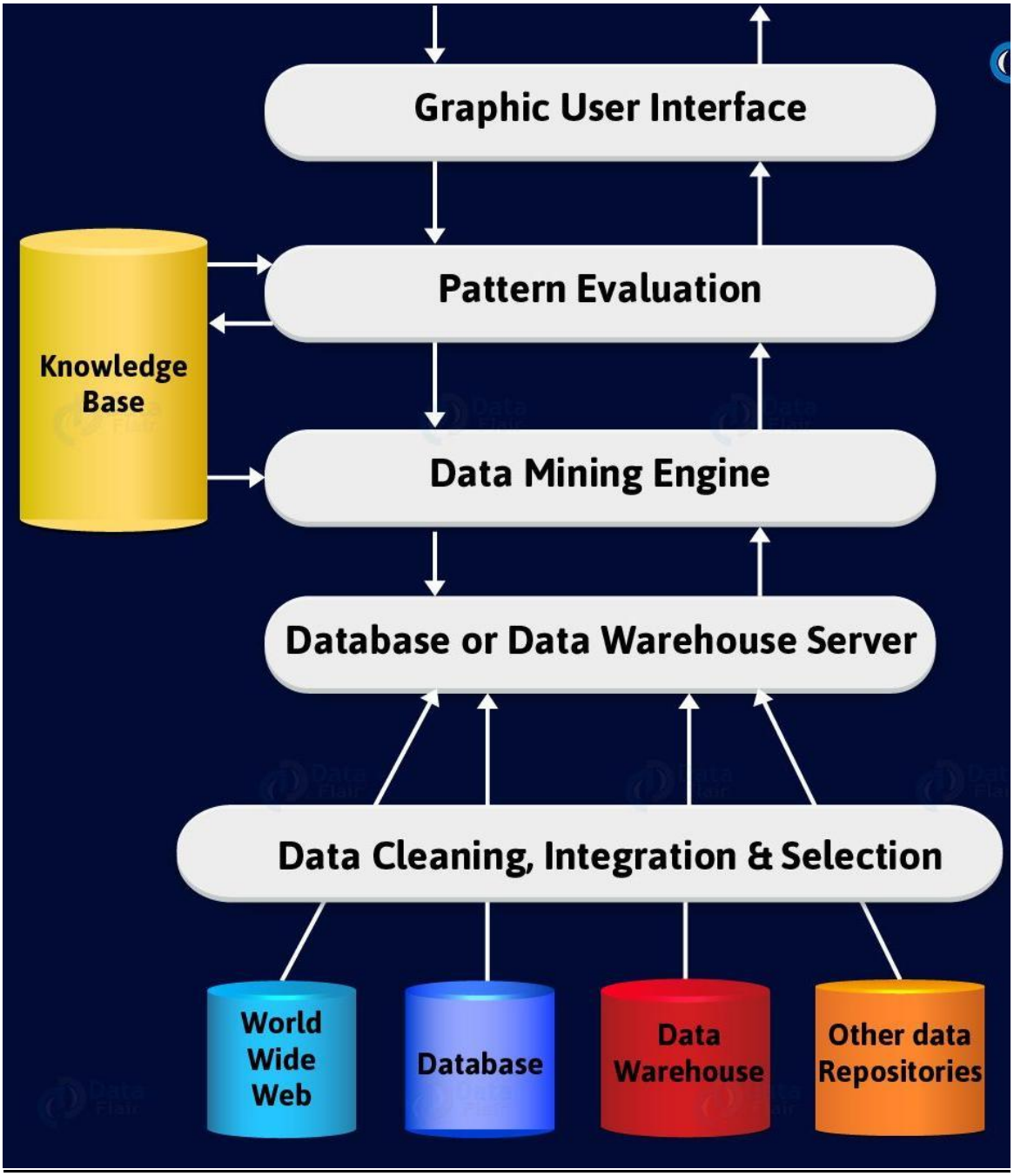


Figure 4.1:A Data Mining Framework

Data Sources

There are so many documents present. That is a database, data warehouse, World Wide Web (WWW). That are the actual sources of data. Sometimes, data may reside even in plain text files or spreadsheets. World Wide Web or the Internet is another big source of data.

Database or Data Warehouse Server

The database server contains the actual data that is ready to be processed. Hence, the server handles retrieving the relevant data. That is based on the data mining request of the user.

Data Mining Engine

In data mining system data mining engine is the core component. As It consists a number of modules. That we used to perform data mining tasks. That includes association, classification, characterization, clustering, prediction, etc.

Pattern Evaluation Modules

This module is mainly responsible for the measure of interestingness of the pattern. For this, we use a threshold value. Also, it interacts with the data mining engine. That's main focus is to search towards interesting patterns.

Graphical User Interface

We use this interface to communicate between the user and the data mining system. Also, this module helps the user use the system easily and efficiently. They don't know the real complexity of the process. When the user specifies a query, this module interacts with the data mining system. Thus, displays the result in an easily understandable manner.

Knowledge Base

In whole data mining process, the knowledge base is beneficial. We use it to guiding the search for the result patterns. The knowledge base might even contain user beliefs and data from user experiences. That can be useful in the process of data mining. The data mining engine might get inputs from the knowledge. That is the base to make the result more accurate and reliable. The pattern evaluation module interacts with the knowledge base. That is on a regular basis to get inputs and also to update it.

4.1.2 System Architecture

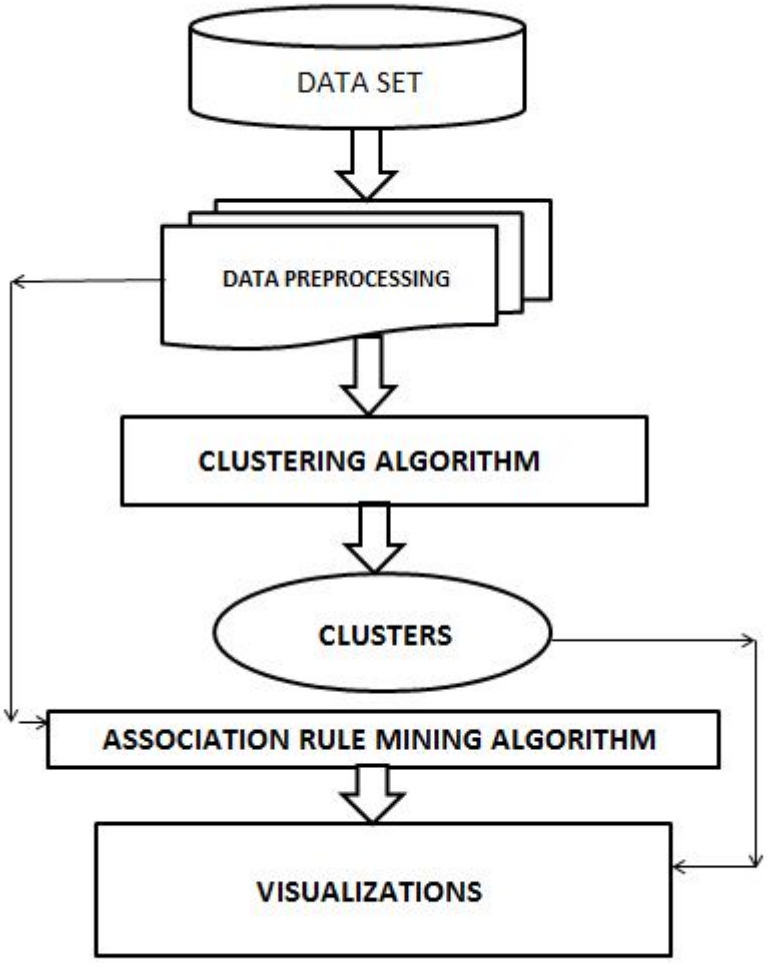


Figure 4.2 :System Architecture

Data pre-processing

Data preprocessing is one of the important tasks in data mining. Data pre-processing mainly deals with removing noise, handle missing values, removing irrelevant attributes in order to make the data ready for the analysis. In this step, our aim is to pre-process the accident data in order to make it appropriate for the analysis.

Clustering algorithm

There are several clustering algorithms exist in the literature. The objective of clustering algorithm is to divide the data into different clusters or groups such that the objects within a group are similar to each other whereas objects in other clusters are different from each

other. Hierarchical clustering technique (e.g. Ward method, single linkage, complete linkage, etc.), K means variant of E-M algorithm have been used in road accident analysis.

K-means clustering procedure

In order to cluster the data set D into k cluster, K-means clustering algorithm perform the following steps:

1. Initially select k random objects as cluster centers or modes.
2. Find the distance between every object and the cluster centre using distance measure .
3. Assign each object to the cluster whose distance with the object is minimum.
4. Select a new center or mode for every cluster and compare it with the previous value of centre or mode; if the values are different, continue with step 2.

Association rules

Association rule mining is a very popular data mining technique that extracts interesting and hidden relations between various attributes in a large data set. Association rule mining produces a set of rules that define the underlying patterns in the data set. The associativity of two characteristics of accident is determined by the frequency of their occurrence together in the data set. A rule A → B indicates that if A occurs then B will also occur.

Visualizations

The various graphs predicting the risk and rules generated by applying association algorithm and statistical visualization from dataset is visualized to the user through the user interface.

4.1.3 Use Case Diagram

A Use Case is a list of actions or event steps, typically defining the interactions between a role and a system, to achieve a goal. The actor can be a human or other external system. A use case diagram in the Unified Modelling Language (UML) is a type of behavioural diagram defined by and created from a use case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

Here in this System use case diagram shown in the figure. The analyst is able to generate the association rules based on the support value given by the user, make risk predictions and analytical patterns. User will be able to view all the analytical graphs , and

obtain the set of rules based on support value of requirement. User can view the risk level as high or low depending on various factors.

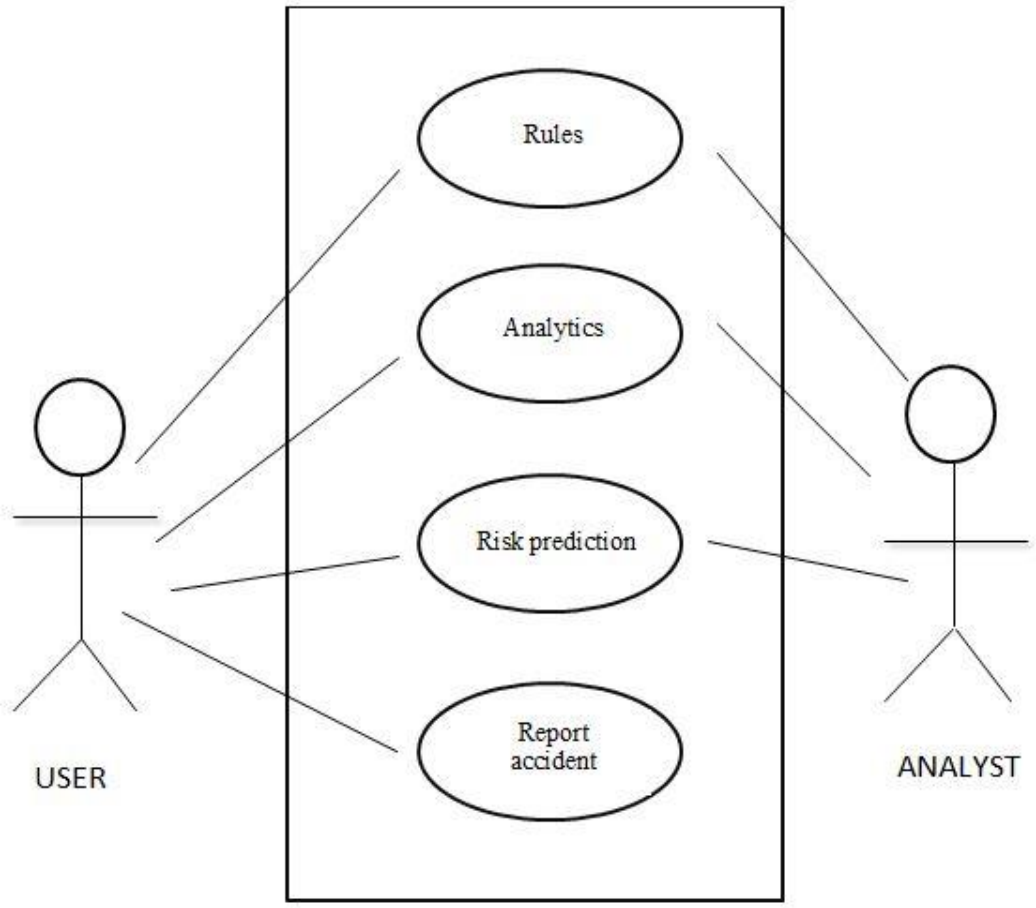


Figure 4.3 Use Case Diagram

4.2 Detailed Design

A detailed design is sometimes referred as developed design or definition. It is the process of taking on and developing the approved concept design.

By the end of the detailed design process, the design should dimensionally correct and co-ordinate, describing all the main components of the building and how they fit together. However, technical aspects of the design may require further development, design by specialists may not yet have been fully incorporated into the design and it will not have been packaged for tender. Detailed design should provide sufficient information for application for statutory approval to be made.

4.2.1 Activity Diagram

An activity diagram shows the sequence of steps that make up a complex process, such as an algorithm or workflow. An activity diagram shows flow of control, similar to a sequence diagram, but focuses on operations rather than on objects. Activity diagram are most useful during the early stages of designing algorithms and workflows.

Activity diagram are graphical representations of workflows of stepwise activities and action with support for choice, iteration and concurrency. An activity diagram shows the overall flow of control.

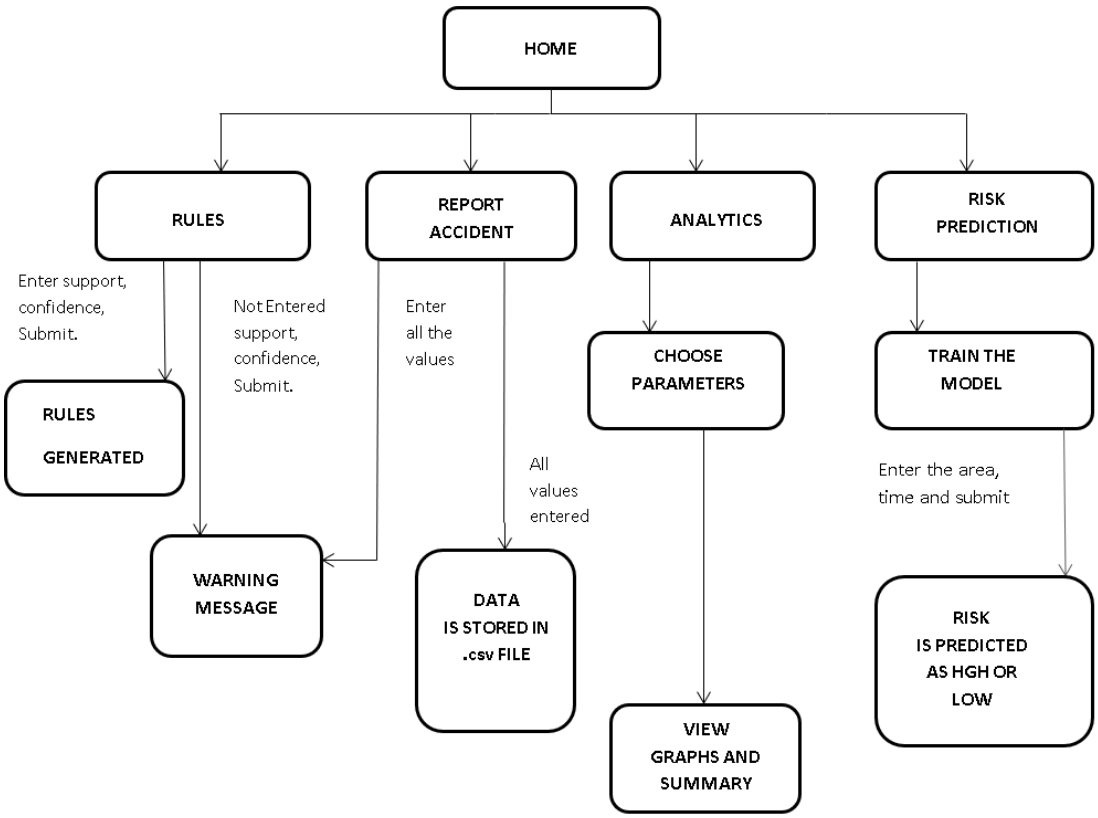


Figure 4.4 Activity diagram of overall system.

Here the activity diagram shows the sequence of activities that are involved in the system which is described as below.

The graphical user interface(GUI) provides the homepage of the system that helps to navigate through the system. When user performs an activity, the changes will be reflected in the system at every state.

When user navigates from home page to the rules pages through the navigating rule button , the rule page opens. On entering the confidence and support values and clicking on mine, the apriori algorithm is performed and rules are generated.

If any values are not entered, the system shows a warning message of values missing. If the user navigates to the report accident button, the report anonymous accident page opens where the user enters the values for the various parameters like date, time , no of casualties, type of accident etc. On leaving the parameters values empty a warning message will be displayed stating missing values. This is done to prevent nan values from the system. On submitting successfully, the data will be stored in the .csv data file.

On navigating from the analytics button, the various types of graph options is presented for the user to select and analyse. For analysis of trends, we select the state for which the trend analysis has to be done and the respective trend graph will be displayed along with the summary of the graphs.

When user navigates from the risk prediction button, risk prediction page is displayed. User now trains the model and after the model is trained the message is displayed about the model being trained. User selects the area and time category to view the risk predicted in that area as either high or low.

CHAPTER 5

IMPLEMENTATION

5.1 Packages

Some of the main packages used in this project are as mentioned below

Table 5.1 List of Packages

Packages	Description
pandas	Pandas is a package for data manipulation and analysis. In particular, it offers data structures and operations.
numpy	NumPy stands for 'Numerical Python' or 'Numeric Python'. It is an open source module of Python which provides fast mathematical computation on arrays and matrices.
matplotlib.pyplot	Matplotlib is a 2d plotting library which produces publication quality figures in a variety of hard copy formats and interactive environments.
apyori.apriori	apyori package is used for association rule mining using apriori algorithm.
sklearn.cluster	It is a machine learning clustering package for forming k clusters with centroids.
seaborn	Seaborn is a Python data visualization library . It provides a high-level interface for drawing attractive and informative statistical graphics.
tkinter	It is a python standard GUI package for user interface from Tk (toolkit)

5.2 User -Defined Functions

A User-Defined Function (UDF) is a function provided by the user of a program or environment, in a context where the usual assumption is that functions are built into the program or environment. Below mentioned are the most important user-defined functions used.

Table 5.2 Function ruleplot ( ) details

Function Name	ruleplot( )
Syntax	def ruleplot( )
Description	This method is used for association rule mining.
Parameters	support , confidence
Calling Function	Rule_display()
Return Type	list [ ]

Table 5.3 Function new_data( ) details

Function Name	new_data( )
Syntax	def new_data( )
Description	This method is called to report anonymous accidents
Parameters	Date,Time,Location,Type of vehicle,Type of accident, No of casualties,Vehicle number
Calling Function	Button(root, text = "Report accident",command=new_data)
Return Type	None

Table 5.4 Function writetocsv () details

Function Name	writetocsv( )
Syntax	def writetocsv( )
Description	This method is used to write the anonymous entry into a file
Calling Function	new_data()
Return Type	None

Table 5.5 Function risk_predict ( ) details

Function Name	risk_predict( )
Syntax	def risk_predict( )
Description	This function is called to estimate the risk based on the area
Parameters	StringVar
Calling Function	Button(root, text = "Risk prediction",command=risk_predict)
Return Type	StringVar

Table 5.6 Function updateRequest ( ) details

Function Name	train ( )
Syntax	def train ( )
Description	This function performs k means clustering.
Calling Function	risk_predict( )
Return Type	Cluster

Table 5.7 Function plot_graph( ) details

Function Name	plot_graph( )
Syntax	def plot_graph( )
Description	This function is called when user wants to view various analytical graphs
Calling Function	Button(root, text = "Analytics",command=plot_graph)
Return Type	Image

Table 5.8 Function graph1( ) details

Function Name	graph1( )
Syntax	def graph1( )
Description	This function is called when user wants to view a graph
Calling Function	plot_graph( )
Return Type	Image

Table 5.9 Function graph4( ) details

Function Name	graph4( )
Syntax	def graph4( )
Description	This function is called when user wants to view a graph based on the state.
Parameters	Area
Calling Function	plot_graph( )
Return Type	Image

5.3 Built-in Functions

Some of the main built in functions used in this project are as mentioned in below table.

Table 5.10 Function stringVar ( ) details

Function Name	stringVar( )
Parameters	String
Return Type	String
Description	used to hold a string from GUI variables

Table 5.11 Function KMeans ( ) details

Function Name	KMeans(n_clusters,n_init,max_iter,algorithm)
Parameters	n_clusters,n_int, max_iter, algorithm
Return Type	GeneratorObject
Description	to form k clusters based on centroid centers

Table 5.12 Function apriori ( ) details

Function Name	apriori(transactions,min_support,min_confidence,min_lift)
Parameters	transactions,min_support,min_confidence,min_lift
Return Type	GeneratorObject
Description	applies apriori algorithm to the data

Table 5.13 Function read_csv( ) details

Function Name	read_csv( )
Parameters	Filename
Return Type	DataFrame
Description	Reads a comma separated values file (csv) into pandas dataframe

Table 5.14 Function append( ) details

Function Name	append( )
Parameters	item
Return Type	-
Description	To add an item to the end of the list

Table 5.15 Function to_datetime( ) details

Function Name	to_datetime( )
Parameters	arg
Return Type	Datetime
Description	converts passed argument to datetime

Table 5.16 Function value_counts( ) details

Function Name	value_counts( )
Parameters	sort,ascending,dropna
Return Type	Series
Description	Returns a series containing count of unique values

Table 5.17 Function plot ( ) details

Function Name	plot( )
Parameters	x, y, data ,c
Return Type	Lines
Description	Returns a line plot of x versus y on data

Table 5.18 Function fillna( ) details

Function Name	fillna( )
Parameters	value,inplace
Return Type	DataFrame
Description	used to fill the NA/NaN values in a dataframe

5.4 ALGORITHM

Association Rule Mining

Step 1: let support and confidence be two input variables

Step 2: read the csv file on which rule mining is done and store it in a data frame using pd.read_csv()

Step 3: preprocess data by performing feature selection

Step 4: compute association_rules by applying apriori algorithm with input values of min_support,min_confidence and min_lift=1.0

Step 5: store the association results as a list of association_rules

Step 6: for item in association results do

append rules as item -> item [0]

Step 7: print the rules as computed when clicked on the mine button

Analytics to plot graphs

Step 1: select the analytics button to view various graphs

Step 2: select the area for viewing the trend graph

Step 3: read the csv file on which analytics has to be done and store it in a data frame

Step 4: plot the graphs based on the selected option

Step 5: store it in the form of an image and display on the front end

Step 6: Summary generated from the graphs.

Risk Estimation algorithm

Step 1: read the cas data of Bengaluru that has speed, area etc.

Step 2: convert the timestamp to datetime format using to_datetime()

Step 3: based on the time of hour we categorize the observation of vehicle time into peak hours,early hours,regular hours.

Step 4: get the weather data during when the observations were made.

Step 5: merge the weather data and cas data into single dataframe.

Step 6: obtain the accident prone zones of bengaluru and pothole data and map them to the areas and merge them with with previous data frame

Step 7: fill the missing values(NaN) with mode values of the column with function fillna()

Step 8: map all the columns to its numerical equivalent in the form of flags.

Step 9: write the entire data to consolidated excel sheet

Step 10: apply k-means clustering with EM variant by specifying algorithm=full and max_iter = 2000 and n_clusters=2

E-step, where each object is assigned to the centroid such that it is assigned to the most likely cluster.

M-step, where the model (=centroids) are recomputed.

Step 11: fit data into the clusters so that data converge to the centroids.

Step 12: based on the clusters formed predict the risk estimated as low or high.

5.5 Sample Code

Some of the sample code used is specified below

```
import pandas as pd

import numpy as np

from tkinter import *

from tkinter import ttk

from tkinter.filedialog import askopenfilename

root = Tk()

root.title('Data Mining Based Risk Estimation of Road Accidents')

root.geometry('850x650')

root.configure(background="white")

var = StringVar()

label = Label( root, textvariable = var,font=('arial',20,'bold'),bd=20,background="white")

var.set("Data Mining Based Risk Estimation of Road Accidents")

label.grid(row=0,columnspan=6)

def Rule_display():

    root1=Tk()

    root1.title("Rules")

    root1.geometry('800x800')

    root1.configure(background="white")

    label_6 = ttk.Label(root1, text = 'support',font=("Helvetica", 16),background="white")

    label_6.grid(row=1,column=0)

    Entry_6 = Entry(root1)

    Entry_6.grid(row=1,column=1)
```

```
    label_7 = ttk.Label(root1, text = 'confidence',font=("Helvetica", 16),
background="white")
     label_7.grid(row=2,column=0)
    Entry_7 = Entry(root1)
    Entry_7.grid(row=2,column=1)
    root1.mainloop()
def plot_graph():
    root10 = Tk()
    root10.title('GRAPHS')
    root10.geometry('400x400')
    root10.configure(background="white")
    from PIL import ImageTk,Image
    def graph1():
        def graph1():
        df=pd.read_csv('3g.csv')
        ncount=df['Nature of Accident'].value_counts()
        ncount=ncount[:10,]
        plt.figure(figsize=(10,5)
        sns.barplot(ncount.index,ncount.values,alpha=0.8)
        plt.title('Cause of accident')
        plt.xlabel('causes')
        plt.ylabel('occurance')
        plt.savefig('graph1.png',dpi=199)
        image = Image.open("graph1.png")
```

```
        image = image.resize((550, 250), Image.ANTIALIAS)

        img = ImageTk.PhotoImage(image)

        panel1 = Label(root, image=img)

        panel1.image = img

        panel1.grid(row=3,column=0)

        Text_aa.place(x=50,y=620,height=200,width=600)

        Text_aa.delete('1.0',END)

        for i,j in zip(ncount.index,ncount.values):

            analysis=" > the cause of accident is \t "+ str(i) +"with \t" +str(j)+" occurances and "+str(j/806)+" probability \n"

            Text_aa.insert(INSERT,analysis)

B_rule = Button(root, text = "Rules",height=1,padx=16,pady=16,bd=8,font=('arial',16,'bold'),width=10,bg="white",command=Rule_display)

B_rule.grid(row=1,column=1)

B = Button(root, text = "Risk prediction",height=1,padx=16,pady=16,bd=8,font=('arial',16,'bold'),width=10,bg="white",command=risk_predict)

B.grid(row=2,column=1)

label_00 = ttk.Label(root,background="white")

label_00.grid(row=1,column=3)

B1 = Button(root, text = "Analytics",height=1,padx=16,pady=16,bd=8,font=('arial',16,'bold'),width=10,bg="white",command=plot_graph)

B1.grid(row=1,column=3)
```

```
B3 = Button(root, text = "Report
accident",height=1,padx=16,pady=16,bd=8,font=('arial',16,'bold'),width=10,bg="white",c
ommand=new_data)

B3.grid(row=2,column=3)

root.mainloop()
```

Code for Rule mining

```
if not Entry_6.get():
        messagebox.showwarning("missing value","please range the support value")
    elif not Entry_7.get():
        messagebox.showwarning("missing value","please range the confidence value")
    elif(not Entry_6.get() or not Entry_7.get()):
        messagebox.showwarning("missing value","please range the support or
confidence value")
    import matplotlib.pyplot as plt
    import matplotlib.cm as cm
    from apyori import apriori
    import pandas as pd
    import numpy as np
    df=pd.read_csv('3g.csv')
    df.shape
    df.columns
    del df['SrNo']   del df['Fatal']   del df['Grevious']   del df['Minor']   del df['Injured']
del df['Date']   del df['a']
     records = []
    for i in range(0, 807):
```

```
        records.append([str(df.values[i,j]) for j in range(0, 10)])
    association_rules = apriori(records, min_support=float(Entry_6.get()),
min_confidence=float(Entry_7.get()), min_lift=1.0)
    association_results = list(association_rules)
    for item in association_results:
            pair = item[0]
        items = [x for x in pair]
    import random
    import matplotlib.pyplot as plt


    support=[]
    confidence=[]
    lift=[]
    color=[]
    for a in association_results:
        support.append(a[1])
        confidence.append(a[2][0][2])
        lift.append(a[2][0][3])
        color.append(a[2][0][3]*20.0)
    print(len(support))
    rules = []
    for item in association_results:
        # first index of the inner list
        # Contains base item and add item
```

```
            pair = item[0]
            items = [x for x in pair]
            rules.append("Rule: " + str(items)+"->"+ items[0]+"\n")
            #print("Rule: " + str(items)+"->"+ items[0])
            print(rules)
        rules = "".join(x for x in rules)
        Text_rule.delete('1.0',END)
        Text_rule.insert(INSERT,rules)
    Text_rule = Text(root1)
    Text_rule.grid(row=3,column=1)
    B3 = Button(root1, text =
"mine",height=1,padx=16,pady=16,bd=8,font=('arial',16,'bold'),width=10,bg="white",co
mmand=ruleplot)
    B3.grid(row=4,column=1
  root1.mainloop()
```

Code for reporting anonymous entry

```
data_new = []
def new_data():
    root10 = Tk()
    root10.title('Report Anonymous Accidents')
    root10.geometry('400x400')
    root10.configure(background="white")
    label_1 = ttk.Label(root10, text ='Date',font=("Helvetica", 16),background="white")
    label_1.grid(row=0,column=0)

    Entry_1 = Entry(root10)
    Entry_1.grid(row=0,column=1)
```

```
    label_2 = ttk.Label(root10, text = 'Time',font=("Helvetica", 16),background="white")
    label_2.grid(row=1,column=0)

    Entry_2 = Entry(root10)
    Entry_2.grid(row=1,column=1)

    label_3 = ttk.Label(root10, text = 'Location',font=("Helvetica",
16,),background="white")
    label_3.grid(row=2,column=0)

    Entry_3 = Entry(root10)
    Entry_3.grid(row=2,column=1)

    label_4 = ttk.Label(root10, text = 'Type of vehicle' ,font=("Helvetica",
16),background="white")
    label_4.grid(row=3,column=0)

    Entry_4 = Entry(root10)
    Entry_4.grid(row=3,column=1)

    label_5 = ttk.Label(root10, text = 'Type of accident',font=("Helvetica",
16),background="white")
    label_5.grid(row=4,column=0)

    Entry_5 = Entry(root10)
    Entry_5.grid(row=4,column=1)

    label_6 = ttk.Label(root10, text = 'No of Casualities',font=("Helvetica",
16),background="white")
    label_6.grid(row=5,column=0)

    Entry_6 = Entry(root10)
    Entry_6.grid(row=5,column=1)
```

```
    label_7 = ttk.Label(root10, text = 'Vehicle Number',font=("Helvetica",
16),background="white")
    label_7.grid(row=6,column=0)

    Entry_7 = Entry(root10)
    Entry_7.grid(row=6,column=1)
    global model,labelText
    def writetocsv():
        global model,labelText
        if (not Entry_1.get() or not Entry_2.get() or not Entry_3.get() or not Entry_4.get() or
not Entry_5.get() or not Entry_6.get() or not Entry_7.get()) :
            messagebox.showinfo("missing value","enter data")
        else:

data_new.append(Entry_1.get()+","+Entry_2.get()+","+Entry_3.get()+","+Entry_4.get()+
","+Entry_5.get()+","+Entry_6.get()+","+Entry_7.get()+"\n")

            file = open('data.csv','w')
            file.writelines(data_new)
            file.close()

    label_7 = Button(root10, text = 'submit',font=("Helvetica",
16),background="white",command = writetocsv)
    label_7.grid(row=7,column=0)
```

Code for risk estimation

```
root11 = Tk()
    root11.title('RISK PREDICTION')
    root11.geometry('400x400')
    root11.configure(background="white")

    var = StringVar()
    label = Label( root11, textvariable =
var,font=('arial',20,'bold'),bd=20,background="white")
    var.set("RISK PREDICTION")
```

```
    label.grid(row=0,columnspan=6)

    label_1 = ttk.Label(root11, text ='Area',font=("Helvetica", 16),background="white")
    label_1.grid(row=1,column=0)
    tkvar = StringVar(root11)
    choices = ['A
Narayanapura','Agaram','Banasavadi','Basavanapura','Bellanduru','Benniganahalli','Bharat
hi Nagar','BTM Layout','C V Raman Nagar','Chickpete','Devasandra','Dharmaraya
Swamy Temple','Dodda Nekkundi','Domlur','Garudachar
Playa','Gurappanapalya','Hagadur','HAL
Airport','Halsoor','Hemmigepura','Horamavu','Hoysala Nagar','HSR Layout','Hudi','J P
Nagar','Jaraganahalli','Jayanagar East','Jeevanbhima Nagar','Jogupalya','K R
Puram','Kacharkanahalli','Kadugodi','Kammanahalli','Konena
Agrahara','Madivala','Marathahalli','New Tippasandara','Other','other','Ramamurthy
Nagar','Sampangiram Nagar','Sarakki','Shantala Nagar','Singasandra','Sudham
Nagara','Varthuru','Vasanthpura','Vijnana Nagar','Vijnanapura','Yelchenahalli']
    popupMenu = OptionMenu(root11, tkvar, choices[1], *choices)
    popupMenu.grid(row=1,column=1)
    tkvar.set('Select area')

    def train():
        import pandas as pd
        import numpy as pd
        bdf = pd.read_excel('bangalore-cas-alerts.xlsx')
        bdf.info()
        bdf = bdf.rename(columns = {'deviceCode_time_recordedTime_$date':'timestamp'})
        bdf['timestamp'] = pd.to_datetime(bdf['timestamp'])
        bdf['eventDate'] = pd.to_datetime(bdf['timestamp'])
        bdf['eventDate'] = bdf['eventDate'].dt.strftime('%Y%m%d')
        bdf['e_hour'] = pd.to_datetime(bdf['timestamp'], format = '%H:%M:%S').dt.hour
        bdf['ehourCat'] = 0
        bdf['ehourCat'] = np.where((bdf['e_hour'] >= 0) & (bdf['e_hour'] < 6), 1,
bdf['ehourCat'])
        bdf['ehourCat'] = np.where((bdf['e_hour'] >= 6) & (bdf['e_hour'] < 10), 2,
bdf['ehourCat'])
```

```
    bdf['ehourCat'] = np.where((bdf['e_hour'] >= 10) & (bdf['e_hour'] < 16), 3,
bdf['ehourCat'])
    bdf['ehourCat'] = np.where((bdf['e_hour'] >= 16) & (bdf['e_hour'] < 21), 4,
bdf['ehourCat'])
    bdf['ehourCat'] = np.where((bdf['e_hour'] >= 21) & (bdf['e_hour'] < 24), 5,
bdf['ehourCat'])


    bwdf = pd.read_excel('bangalore-weather.xlsx')
    bwdf['w_hour'] = pd.to_datetime(bwdf['time'], format= '%H:%M').dt.hour
    bwdf['hourCat'] = 0
    bwdf['hourCat'] = np.where((bwdf['w_hour'] >= 0) & (bwdf['w_hour'] < 6), 1,
bwdf['hourCat'])
    bwdf['hourCat'] = np.where((bwdf['w_hour'] >= 6) & (bwdf['w_hour'] < 10), 2,
bwdf['hourCat'])
    bwdf['hourCat'] = np.where((bwdf['w_hour'] >= 10) & (bwdf['w_hour'] < 16), 3,
bwdf['hourCat'])
    bwdf['hourCat'] = np.where((bwdf['w_hour'] >= 16) & (bwdf['w_hour'] < 21), 4,
bwdf['hourCat'])
    bwdf['hourCat'] = np.where((bwdf['w_hour'] >= 21) & (bwdf['w_hour'] < 24), 5,
bwdf['hourCat'])
    bwdf = bwdf.drop_duplicates(subset = ['weatherDate', 'hourCat'], keep = 'first')
    bwdf['ehourCat'] = bwdf['hourCat']
    bwdf['weatherDate'] = bwdf['weatherDate'].astype(str)
    bdf['weatherDate'] = bdf['eventDate']
    bdf['weatherDate'] = bdf['weatherDate'].astype(str)
    b1 = pd.merge(bdf, bwdf, on = ['weatherDate', 'ehourCat'], how = 'left')
    b1 = b1.rename(columns = {'deviceCode_location_wardName':'Area'})
    badf = pd.read_excel('bangalore-accident-zones.xlsx')
    b = pd.merge(b1, badf, on = ['Area'], how = 'left')
    b = b.rename(columns = {'deviceCode_pyld_alarmType':'Alarm_Type'})
    b = b.rename(columns = {'deviceCode_pyld_speed':'Plying_Speed'})
    b['hasOversped'] = np.where(b.Plying_Speed > 60, 1, 0)
    b['hasOversped'] = np.where(b.Alarm_Type == 'Overspeed', 1, b['hasOversped'])
    for column in ['temperature', 'visibility', 'condition']:
      b[column].fillna(b[column].mode()[0], inplace=True)
```

```
b['visibility'] = np.where(b['visibility'] < 10, 0, 1)
df = b.copy()
df['hasOversped'] = np.where(b.hasOversped == 1, 'Yes', 'No')
df['visibility'] = np.where(b.visibility == 0, 'Low', 'High')
df['ehourCat'] = b['ehourCat'].map({1: 'Early', 2: 'PeakM', 3: 'RegularM'})
b['Accident_Severity'] = b['Accident_Severity'].map({'High': 3, 'Medium': 2, 'Low': 1})
b['Pothole_Severity'] = b['Pothole_Severity'].map({'High': 3, 'Medium': 2, 'Low': 1})
b['Alarm_Type'] = b['Alarm_Type'].map({'PCW': 1, 'FCW': 2, 'Overspeed': 3, 'HMW': 4, 'UFCW': 5, 'LDWL': 6, 'LDWR': 7})
b['condition'] = b['condition'].map({'Clear': 1, 'Sunny': 2, 'Passing clouds': 3,
    'Broken clouds': 4, 'Scattered clouds': 5, 'Fog': 6, 'Haze': 7, 'Partly cloudy': 8,
    'Mild': 9, 'Drizzle. Broken clouds': 10})
b['Area'] = b['Area'].map({'Kadugodi': 1, 'Garudachar Playa': 2, 'Hudi': 3, 'Other': 4, 'Devasandra': 5,
    'Hagadur': 6, 'Bellanduru': 7, 'Marathahalli': 8, 'Dodda Nekkundi': 9, 'Varthuru': 10,
    'HAL Airport': 11, 'Vijnana Nagar': 12, 'Konena Agrahara': 13, 'A Narayanapura': 14,
    'C V Raman Nagar': 15, 'Jeevanbhima Nagar': 16, 'HSR Layout': 17, 'Domlur': 18, 'Jogupalya': 19,
    'Hoysala Nagar': 20, 'New Tippasandara': 21, 'Benniganahalli': 22, 'Singasandra': 23,
    'Basavanapura': 24, 'Halsoor': 25, 'Agaram': 26, 'Shantala Nagar': 27, 'Sampangiram Nagar': 28,
    'Sudham Nagara': 29, 'Dharmaraya Swamy Temple': 30, 'Chickpete': 31, 'Banasavadi': 32,
    'Horamavu': 33, 'Kacharkanahalli': 34, 'Kammanahalli': 35, 'Vijnanapura': 36, 'Ramamurthy Nagar': 37,
    'K R Puram': 38, 'BTM Layout': 39, 'Madivala': 40, 'Gurappanapalya': 41, 'J P Nagar': 42, 'Sarakki': 43,
    'Jaraganahalli': 44, 'Vasanthpura': 45, 'Hemmigepura': 46, 'Yelchenahalli': 47,
    'Jayanagar East': 48, 'Bharathi Nagar': 49, 'other': 4})

writer = pd.ExcelWriter('bangalore-consolidated-data.xlsx')
```

```
    b.to_excel(writer, index = False, sheet_name = 'Sheet1')
    df.to_excel(writer, index = False, sheet_name = 'Sheet2')
    writer.save()

    del b['deviceCode_deviceCode'], b['deviceCode_location_latitude'],
b['deviceCode_location_longitude']
    del b['w_hour'], b['Mapped_Location'], b['timestamp'], b['e_hour'], b['weatherDate']
    del b['hourCat'], b['time'], b['temperature'], b['eventDate'], b['Plying_Speed']

    del df['deviceCode_deviceCode'], df['deviceCode_location_latitude'],
df['deviceCode_location_longitude']
    del df['w_hour'], df['Mapped_Location'], df['timestamp'], df['e_hour'],
df['weatherDate']
    del df['hourCat'], df['time'], df['temperature'], df['eventDate'], df['Plying_Speed']

    from sklearn.cluster import KMeans
    X = b.values.astype(np.float)
    kmeans = KMeans(n_clusters = 2, max_iter = 2000, algorithm = 'full').fit(X)
    kmf2labels = kmeans.labels_
    kmf2labels = kmf2labels.tolist()
    print('Finished clustering using K-Means')

    b['labels'] = kmf2labels
    df['labels'] = kmf2labels
    df['labels'] = df['labels'].map({0: 'High', 1: 'Low'})

    def main_risk_predict():
        area = tkvar.get()

        if kmf2label['Area'][tkvar.get()] ==:
            Entry_12.delete(0,END)
            Entry_12.insert(0,'HIGH')
        else :
            Entry_12.delete(0,END)
            Entry_12.insert(0,'LOW')
```

```
    label_7 = Button(root11, text = 'submit',font=("Helvetica",
16),background="white",command = main_risk_predict)
    label_7.grid(row=2,column=0)

    Entry_11.delete(0,END)
    Entry_11.insert(0,'Finished clustering using K-Means')

  Entry_12 = Entry(root11)
  Entry_12.grid(row=2,column=1)
  Entry_11 = Entry(root11)
  Entry_11.grid(row=0,column=1)

  label_8 = Button(root11, text = 'train',font=("Helvetica",
16),background="white",command = train)
  label_8.grid(row=0,column=0)
```

Code for analytics:

```
    import numpy as np # linear algebra

    import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

    import os

    import matplotlib.pyplot as plt

    import seaborn as sns

    df=pd.read_csv('Details_of_road_accident_deaths_by_situation_state.csv

    t=df[df['CAUSE']=='Total Truck/Lorry']

    t.head()

    r=t[t['STATE/UT']== tkvar.get()]

    fig = plt.figure(figsize=(20,10))

    ax = plt.axes()

    x = np.linspace(0, 10, 1000)
```

```
        ax.plot(r['Year'],r['Total'])
        plt.title('NO OF ACCIDENTS DUE TO TRUCK AND LORRY')
        plt.xlabel('year')
        plt.ylabel('no of accidents')
        plt.savefig('graph4.png',dpi=199)
        image = Image.open("graph4.png")
        image = image.resize((750, 350), Image.ANTIALIAS)
        img = ImageTk.PhotoImage(image)
        panel3 = Label(root, image=img)
        panel3.image = img
        panel3.grid(row=3,column=0)
    label_1 = ttk.Label(root10,background="white")
    label_1.grid(row=1,column=0)
    var = StringVar()
    label = Label( root10, textvariable =
var,font=('arial',20,'bold'),bd=20,background="white")
    var.set("RISK PREDICTION")
    label.grid(row=0,columnspan=6)
    label_1 = ttk.Label(root10, text ='Area',font=("Helvetica", 16),background="white")
    label_1.grid(row=1,column=0)
    tkvar = StringVar(root10)
    choices = ['ANDHRA
PRADESH','ASSAM','BIHAR','CHHATTISGARH','GOA','GUJARAT','HARYANA','HI
MACHAL PRADESH','JAMMU &
KASHMIR','JHARKHAND','KARNATAKA','KERALA','KERALA','MADHYA
PRADESH','MAHARASHTRA','MANIPUR','MEGHALAYA','MIZORAM','NAGALA
```

```
ND','ODISHA','PUNJAB','RAJASTHAN','SIKKIM','TAMIL
NADU','TRIPURA','UTTAR PRADESH','UTTARAKHAND','WEST
BENGAL','TOTAL (STATES)','A & N ISLANDS','CHANDIGARH','D & N
HAVELI','DAMAN & DIU','DELHI
(UT)','LAKSHADWEEP','PUDUCHERRY','TOTAL (UTs)','TOTAL (ALL INDIA)',]

    popupMenu = OptionMenu(root10, tkvar, choices[1], *choices)

    #Label(root, text="select area",background="purple2").grid(row=0,column=0)

    popupMenu.grid(row=1,column=1)

    tkvar.set('Select area')
```

CHAPTER 6

TESTING

Testing is an important phase in the development life cycle of the product. This was a phase where the error remaining from all the phases was detected. Hence testing performs a very critical role for quality assurance and ensuring the reliability of the software. During the testing, the program to be tested was executed with a set of test cases and the output of the program for the test errors was evaluated to determine whether the program is performing as expected. Error were found and corrected by using the following testing steps and correction was recorded for future references. Thus, a series of testing was performed on the system before it was ready for implementation.

- **Test Environment**

A testing environment is a setup of software and hardware on which the testing team is going to perform the testing of the newly built software product. This setup consists of the physical setup which includes hardware and logical setup that includes Server operating system, client operating system, database Server, front end running environment, browser or any other software components required to run this software product.

This testing setup is to be built on both server and client. The software was the tested on the following platforms:

- Spyder/ Jupyter Integrated development Environment (IDE)
- Windows Operating System (OS)

- **Test Cases**

A test case is a document which has a set of test data, preconditions, expected result and post conditions for a particular test scenario in order to verify compliance against specific requirements.

- Features to be tested
- Items to be tested
- Purpose of testing
- Pass/Fail Criteria

6.1 Unit Testing

Unit testing is the testing of individual hardware or software units or groups of tested units. Using whitebox testing techniques, testers verify that the code does what it is intended to do at a very low natural level.

Unit testing is generally done within a class or a component. Unit testing focuses verification effort on the unit of software design (module). Using the unit test plans prepared in the design phase of the system development as a guide, important control paths are tested to uncover errors within the boundary of the modules.

Each unit in this project was thoroughly tested to check if it might fail in any possible situation. This testing was carried out at the completion of each unit. At the end of the unit testing phase, each unit was found to be working satisfactorily in regard to the expected output from the module. Table 5.1 shows the possible unit test cases.

Table 6.1 Unit testing

Serial no.	Test case	Input	Expected output	Actual output	Remarks
1	Test case for support and confidence	Valid support and valid confidence	Rules are generated	Rules are generated	PASS
2	Test case only for support	Valid support and no value for confidence	Please range the confidence value	Please range the confidence value	PASS
3	Test case only for confidence	Valid confidence or no value for support	Please range the support value	Please range the support value	PASS
4	Test case for no input support and confidence	No input for support and confidence	Please range the support and	Please range the support and	PASS

			confidence value	confidence value	
5	Test case for report accidents	Valid values for fields in the form	Data added to data.csv file	Data added to data.csv file	PASS
6	Test case for report accidents	No input for fields	Enter data	Enter data	PASS
7	Test case for graphs	Select graph to be generated	Graph will be displayed	Graph will be displayed	PASS
8	Test case for risk estimation	Select area for risk estimation	Risk will be estimated	Risk will be estimated	PASS

6.2 Integration Testing

Integration testing is the testing in which software components, hardware components or both are combined and tested to evaluate the interaction between them.

Using both black and white testing techniques, the tester verifies that units work together when they are integrated into a larger code base.

Data can be lost across an interface one module can have an adverse effect on the others sub functions, when combined it may not produce the desired major function. Also the global data structures can present problems. Integration testing is a symmetric technique for constructing the program structure while at the same time conducting tests to uncover errors associated with the interface. Table 5.2 shows the test cases for Integration testing .The related modules are combined and tested.

Table 6.2 Test cases for Integration testing

Serial no.	Test case	Input	Excepted output	Actual output	Remarks
1	Test case for Rule module	Valid support and valid confidence	Rules are generated	Rules are generated	PASS
2	Test case for Report accident module	Valid values for fields in the form	Data added to data.csv file	Data added to data.csv file	PASS
3	Test case for graph module	Select graph to be generated	Graph will be displayed	Graph will be displayed	PASS
4	Test case for risk estimation module	Select area for risk estimation	Risk will be estimated	Risk will be estimated	PASS

6.3 System Testing

System testing is the testing conducted on a complete ,integrated system to evaluate the system compliance with its specified requirements .System testing involves putting the new program in many different environments to ensure that the program work in typical customer environments with various versions and types of operating systems and/or applications.

Table 6.3 System testing

Serial no.	Test case	Input	Excepted output	Actual output	Remarks
1	Checking rules and graph and estimating risk	Enter appropriate parameters and selection	Rules will be generated, analysis will be displayed and risk will be estimated	Rules will be generated, analysis will be displayed and risk will be estimated	PASS

System testing is actually a series of different tests whose primary purpose is to fully exercise the computer-based system. Although each test has a different purpose, the main purpose is to verify that all the system elements have been properly integrated and perform the allocated functions. Table 5.3 represents System testing for this project. All the modules rule mining, risk estimation, analytics and the whole system was tested to check if it met the desired requirements

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENTS

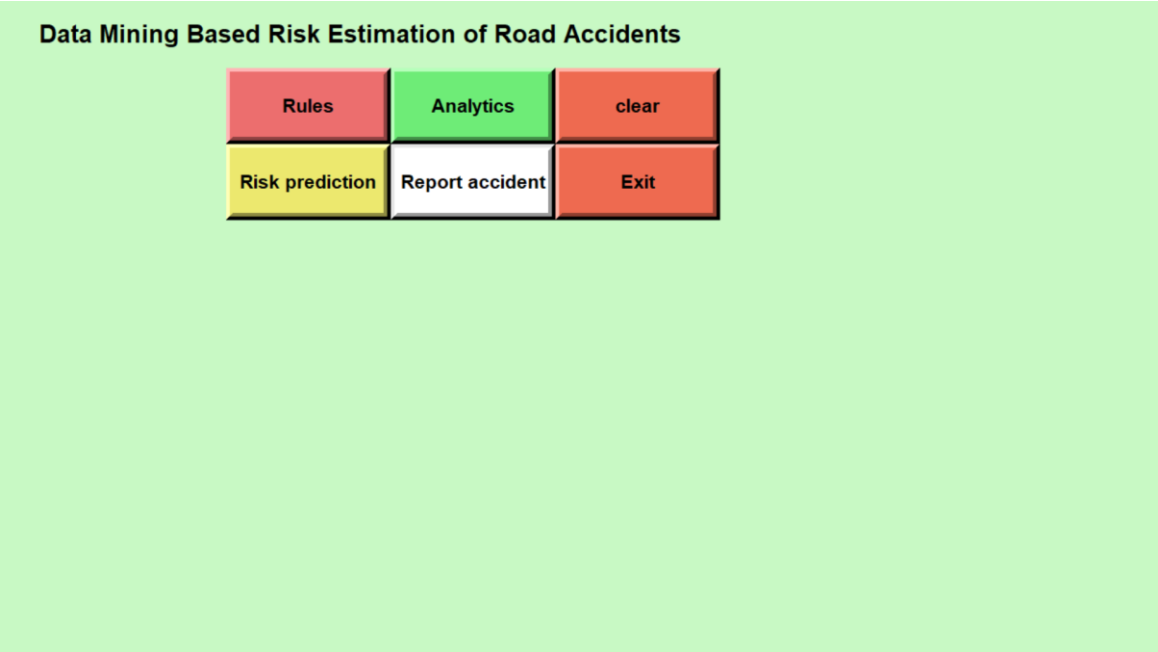
Conclusion

In this report, we have discussed that by using data mining techniques we see how cluster analysis helps in determining the accident prone zones in Bengaluru by estimating the risk as low and high based on various factors. We apply rule mining apriori algorithm to determine various factors that occur together and are mainly responsible for causing accidents. This can help in minimizing the road accidents if important measures are taken from the given insights about accidents to reduce them and to provide timely access to the needy.

Future Enhancement

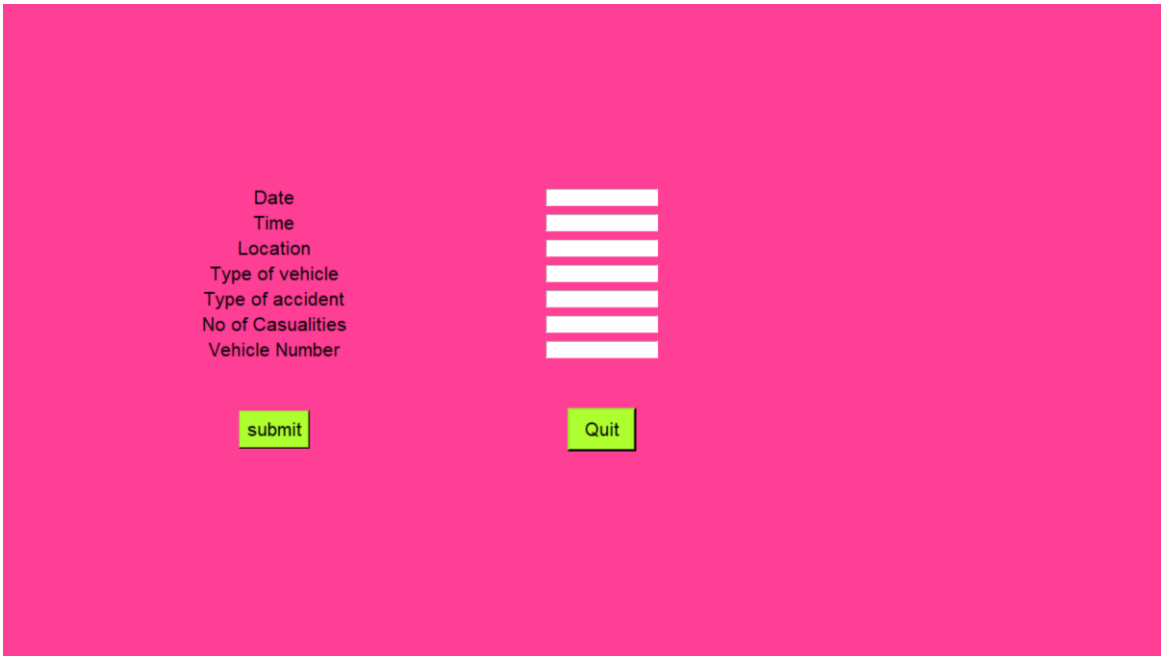
In the future one can try to improve the efficiency of the model and try to take preventive measures to reduce the accidents. One can develop an automatic emergency alarm response system to alert the users. One can also try to expand the model to data mining of other regions. One can try to provide a web application based solution and implementation.

SNAPSHOTS

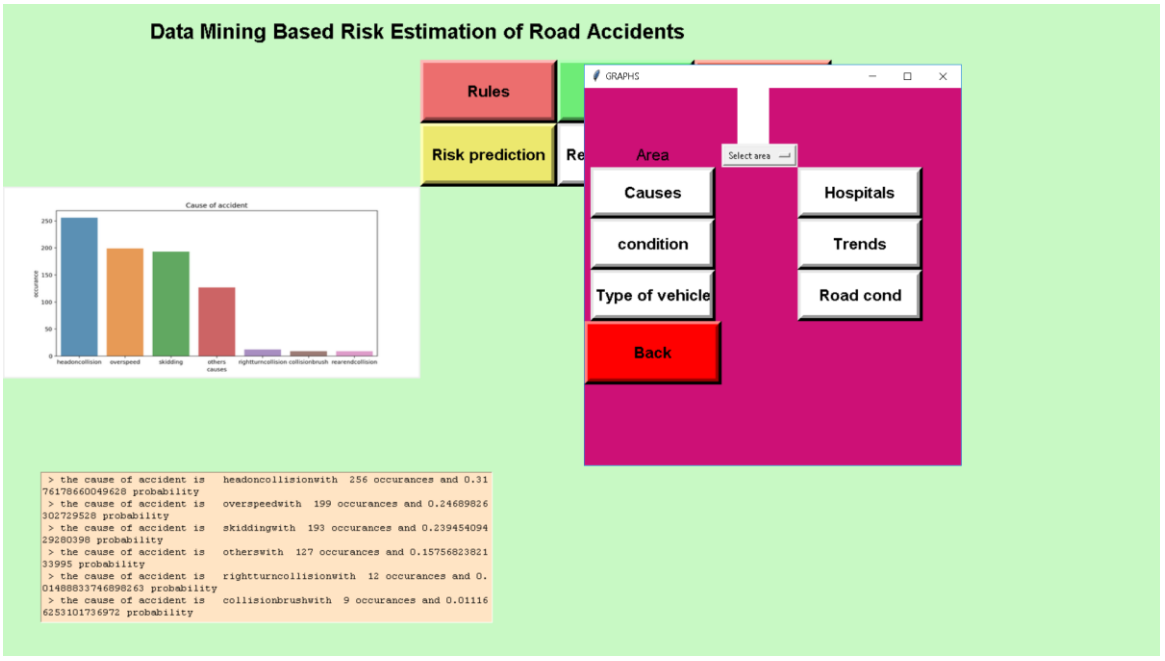


Snapshot 1:Homepage

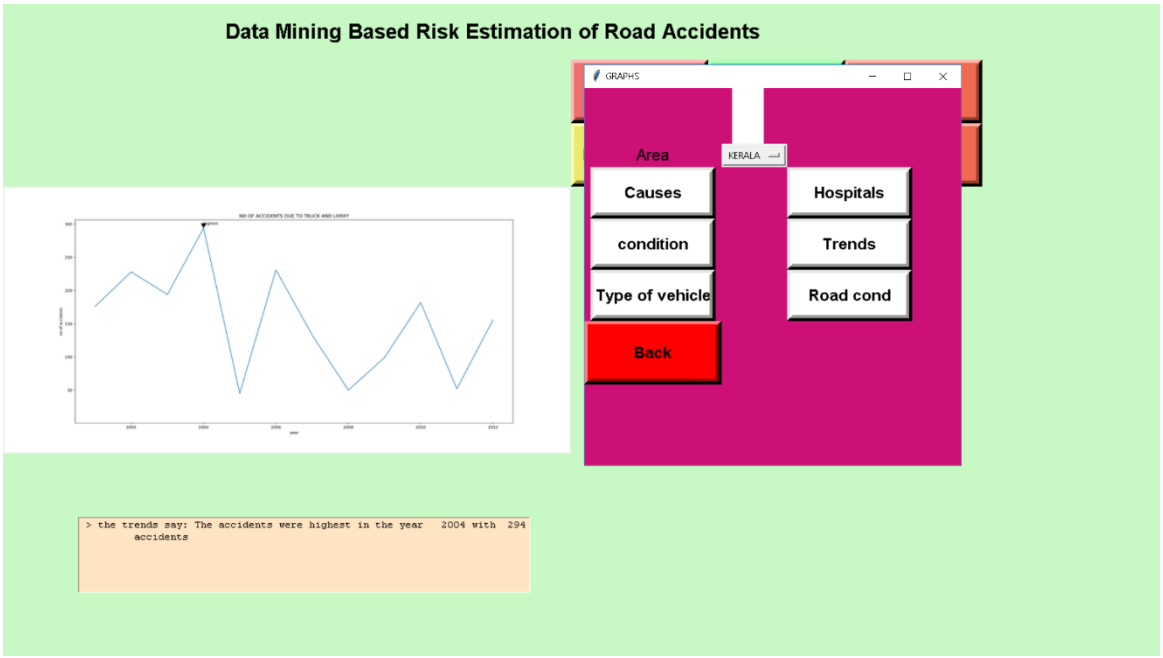
Snapshot 2:Rule mining



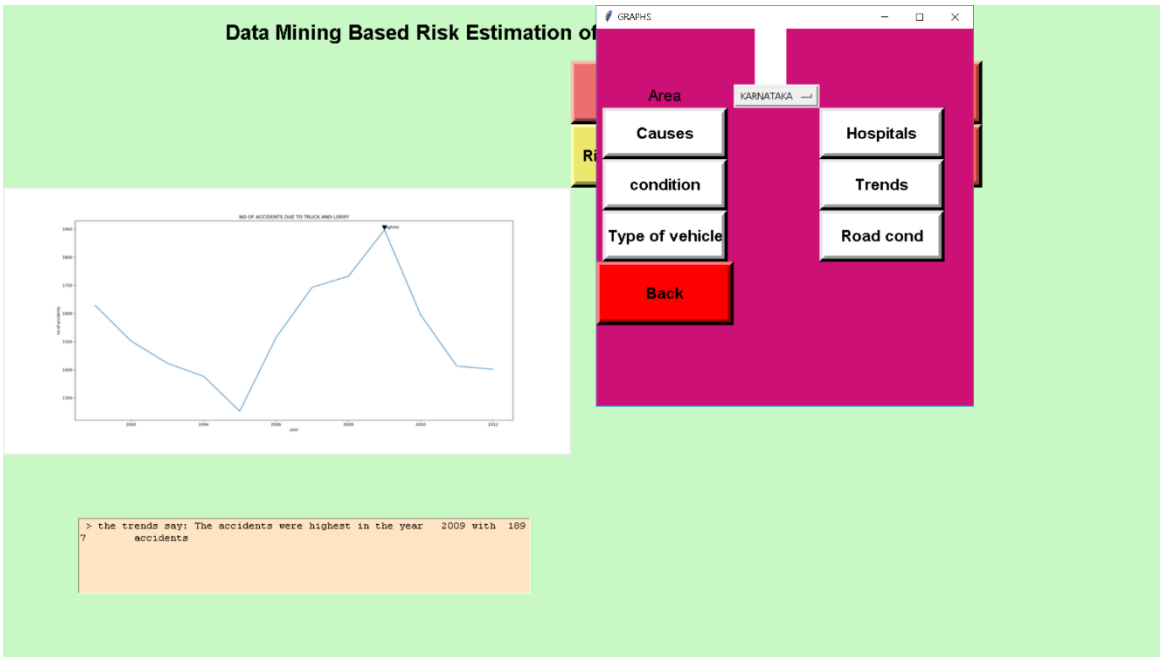
Snapshot 3: Reporting accidents



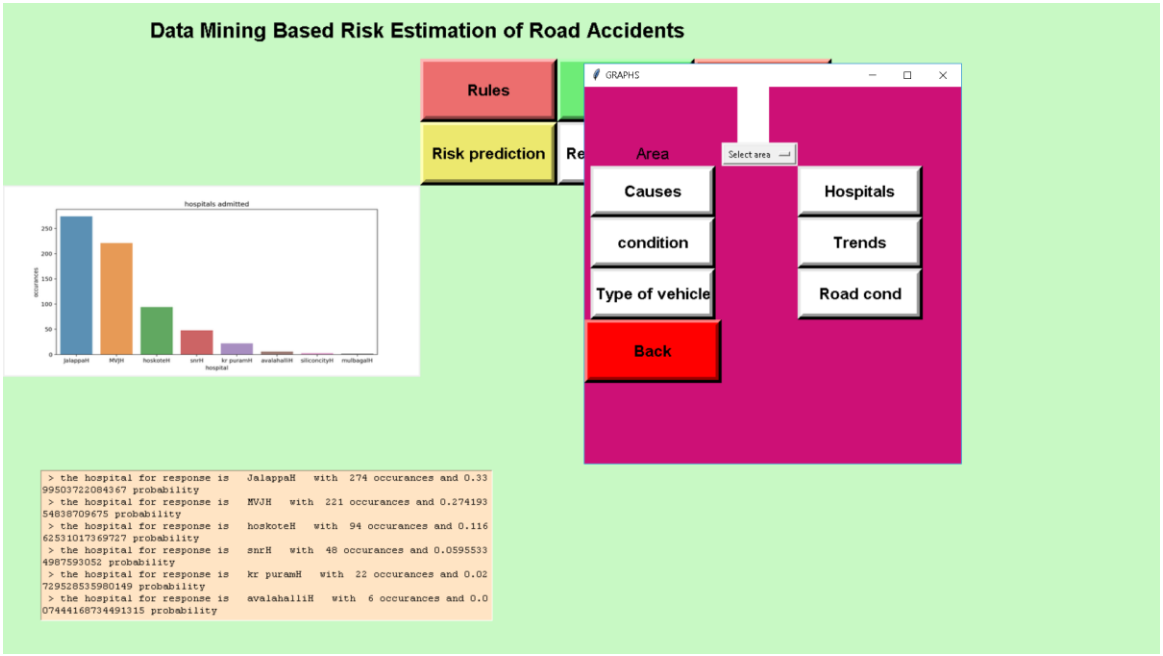
Snapshot 4:Generating causes graph



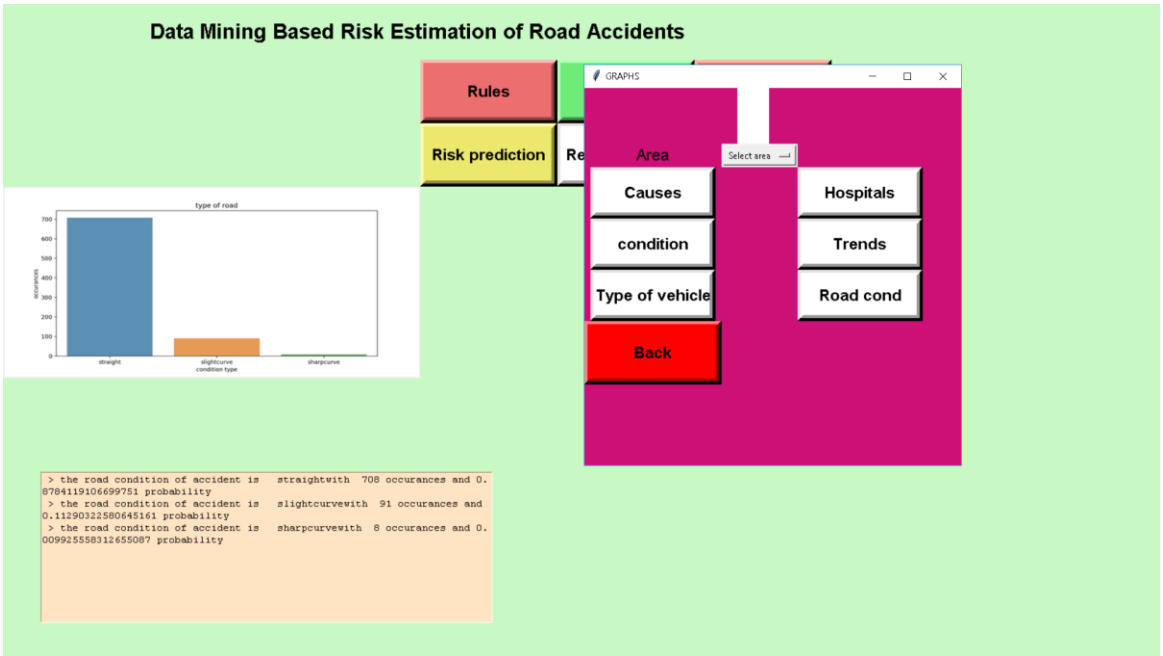
Snapshot 5: Generating Trends graph



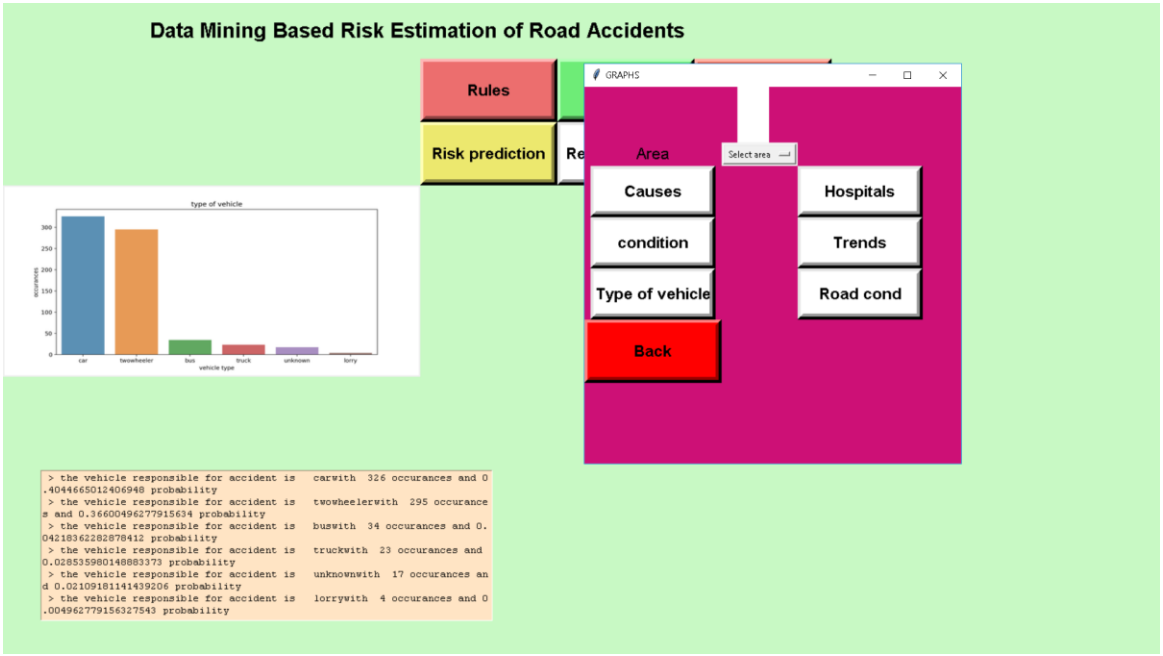
Snapshot 6: Generating Trends graph



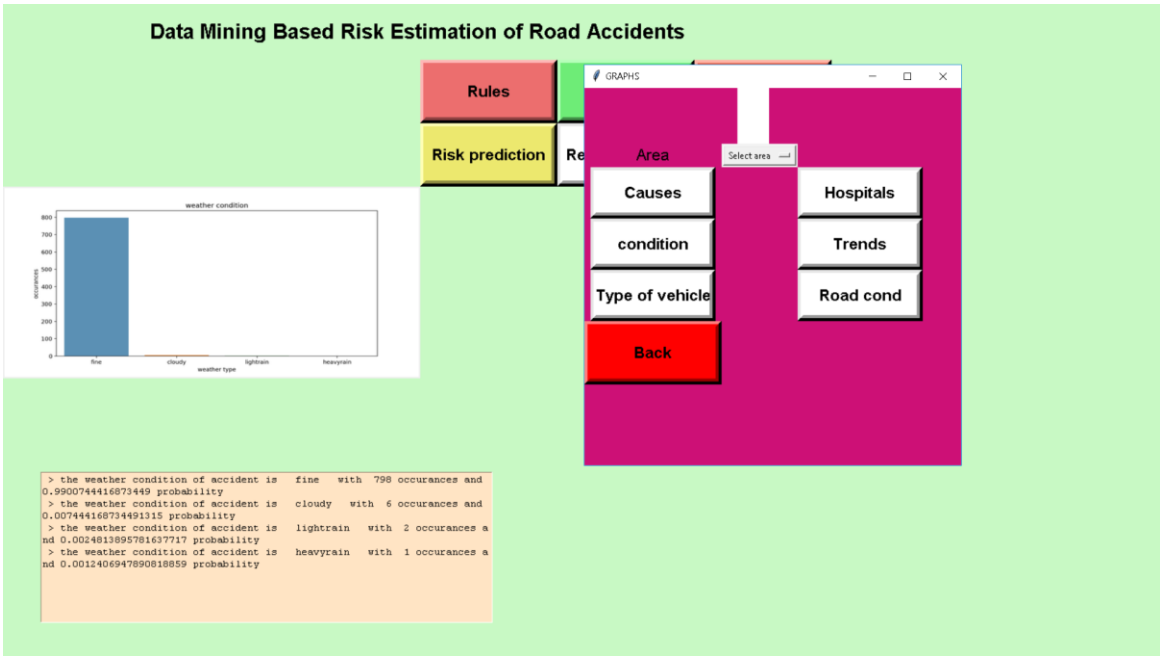
Snapshot 7: Generating graph of hospitals



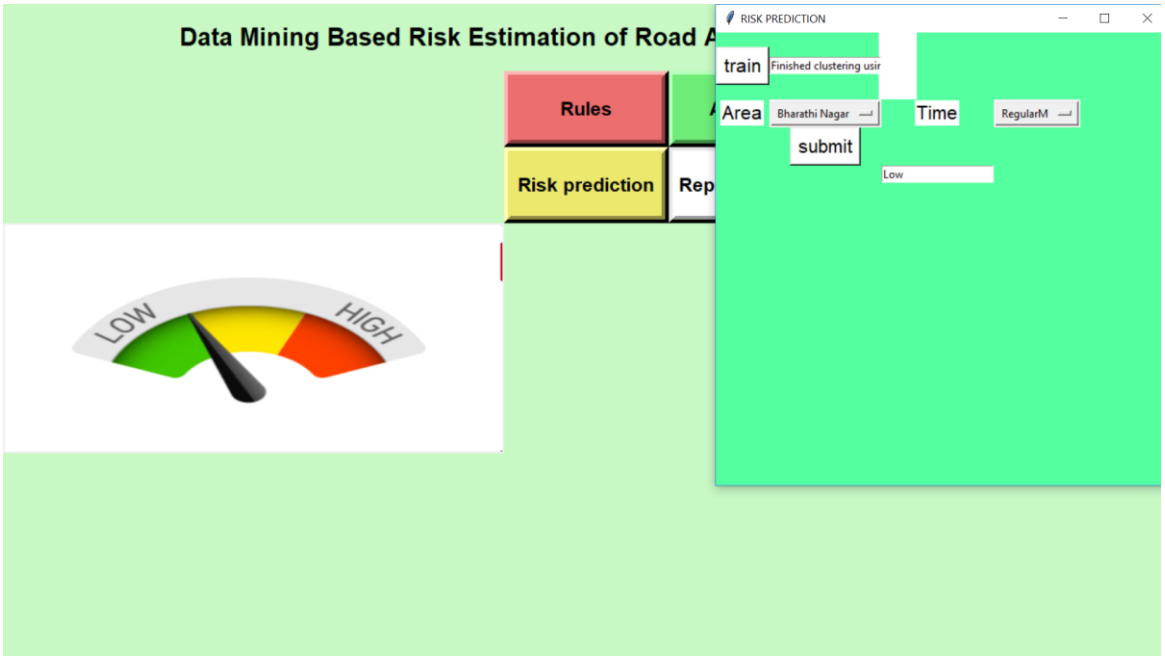
Snapshot 8: Generating graph of type of road condition



Snapshot 9: Generating graph of type of vehicle

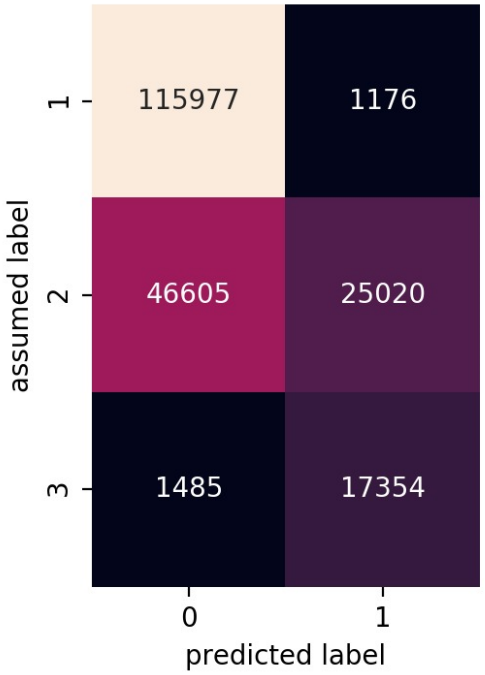


Snapshot 10: Generating graph of weather condition



Snapshot 11: Risk Prediction

ANALYSIS OF THE KMEANS CLUSTERING



Clusters	0	1
Labels		
1	115977	1176
2	46605	25020
3	1485	17354

CLUSTERS: 0-LOW 1-HIGH

LABELS: 0-LOW 1-MEDIUM 2-HIGH

ANNEXURE A

GLOSSARY

Data mining

An information extraction activity whose goal is to discover hidden facts contained in data. Using a combination of machine learning, statistical analysis, modeling techniques, data mining finds patterns and subtle relationships in data.

Clustering

Clustering algorithms find groups of items that are similar. It divides a data set so that records with similar content are in the same group, and groups are as different as possible from each other.

Association rule mining

An association algorithm creates rules that describe how often events have occurred together.

Support

The measure of how often the collection of items in an association occur together as a percentage of all the transactions.

Confidence

Confidence of rule "B given A" is a measure of how much more likely it is that B occurs when A has occurred. It is expressed as a percentage, with 100% meaning B always occurs if A has occurred.

Trend analysis

A series of measurements taken at consecutive points in time. Data mining products which handle time series incorporate time-related operators such as moving average.

Visualization

Visualization tools graphically display data to facilitate better understanding of its meaning.

ANNEXURE B

ACRONYMS

RTA	Road Traffic Accidents
RTO	Regional Transport Office
CSV	Comma Separated Values
UML	Unified Model Language
GUI	Graphical User Interface
IDE	Integrated Development Environment
WHO	World Health Organization

ANNEXURE C

LANGUAGE DESCRIPTION

Overview of Python

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently whereas other languages use punctuation, and it has fewer syntactic constructions than other languages.

- **Python is Interpreted** − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- **Python is Interactive** − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- **Python is Object-Oriented** − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- **Python is a Beginner's Language** − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

Python Features

Easy-to-learn − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.

Easy-to-read − Python code is more clearly defined and visible to the eyes.

Easy-to-maintain − Python's source code is fairly easy-to-maintain.

A broad standard library − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.

Interactive Mode − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.

Portable − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.

Extendable − You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.

Databases − Python provides interfaces to all major commercial databases.

GUI Programming − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.

Scalable − Python provides a better structure and support for large programs than shell scripting.

Tkinter Programming

Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.

Creating a GUI application using Tkinter is an easy task. All you need to do is perform the following steps −

- Import the *Tkinter* module.
- Create the GUI application main window.
- Add one or more of the above-mentioned widgets to the GUI application.
- Enter the main event loop to take action against each event triggered by the user.

BIBLIOGRAPHY

[1] Ameratunga S, Hijar M, Norton R. Road-traffic injuries: Confronting disparities to address a global-health problem. [Last cited on 2011 June 27];Lancet. 2006 367:1533–40.

[2] Nantulya VM, Reich MR. The neglected epidemic: Road traffic injuries in developing countries. [Last cited 2011 June 27];BMJ. 2002 324:1139–41. Available from: www.bmj.com/content/324/7346/1139.full . [PMC free article] [PubMed] [Google Scholar]

[3] Ayushi Jain, Garima Ahuja, Anuranjana, Deepti Mehrotra, Data mining approach to analyze the road accidents in india, proc of IEEE, 2016

[4] Global status report on road safety: supporting a decade of action, Geneva, World health organization, 2013.

[5] Open Government Data (OGD) platform india [online].available 2016: https://data.gov.in/catalogs/sector/Transport-9383

[6] Ms. Gagandeep Kaur, Harpreet Kaur, Prediction of the cause of accident and accident prone location on roads using data mining techniques, Proc of IEEE,2017

[7] Handbook of Research on Computational Science and Engineering: Theory and practice.

[8] a Makarova, Ksenia Shubenkova, Eduard Mukhametdinov, and Anton Pashkevich, Safety related problems of transport system and their solutions, proc of IEEE, 2018

[9] WHO, "Global status report on road safety 2015"

[10] Peden, World Health Organization. Ed. by Margie- 2004. World report on road traffic injury prevention. Geneva: World Health Organization.

[11] ADB-ASEAN Regional Safety Program, Country Report: CR09, Final Report, Thailand -(2004)

[12] https://www.tutorialspoint.com/python/python_overview.htm

[13] ADB-ASEAN Regional Safety Program, Country Report: CR09, Final Report, Thailand -(2004)

[14] http://morth.nic.in/showfile.asp?lid=2904

[15] Sachin Kumar and Durga Toshniwal, A novel framework to analyze road accident time series data, Springer Paper, 2016

[16] http://pubs.sciepub.com/acis/3/1/3/

[17] K Meshram and H.S. Goliya" Accident Analysis on National Highway-3 between Indore to Dhamnod" International Journal of Application or Innovation in Engineering & Management (IJAIEM) Volume 2, Issue 7, July 2013.

[18] Rakesh mehar and Pradeep kumar agarwal "A systematic approach for formulation of a road safety improvement program in India", ScienceDirect 2013

[19] E.S. Park et al," safety effects of wider edge lines on rural, two-lane highways", Accident Analysis and prevention, IJSRD, vol-48,317-325, 2012.

[20] Michael Williamson and Huaguo Zhou "Develop Calibration Factors For Crash Prediction Model For Rural Two-Lane Roadways InIllinois "Procedia Social and Behavioral Sciences43, ScienceDirect2012

[21] Amir h. Ghodset al." Effect of car/truck differential speed limits on two-lane highways safety operation using microscopic simulation", ScienceDirect 2012.

[22] https://www.mapsofindia.com/my-india/india/road-accidents