VISVESVARAYA TECHNOLOGICAL UNIVERSITY
Machhe, Belagavi– 590018

DISSERTATION

ON

"DATA MINING FRAMEWORK TO ANALYZE ROAD ACCIDENT DATA"

Submitted in the partial fulfillment of the requirement for the award of Degree

BACHELOR OF ENGINEERING
IN
"INFORMATION SCIENCE & ENGINEERING "

Submitted by

AISHWARYA SASEENDRAN(1CD13IS005)

Under the guidance of
Mrs. Bharani B.R.
Asst. Professor, Dept. of ISE
Citech, Bangalore.

2016 - 2017

Department Of Information Science and Engineering

CAMBRIDGE INSTITUTE OF TECHNOLOGY
BANGALORE – 560036

CAMBRIDGE INSTITUTE OF TECHNOLOGY
BANGALORE – 560036

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project work entitled "**DATA MINING FRAMEWORK TO ANALYZE ROAD ACCIDENT DATA**" is a bonafide work carried out by **Ms.AISHWARYA SASEENDRAN(1CD13IS005)** in partial fulfillment for the award of **Bachelor of Engineering** in **"Information Science and Engineering"** of the **Visvesvaraya Technological University, Belgavi**, during the year 2016–2017. It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the report. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the said Degree.

Mrs.Bharani B.R	**Dr. Satyanarayan Reddy K**	**Dr. Suresh L**
Asst. Professor	Head of the Dept.,	Principal
Dept of ISE	Dept of ISE	Citech

Place: Bangalore
Date:

External Viva

Examiner Name	**Signature**
1: _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _	_____
2: _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _	_____

DECLARATION

I, AISHWARYA SASEENDRAN, a student of VIII semester B.E, Information Science and Engineering, Cambridge Institute of Technology, hereby declare that the project titled "**DATA MINING FRAMEWORK TO ANALYZE ROAD ACCIDENT DATA**" has been carried out by me and submitted in partial fulfillment of the course requirements of VIII semester **Bachelor of Engineering** in **Information Science and Engineering** as prescribed by **Visvesvaraya Technological University, Belgavi**, during the academic year 2016-2017.

I also declare that, to the best of my knowledge and belief, the work reported here is not from part of any other report on the basis of which a degree or award was conferred on an earlier occasion on this by any other student.

Date:

Place: Bangalore

AISHWARYA SASEENDRAN

(1CD13IS005)

ACKNOWLEDGEMENT

I would like to place on record my deep sense of gratitude to **Shri D. K. Mohan, Chairman, Cambridge Group of Institutions, Bangalore** for providing excellent Infrastructure and Academic Environment at CITECH without which this work would not have been possible.

I am extremely thankful to **Dr.Suresh L, Principal, Cambridge Institute of Technology, Bangalore** for providing me Academic ambience and Laboratory facilities to in, and everlasting motivation to carry out this work and shaping our careers.

I express my sincere gratitude to **Dr.Satyanarayan Reddy K, HOD, Dept. Of Information Science And Engineering, Cambridge Institute of Technology, Bangalore** for his stimulating guidance, continuous encouragement and motivation throughout the course of present work.

I also wish to extend my thanks to **Mrs.Bharani B R, Assistant Professor , Dept. Of Information Science And Engineering, Cambridge Institute of Technology, Bangalore** for her critical, insightful comments, guidance and constructive suggestion and support to improve the quality of this seminar work.

I also wish to extend my thanks to our seminar coordinator **Mr. Vinayaka S P , Assistant Professor, Dept. Of Information Science And Engineering, Cambridge Institute of Technology, Bangalore** for his guidance and constructive suggestion and support to improve the quality of this seminar work.

I take this opportunity to thank all my friends, classmates who always stood by me in difficult situation also helped me in technical aspects and last but least I wish to express deepest sense of gratitude to my parents who were a constant source of encouragement and stood by me as a pillar of strength for completing this work successfully.

AISHWARYA SASEENDRAN

ABSTRACT

In the developed as well as developing countries, Infrastructure development is one of the major investment by the government, while safety of passengers on roads is of utmost importance. A road optimization during the construction or during maintenance phase, requires that the engineers analyze all the parameters that play a crucial role in ensuring safety for the passengers and preventing accidents. One of the key objectives in accident data analysis is to identify the main factors associated with road accidents. The data to be analyzed(both structured and unstructured) is collected from various sources and has several attributes.It is a challenge to gather all such relevant data, detect and analyze it together to generate decision trees that give insights on previous accidents. For this purpose,we propose to harness the power of Data Mining technologies like Hadoop.The analysis will be represented in the form of Decision tree which can be represented graphically.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 PROJECT PREFACE

Road accidents are uncertain and unpredictable incidents and their analysis requires the knowledge of the factors affecting them.

The major problem in the analysis of accident data is its Heterogenous nature.Thus, heterogeneity must be considered during analysis of the data, otherwise some relationship between the data may remain hidden.

Although, researchers used segmentation of the data to reduce this heterogeneity using some measures such as expert knowledge, but there is no guarantee that this will lead to an optimal segmentation which consists of homogeneous groups of road accidents.Therefore, cluster analysis can assist the segmentation of road accidents.

1.2 PROJECT DEFINITION

This is a research based data analysis project in which we try to analyze a large data set not capable of being analyzed by typical database or data analysis software like Excel.

To overcome this, we try to implement distributed processing using Hadoop and pipe the result with Apache Zepellin to analyze and visualize the data set and generate a decision tree.

1.3 PROJECT SCOPE

Traffic Engineers and Government agencies can:

- Identify the basic nature of accidents happening in the selected highway stretches
- Identify the root cause of accidents based on the collected data.
- Identify the features of the road causing the accident.
- Identify and Compare various road segments for optimization.
- Identify road intersection types and the frequency of accidents.
- Run instructional recommendation system.

1.4 TOOLS REQUIRED

1.4.1 HARDWARE REQUIREMENTS

Processor Type: Intel Core I5-6200u Cpu

Processor Speed: 2.30 Ghz

Hard Disk: 1 Tb

Ram: 8 Gb

1.4.2 SOFTWARE REQUIREMENTS

Operating System: Ubuntu

Front End: Apache Zeppelin

Back End: Scala-Spark

What is Hadoop?

Hadoop is an open source platform that provides excellent data management provision. It is a framework that supports the processing of large data sets in a distributed computing environment.

Hadoop makes it possible to run applications on systems with thousands of commodity hardware nodes, and to handle thousands of terabytes of data. Its distributed file system facilitates rapid data transfer rates among nodes and allows the system to continue operating in case of node failure.

This approach lowers the risk of catastrophic system failure and unexpected data loss, even if a significant number of nodes become inoperative. Consequently, Hadoop quickly emerged as a foundation for big data processing tasks, such as scientific analytics, business and sales planning and processing enormous volumes of sensor data, including from internet of things sensors.

The core of Apache Hadoop consists of a storage part, known as Hadoop distributed file system (HDFS), and a processing part which is a MapReduce programming model. Hadoop splits files into large blocks and distributes them across nodes in a cluster. It then transfers packaged code into nodes to process the data in parallel.

This approach takes advantage of data locality, where nodes manipulate the data they have access to. This allows the dataset to be processed faster and more efficiently than it would be in a more conventional supercomputer architecture that relies on a parallel file system where computation and data are distributed via high-speed networking.

Why Hadoop?

The Apache Hadoop software library is a framework that allows for the distributed processing of large data sets across clusters of computers using programming models. It is designed to scale-up from single server to thousands of machines, each machine offering local computation and storage.

Rather than rely on hardware to deliver high-availability, the library itself is designed a highly available service on top of a cluster of computers, each of which may be prone to failures.

Hadoop provides a cost-effective storage solution for business and facilitates businesses to easily access new data sources and tap into different types of data to produce value from that data.

It is a highly scalable storage platform. Unique storage method of Hadoop is based on a distributed file system that basically 'maps' data wherever it is located on a cluster. The tools for data processing are often on the same servers where the data is located, resulting in much faster data processing.

Hadoop is fault tolerance. When data is sent to an individual node, that data is also replicated to other nodes in the cluster, which means that in the event of failure, there is another copy available for use.

Hadoop MapReduce

MapReduce is a programming paradigm at the heart of Apache Hadoop for providing massive scalability across hundreds or thousands of Hadoop clusters on commodity hardware. The MapReduce model processes large unstructured data sets with distributed algorithm on Hadoop cluster.

MapReduce is a processing technique and a program model for distributed computing based on java. The MapReduce algorithm contains two important tasks, Map and Reduce. ,Map takes a set of data and converts it into another set of data, where individual elements are broken down into tuples (key/value pairs).

Secondly, reduce task, which takes the output from a map as an input and combines those data tuples into a smaller set of tuples, As the sequence of the name MapReduce implies, the reduce task is always performed after the map job.

The major advantage of MapReduce is that it is easy to scale data processing over multiple computing nodes. Under the MapReduce model, the data processing primitives are called mappers and reducers. Decomposing a data processing application into mappers and reducers is sometimes nontrivial. But, once we write an application in the MapReduce form, scaling the application to run over hundreds, thousands, or even tens of thousands of machines in a cluster is merely a configuration change. This simple scalability is what has attracted many programmers to use the MapReduce model.

Algorithm

- Generally, MapReduce paradigm is based on sending the computer to where the data resides.
- MapReduce program executes in three stages, namely map stage, shuffle stage, and reduce stage.
    - **Map stage**: The map or mapper's job is to process the input data. Generally, the input data is in the form of file or directory and is stored in the Hadoop file system (HDFS). The input file is passed to the mapper function line by line. The mapper processes the data and creates several small segments of data.

- **Reduce stage**: This stage is the combination of the Shuffle stage and the **Reduce** stage. The Reducer's job is to process the data that comes from the mapper. After processing, it produces a new set of outputs, which will be stored in the HDFS.

- During a MapReduce job, Hadoop sends the Map and Reduce tasks to the appropriate servers in the cluster.

- The framework manages all the details of data-passing such as issuing tasks, verifying task completion, and copying data around the cluster between the nodes.

- Most of the computing takes place on nodes with data on local disks that reduces the network traffic.

- After completion of the given tasks, the cluster collects and reduces the data to form an appropriate result, and sends it back to the Hadoop server.

Apache Zeppelin

Apache Zeppelin is a multi-purposed web-based notebook that enables interactive data analytics. You can make data driven, interactive and collaborative documents with SQL, SCALA and more to Hadoop and Spark.

Apache Spark integration

Especially, Apache Zeppelin provides built-in Apache Spark integration. You don't need to build a separate module, plug-in or library for it.

Apache Zeppelin with Spark integration provides

- Automatic SparkContext(sc) and SQLContext injection
- Runtime jar dependency loading from local filesystem or maven repository. Learn more about dependency loader.
- Canceling job and displaying its progress.

Data Visualization

Some basic charts are already included in Apache Zeppelin. Visualizations are not limited to SparkSQL query, any output from any language backend can be recognized and visualized.

Scala-Spark

Scala Programming is comparatively less complex unlike Java. A single complex line of code in Scala can replace 20-25 lines of complex java code making it a preferable choice for data processing on Apache Spark.

Apache Spark is a lightning-fast cluster computing technology designed for fast computation. It is based on Hadoop MapReduce and it extends the MapReduce model to efficiently use it for more types of computation. The main feature of Spark is its in-memory cluster computing that increases the processing speed of an application.

Ubuntu Over Windows:

1. **Security:** Due to the low usage ration an emphasis on security it has a lower chance of being infected.
2. **Centralized Software Repository:** A windows store like App in which you can download and install Apps without the hassle of opening the websites every time and adding new repositories for software which can be updated together from one location.
3. **Unix Environment:** Piping and Redirection combined with traditional UNIX utilities provide a better environment than windows.
4. **Command line:** UNIX makes it easy to write programs and to interact. Command line applications can easily be used in batch files or scripts, which is great for automated testing or builds.
5. **Resources:** Ubuntu consumes lesser resources and is somewhat faster when compared with Windows.

CHAPTER 2

LITERATURE SURVEY

Review of literature is important in any research work. Many researchers have carried out research work in the area of road accidents. Some of them have analyzed accident data in different ways. Some of them Identification of Black spot zone. Some of them have developed accident models for forecasting future accident trends. They have also proposed strategies for road safety.

In the present chapter literature review is carried out covering the different issues related to road accident and road safety.

Yannis T.H. (2014) was presented A Review of The Effect of Traffic and Weather Characteristics on Road Safety. 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 nonlinear 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.

K. Meshram and H.S. Goliya (2013) were presented an analysis of accidents on small portion NH-3 Indore to Dhamnod. The data for analysis is collected for the period of 2009 to September 2011. More accidents occurred in Manpur region by faulty road geometry. The trend of accidents occurring in urban portion (Indore) is more than 35 % to rate of total accidents in each year.This may due to high speeds and more vehicular traffic. In the present study area the frequency of fatal accidents are 2 in a week and 6 for minor accidents in a week. More number of accident observed in 6 p.m. to 8 p.m. duration because in that time more buses are travels between villages and city. One fatal and five casualties are occurring per km per year in the study area. The volume of the trucks passing through study corridor is increasing by year. At Rajendra

Nagar from 2000 onwards the traffic is reduced due to the construction of by passes in that area.

Rakesh Mehar and Pradeep Kumar Agarwal(2013) were highlighted the deficiencies in the present state of the art and also presents some basic concepts so that systematic approach for formulation of a road safety improvement program in India can be developed. The study presents basic concepts to develop an accident record system, for ranking of Safety hazardous locations, for identification of safety improvement measures and to determine priorities of safety measures. It is expected that this study will provide a systematic approach for development of road safety improvement program in India and thus pave the way for improving safety on Indian roads.

E.S.Park (2012) studies the safety effect of wider edge lines was examined by analyzing crash frequency data for road segments with and without wider edge lines. The data from three states, Kansas, Michigan, and Illinois, have been analyzed. Because of different nature of data from each state, a different statistical analysis approach was employed for each state: an empirical Bays, before-after analysis of Kansas data, an interrupted time series design and generalized linear segmented regression analysis of Michigan data, and a cross sectional analysis of Illinois data. Although it is well-known that causation is hard to establish based on observational studies, the results from three extensive statistical analyses all point to the same findings. The consistent findings lend support to the positive safety effects of wider edge lines installed on rural, two-lane highways. In conclusion, this study lends scientific support to the positive safety effects of wider edge lines installed on rural two-lane highways. Although the magnitudes of crash reductions were somewhat different from state to state, the results point in the same direction.

Amir H. Ghods et al. (2012) Differential speed strategies increased the number and rate of car-truck overtakes over the range of volumes considered in this analysis. This suggests a negative effect on safety resulting from differential speed strategy applied to two-lane rural highways. On a positive side DSL and MSL strategies have reduced the number of car-car overtakes at different volumes, hence increasing safety. This latter relationship suggests a calming effect of slower trucks on the speed of the traffic stream, which results in fewer interactions between cars. No significant effect was observed concerning differential speed control strategies and both average TTC and PTDO. The effect on TTC was due to volume; highest TTC for car-car and car-truck interactions at very low volumes, decreasing to a minimum in the range between 500 vph to 800 vph and increasing slightly thereafter. This indicator

suggests the highest head-on risk is experienced in the mid volume region. The average speed of traffic decreases in a nonlinear fashion with volume with differential speed strategies indicating a downward shift in this relationship.

Michael Williamson and Huaguo Zhou (2012) were the development of calibration factors for crash prediction models in the new Highway Safety Manual (HSM) for rural two-lane roadways in Illinois. The crash prediction modes (so called Safety Performance Functions (SPF)) in the HSM were developed using data from multiple states, therefore the models must be calibrated to account for local factors, such as weather, roadway conditions, and drivers' characteristics. In this study, two calibration factors were developed for two different SPFs to give a better prediction of crash frequencies on rural two lane roadways in Illinois. This study determined the SPF that best predicts the crashes was developed specifically for rural two-lane Two-way roadways in Illinois. It is recommended that local SPFs be developed and compared to the HSM SPF when evaluating the safety of a roadway.

R.R. Dinu, A. Veeraragavan (2011) was presented Random Parameter Models for Accident Prediction on Two-Lane Undivided Highways in India.Based on three years of accident history, from nearly 200 km of highway segments, is used to calibrate and validate the models. The results of the analysis suggest that the model coefficients for traffic volume, proportion of cars, motorized two-wheelers and trucks in traffic, and driveway density and horizontal and vertical curvatures are randomly distributed across locations.They have concluded with a discussion on modeling results and the limitations of the present study.

CHAPTER 3

SYSTEM ARCHITECTURE & MODULES

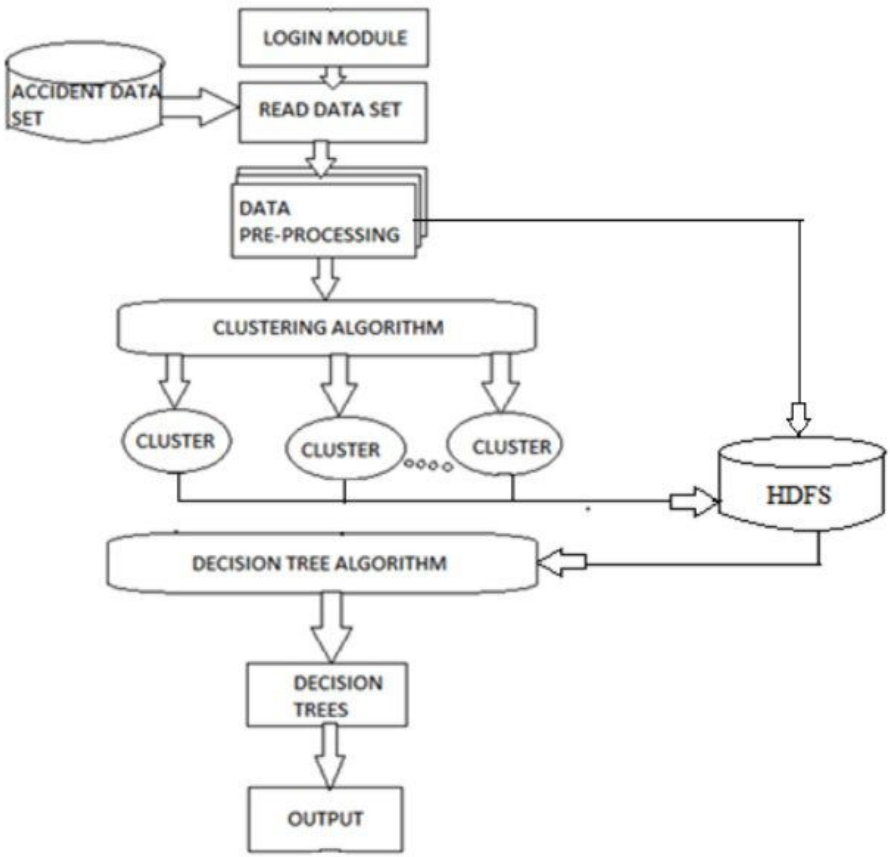


FIGURE 3.1 ARCHITECTURE OF PROPOSED SYSTEM

Basically, there are five modules that complete the project as listed below:

MODULE 1: LOGIN MODULE

- Username and Password authentication
- Starting Hadoop distributed file system(HDFS)

MODULE 2: DATA PRE-PROCESSING

- Converting Unstructured data to Structured data for Pre-Processing
- Data cleaning
  - Removing Missing Values
  - Removing Noisy data
  - Removing duplicate records
- Integration of data sets
- Fragmentation and replication for Hadoop

MODULE 3: CLUSTERING MODULE

- Setting Up Master and Slave nodes.
- Processing MapReduce jobs in a parallel environment.
- Fetching MapReduce Gain output to Zepellin for tree induction.

MODULE 4: ATTRIBUTE SELECTION AND TREE INDUCTION

- A Data Mining functionality for generating Decision Tree.
- Preparing Training Data set.
- Preparing Validation Data set.

MODULE 5: VISUALIZATION WITH APACHE ZEPELLIN

- Data Visualization.
- Other Statistical analysis

3.1 LOGIN MODULE

This phase of the project involves login part of the particular data set. In this, we can login using a particular username and password generated for the particular data.

If we want to login to a particular data set, we can login using that username and password that uses Hadoop technology.

- **Username and password authentication**

  The *su* command is used to enter the username and password to gain privileges into the Ubuntu account.

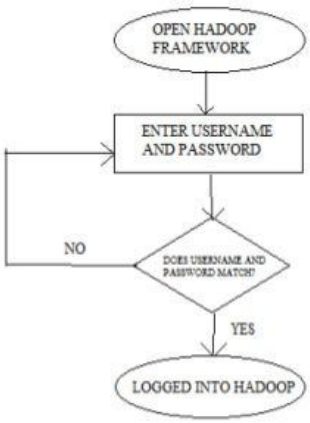


Figure 3.1.1 FLOWCHART OF LOGIN MODULE

- **Starting Hadoop distributed file system(HDFS)**

  ssh localhost command is used to enable secure shell. This is required to start all the daemon process on all the nodes from one machine.

  start -dfs.sh command,*start -yarn.sh* command are used to start the Hadoop distributed file system with the Namenode and YARN daemons.

  stop -dfs.sh command, *stop -yarn.sh* command are used to terminate the Hadoop distributed file systemand YARN daemons.

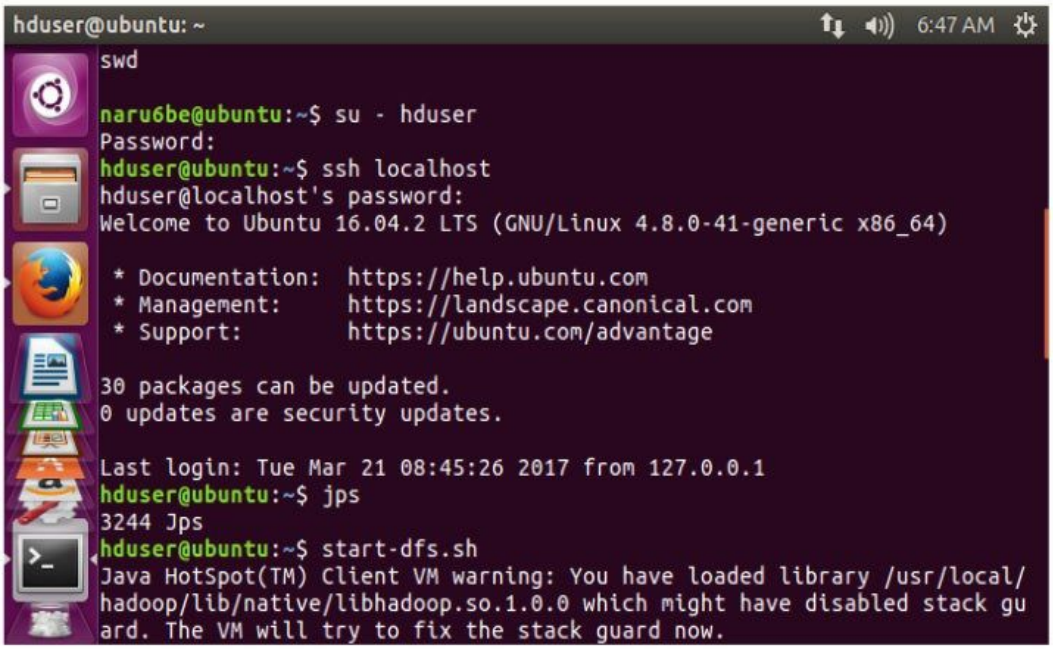


FIGURE 3.1.2: LOGIN MODULE(a) STARTING HDFS DAEMONS

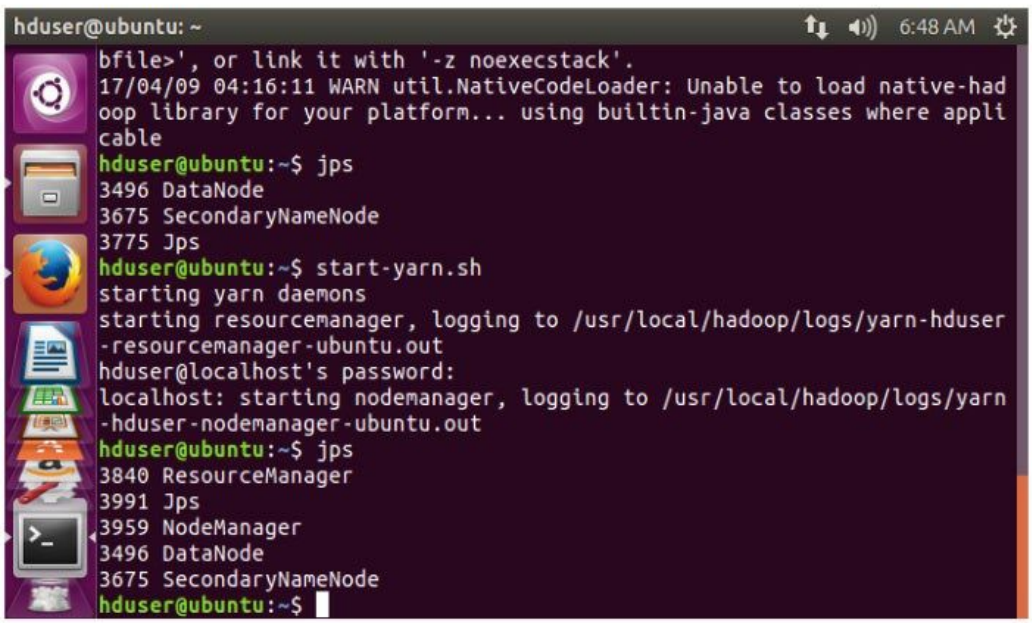


FIGURE 3.1.3: LOGIN MODULE(b) STARTING YARN DAEMONS

3.2 DATA PRE-PROCESSING

- Converting Unstructured data to Structured data for Pre-Processing

Three words to describe Big Data are:

- Volume
- Velocity
- Variety

The concept of developing processes to manage the increasing 'volumes' and 'velocity' of data almost seems conceivable.

Structured data is relatively simple and easy to use in process improvements as the data generally resides in databases in the form of columns and rows. It is grouped into relations or classes based upon shared characteristics. The data is generally allocated attributes (data descriptions) related to the classes within each group to help in ordering and logically grouping. Finally, it can be described by predefined formats (string or value) with predefined lengths of characters.

This makes structured data a good starting point for anyone looking for robust data to create information upon which to form meaningful insights. Structured data can be queried and analysed to sort, group, filter, count and sum in order to answer business questions or measure process capability. Whilst this doesn't account for the validity of the data it does enable relatively easy processing to verify and observe the data. Structured data forms a large part of the data used by many in process improvements, however this trend is quickly changing as the dominance of unstructured data increases.

Unstructured data is a generic term used to describe data that doesn't sit in databases and is a mixture of textual and non-textual data. Unstructured non-textual data generally relates to media such as images, video and audio files. As the volumes of this type of data increases through the use of smart phones and mobile Internet the need to analyse and understand it grows too. Slightly less unwieldy are unstructured textual data made up of media files (documents, spreadsheets, presentations), email messages and an array of other files generated and stored on corporate networks.

As unstructured data resides on corporate networks, within collaboration tools and in the cloud, it can be extremely difficult to interrogate or event locate. In order to, search the data, processes need to be in place to help tag and sort it. This step is key to allow for semantic searching against key words or contexts. Unstructured data is being utilised in a big way for social media companies wanting to understand their markets and customers in more depth. This presents the same opportunities to many of our businesses to help understand not only its customers better, but operations within.

The challenge for businesses is to develop processes to apply structure to the unstructured nature of the data. For example, determining the level of satisfaction of customers by analysing emails and social media may involve searching for words or phrases. Words and phrases may be grouped into positive, negative or neutral classifications.

At this stage, the unstructured data is transformed to structured data where the groups of words found based upon their classification are assigned a value. A positive word may equal 1, a negative -1 and a neutral 0. This unstructured data can now be stored and analysed as you would with structured data. Much more work is needed in this area to analyse the unstructured non-textual data and many of the big vendors are working on solutions.

- Data cleaning

Data preprocessing is one of the important tasks in data mining. Data preprocessing 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 preprocess the accident data in order to make it appropriate for the analysis.

- Integration of data sets

Data integration involves combining data residing in different sources and providing users with a unified view of them. This process becomes significant in a variety of situations, which include both commercial (such as when two similar companies need to merge their databases) and scientific domains. Data integration appears with increasing frequency as the volume and the need to share existing data explodes. It has become the focus of extensive theoretical work, and numerous open problems remain unsolved.

- Fragmentation and replication for Hadoop

In some operating system's file systems, a data file over a certain size is stored in several chunks or fragments rather than in a single contiguous sequence of bits in one place on the storage medium, a process that is called fragmentation. This allows small unused sections of storage (for example, where old data has been deleted) to be reused.

Replication is the process of making a replica (a copy) of something. A replication (noun) is a copy. The term is used in fields as varied as microbiology (cell replication), knitwear (replication of knitting patterns), and information distribution (CD-ROM replication).

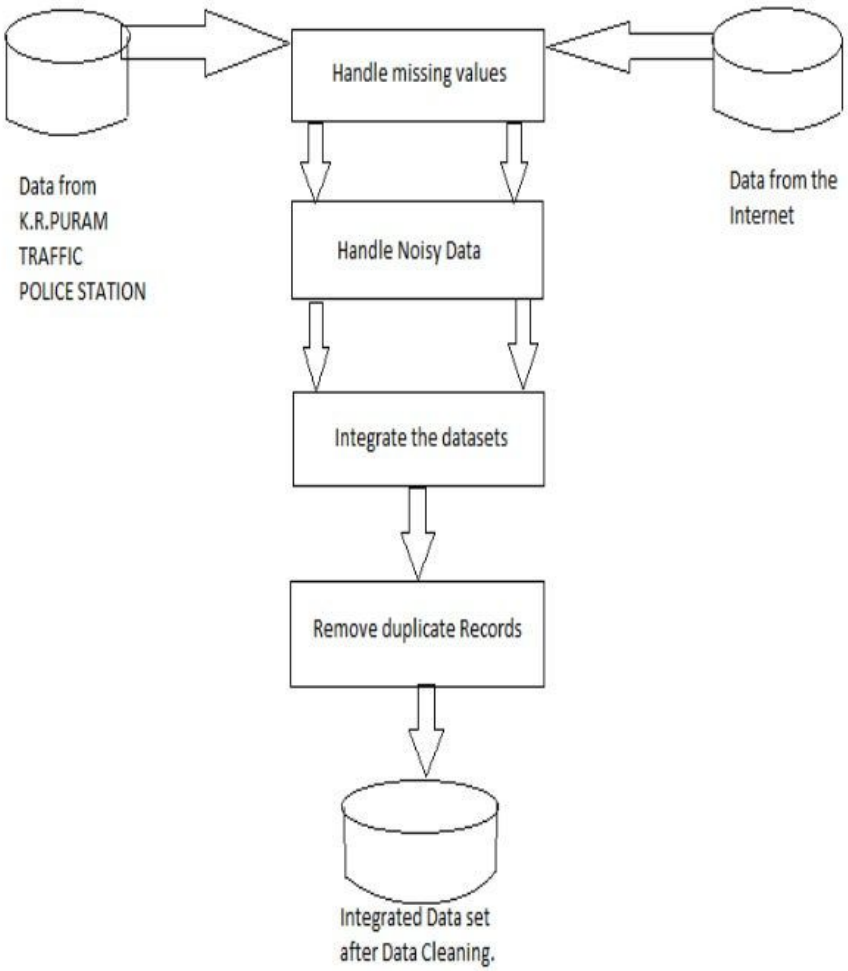


FIGURE 3.2.1: DATA PRE-PROCESSING

Pre-processing is a huge task in any data analysis project as the data input to the system is not in polished format and needs to be cleaned and prepared well using a series of steps before it can be used for the actual data mining process. The initial dataset is in a raw state which is also known as unstructured data format. It is meaningless and cannot be analyzed before structuring it using a metadata dictionary.

Some of the challenges faced in this project related to the data pre-processing are handling missing values and handling noisy values as a part of data cleaning, integrating the data from 3 different datasets using vertical joins and removing duplicate records for multiple index entries. The first phase of data cleaning includes removing the missing values. It is necessary to remove missing values from the data as the values not present can actually affect the results of the analysis in a negative way. So,to the file system we need to omit the records that have missing values.

After removing the missing values from the dataset, we have achieved one of the cleaning tasks. The next task is to remove the noisy values from the dataset. The noisy values in the dataset can be anything from -1 to infinity values garbage integers in place of expected ranges or negative values. According to that approach we handle the noisy values from some attributes by assigning the mean values of the attributes that belong to the same class as the attributes with the noisy values. This ensures that there is no over fitting of data. The next step in preprocessing is that of Integration. Integration in this case mainly has the issue of vertical join and also of removing duplicate records due to multiple indices.

Unstructured data is a generic label for describing data that is not contained in a database or some other type of data structure . If left unmanaged, the sheer volume of unstructured data that's generated each year within an enterprise can be costly in terms of storage. The information contained in unstructured data is not always easy to locate. It requires that data in both electronic and hard copy documents and other media be scanned so a search application can parse out concepts based on words used in specific contexts. This is called semantic search. The below figure shows the Unstructured Accident Data.

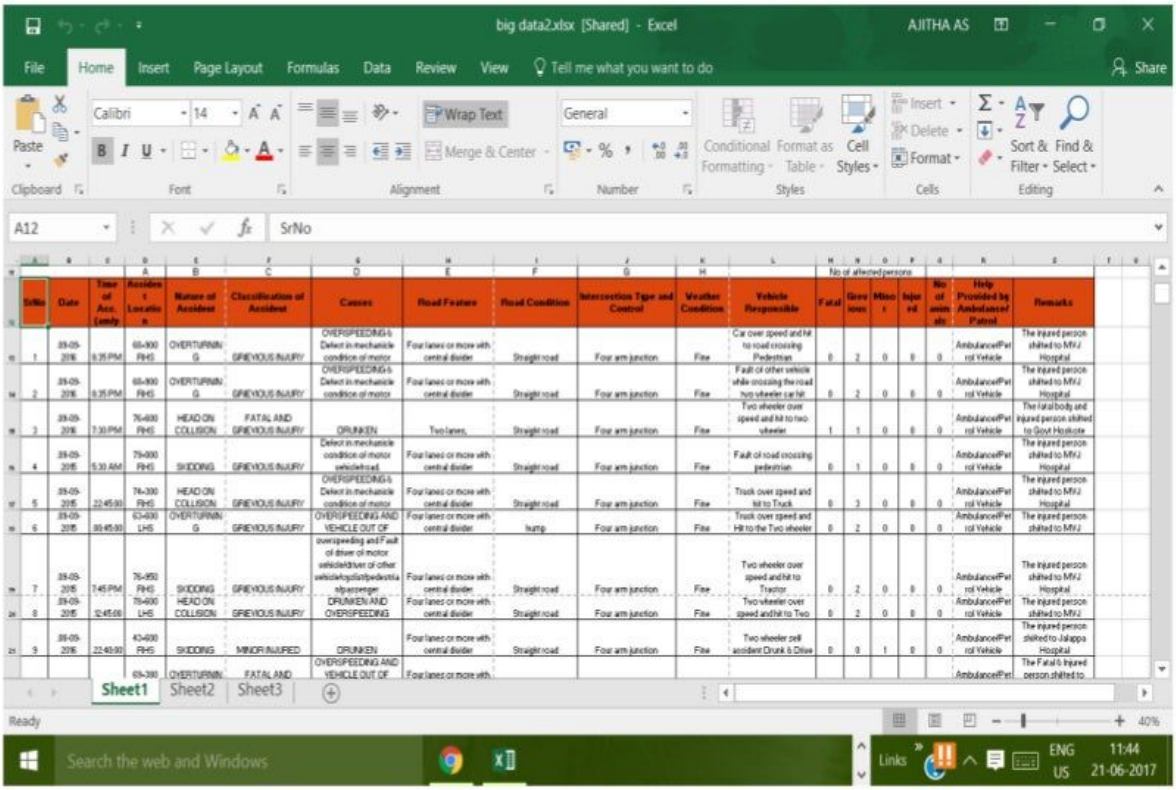


FIGURE 3.2.2: UNSTRUCTURED DATA SET

Structured data refers to any data that resides in a fixed field within a record or file. This includes data contained in relational databases and spreadsheets. Structured data first depends on creating a data model – a model of the types of business data that will be recorded and how they will be stored, processed and accessed. This includes defining what fields of data will be stored and how that data will be stored: data type and any restrictions on the data input.

Structured data has the advantage of being easily entered, stored, queried and analyzed. At one time, because of the high cost and performance limitations of storage, memory and processing, relational databases and spreadsheets using structured data were the only way to effectively manage data. The figure below shows structured Accident data.

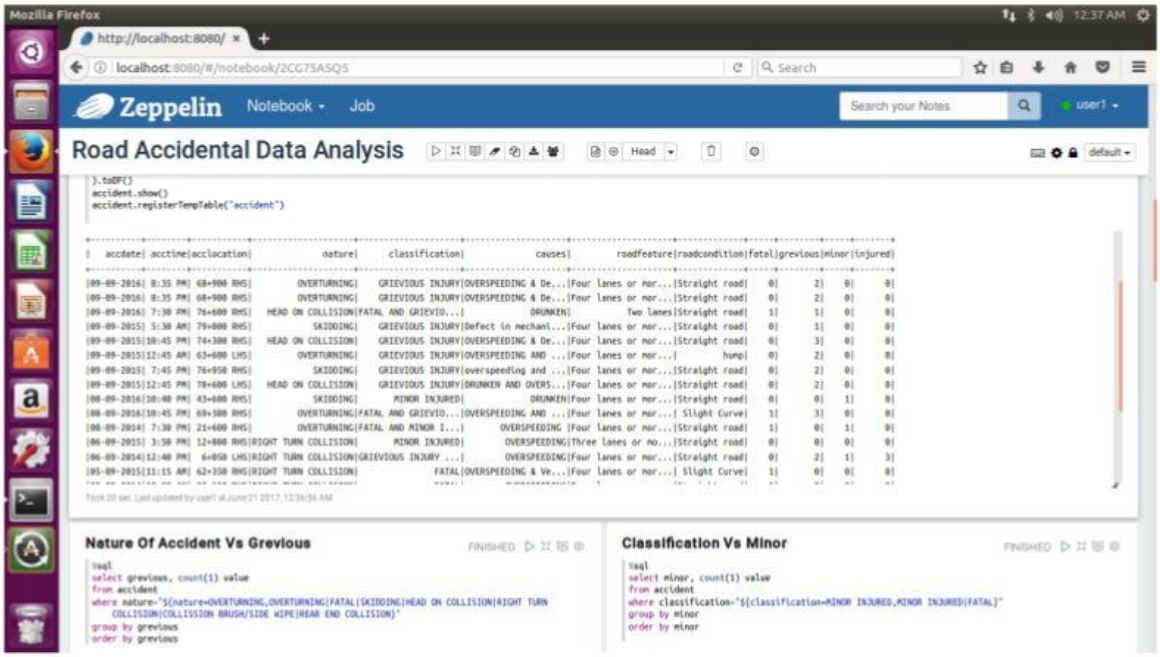


FIGURE 3.2.3: STRUCTURED DATA AFTER PRE-PROCESSING

3.3 CLUSTERING MODULE

A Hadoop cluster is a special type of computational cluster designed specifically for storing and analyzing huge amount of unstructured data in a distributed computing environment.

Clustering analysis is the task of grouping a set of objects in such a way that objects in the same group are more similar to each other than to those in other groups.

This Phase of the project involves the distributed processing of large dataset. It is not possible to load the dataset of 1 GB and above size into the memory of a simple computer which we use at desktop. Hence to process such large amount of data we use the Map Reduce paradigm of Hadoop framework.

The main focus in this phase is to analyze 1GB of data from the dataset and calculate their Information gain, entropy,split info for the attributes.

In a Hadoop cluster, a MapReduce program is known as a *job*. A job is run by being broken down into pieces, known as *tasks*. These tasks are scheduled to run on the nodes in the cluster where the data exists.

MapReduce jobs are executed by YARN in the Hadoop cluster. The YARN ResourceManager spawns a MapReduce ApplicationMaster container, which requests additional containers for mapper and reducer tasks. The ApplicationMaster communicates with the Namenode to determine where all of the data required for the job exists across the cluster. It attempts to schedule tasks on the cluster where the data is stored, rather than sending data across the network to complete a task. The YARN framework and the Hadoop Distributed File System (HDFS) typically exist on the same set of nodes, which enables the ResourceManager program to schedule tasks on nodes where the data is stored.

As the name MapReduce implies, the reduce task is always completed after the map task. A MapReduce job splits the input data set into independent chunks that are processed by map tasks, which run in parallel. These bits, known as *tuples*, are key/value pairs. The reduce task takes the output from the map task as input, and combines the tuples into a smaller set of tuples.

Each MapReduce ApplicationMaster monitors its spawned tasks. If a task fails to complete, the ApplicationMaster will reschedule that task on another node in the cluster.

This distribution of work enables map tasks and reduce tasks to run on smaller subsets of larger data sets, which ultimately provides maximum scalability. The MapReduce framework also maximizes parallelism by manipulating data stored across multiple clusters.

MapReduce

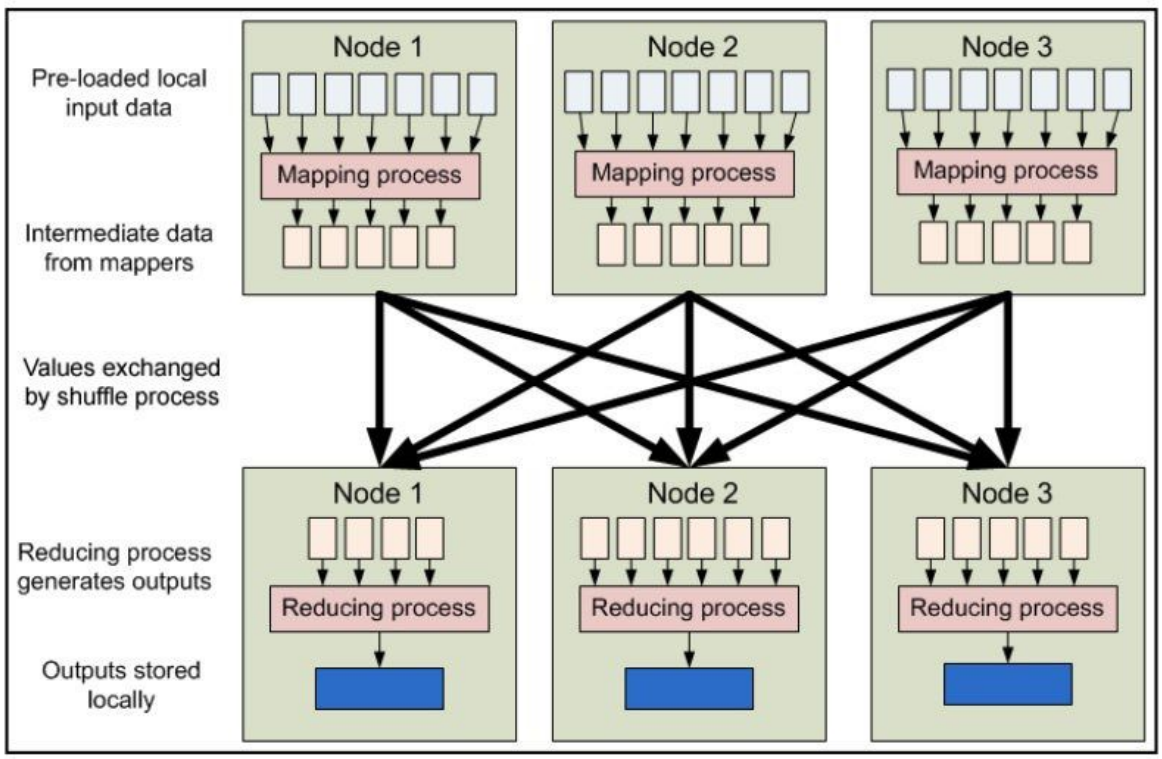


FIGURE 3.3.1: CLUSTERING USING MAPREDUCE

MapReduce is a programming model and an associated implementation for processing and generating Big Data sets with parallel distributed algorithm on a cluster.

A MapReduce program is composed of Map( ) procedure that performs filtering and sorting. The Map function in the Map Reduce program reads each row one by one and does the preliminary processing of counting and calculating. It will then write the output simultaneously into intermediate storage on HDFS or Local file system as per run configurations and a Reduce( ) method that performs a summary operation. The reduce function takes each row from the output of Map function and then aggregates them based on key-value pair, calculates the final gain and outputs it to the file system.

At the end,a simple file which lists the attributes with their respective information details is obtained.This analysis is the result of distributed processing and takes several hours on a small Hadoop cluster of 3-4 nodes.Increasing the number of data nodes in the Hadoop cluster significantly reduces the processing time.

This model is a specialization of *Split-Apply-Combine* strategy for data analysis. It is inspired by the Map and Reduce functions commonly used in Functional programming.

3.4 ATTRIBUTE SELECTION AND TREE INDUCTION

Classification is the process of building a model of classes from a set of records that contain class labels. Decision Tree Algorithm is to find out the way the attributes-vector behaves for a number of instances. Also on the bases of the training instances the classes for the newly generated instances are being found. Decision tree algorithm generates the rules for the prediction of the target variable. With the help of tree classification algorithm, the critical distribution of the data is easily understandable.

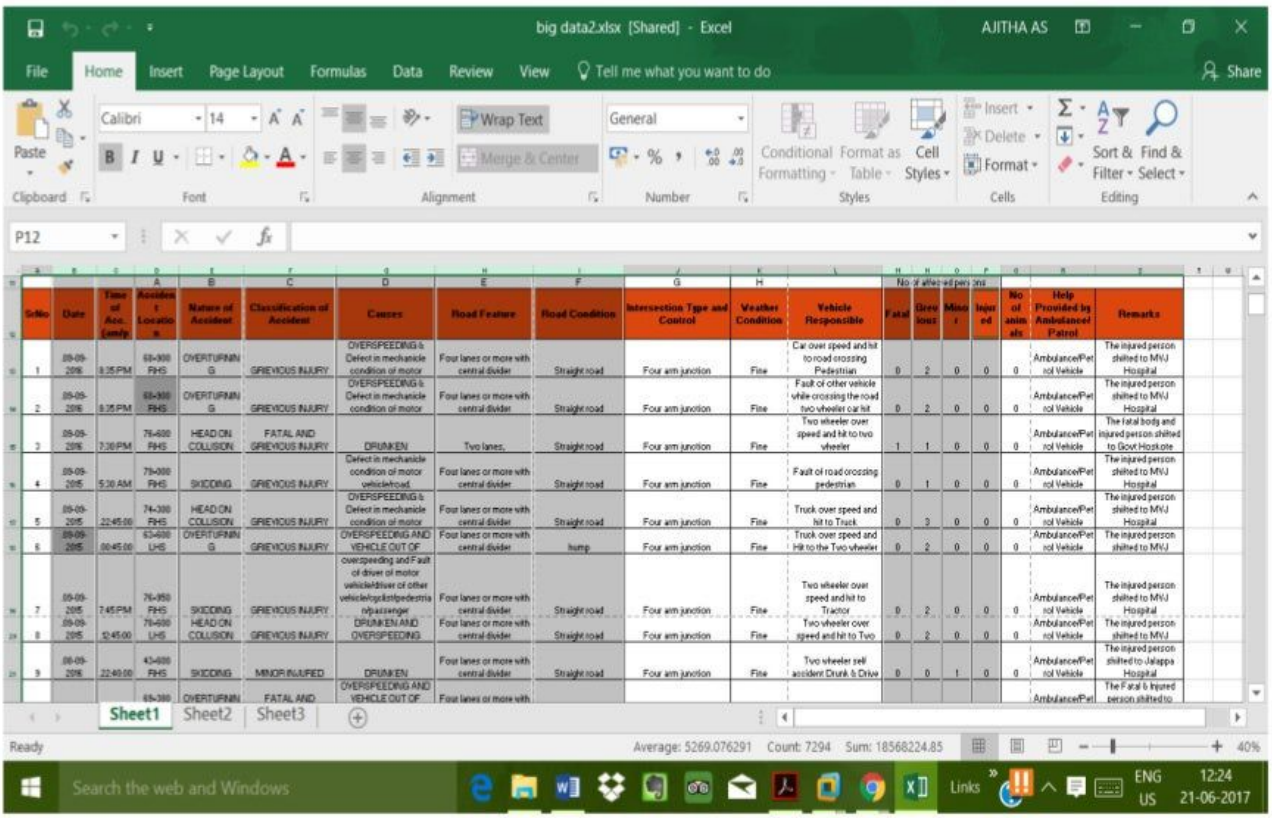


Figure 3.4.1: SELECTED ATTRIBUTES

Attribute selection is a term commonly used in data mining to describe the tools and techniques available for reducing inputs to a manageable size for processing and analysis. Feature selection implies not only cardinality reduction, which means imposing an arbitrary or

predefined cutoff on the number of attributes that can be considered when building a model, but also the choice of attributes, meaning that either the analyst or the modelling tool actively selects or discards attributes based on their usefulness for analysis.

The ability to apply feature selection is critical for effective analysis, because datasets frequently contain far more information than is needed to build the model. For example, a dataset might contain 500 columns that describe the characteristics of customers, but if the data in some of the columns is very sparse you would gain very little benefit from adding them to the model. If you keep the unneeded columns while building the model, more CPU and memory are required during the training process, and more storage space is required for the completed model. Even if resources are not an issue, unneeded columns should be typically removed because they might degrade the quality of discovered patterns, for the following reasons:

- Some columns are noisy or redundant. This noise makes it more difficult to discover meaningful patterns from the data.
- To discover quality patterns, most data mining algorithms require much larger training data set on high-dimensional data set. But the training data is very small in some data mining applications.

J48 classifier is used to increase the accuracy rate of the data mining procedure. The J48 algorithm is WEKA's implementation of the C4.5 decision tree learner. The data mining tool WEKA has been used as an API of MATLAB for generating the J-48 classifiers. J48 is an extension of ID3. The algorithm uses a greedy technique to induce decision trees for 20 classifications and uses reduced-error pruning. The additional features of J48 are accounting for missing values, decision trees pruning, continuous attribute value ranges, derivation of rules, etc.

In the WEKA data mining tool,J48 is an open source Java implementation of the C4.5 algorithm. The WEKA tool provides a number of options associated with tree pruning. In case of potential over fitting pruning can be used as a tool for précising. In other algorithms, the classification is performed recursively till every single leaf is pure, that is the classification of the data should be as perfect as possible.

This algorithm generates the rules from which particularidentity of that data is generated. The objective is progressively generalization of a decision tree until it gains equilibrium of flexibility and accuracy.

To run the experiments, the Data Mining tool WEKA was used.

WEKA is a collection of machine learning algorithms for Data Mining tasks. It contains tools for data preprocessing, classification, regression, clustering, association rules, and visualization.

WEKA has four different modes to work in.

1. Simple CLI: Provides a simple command-line interface that allows direct execution of WEKA commands.
2. Explorer: An environment for exploring data with WEKA.
3. Experimenter: An environment for performing experiments and conduction of statistical tests between learning schemes.
4. Knowledge Flow: Presents a "data-flow" inspired interface to WEKA. The user can select WEKA components from a tool bar, place them on a layout canvas and connect them together in order to form a "knowledge flow" for processing and analyzing data.

In Weka, you have three options of performing attribute selection from command line:

1. The native approach, using the attribute selection classes directly
2. Using a meta-classifier
3. The filter approaches

Native

Using the attribute selection classes directly outputs some additional useful information, like number of subsets evaluated/best merit (for subset evaluators), ranked output with merit per attribute (for ranking based setups).

Meta-classifier

Weka also offers a meta-classifier that takes a search algorithm and evaluator next to the base classifier. This makes the attribute selection process completely transparent and the base classifier receives only the reduced dataset.

Filter

In case you want to obtain the reduced/ranked data and not just output the selected/ranked attributes or using it internally in a classifier, you can use the filter approach.

J48 CLASSIFIER

J48 is an open source Java implementation of the C4.5 algorithm in the Weka data mining tool.J48 made a number of improvements to ID3algorithm. Some of these are:

- Handling both continuous and discrete attributes - In order to handle continuous attributes, j48 creates a threshold and then splits the list into those whose attribute value is above the threshold and those that are less than or equal to it.
- Handling training data with missing attribute values – J48 allows attribute values to be marked as "?" for missing. Missing attribute values are simply not used in gain and entropy calculations.
- Handling attributes with differing costs.
- Pruning trees after creation – J48 goes back through the tree once it's been created and attempts to remove branches that do not help by replacing them with leaf nodes.

In pseudo code, the general algorithm for building decision trees is:

1. Check for the above base cases.
2. For each attribute *a*, find the normalized information gain ratio from splitting on *a*.
3. Let *a_best* be the attribute with the highest normalized information gain.
4. Create a decision *node* that splits on *a_best*.
5. Recur on the sublists obtained by splitting on *a_best*, and add those nodes as children of *node*.

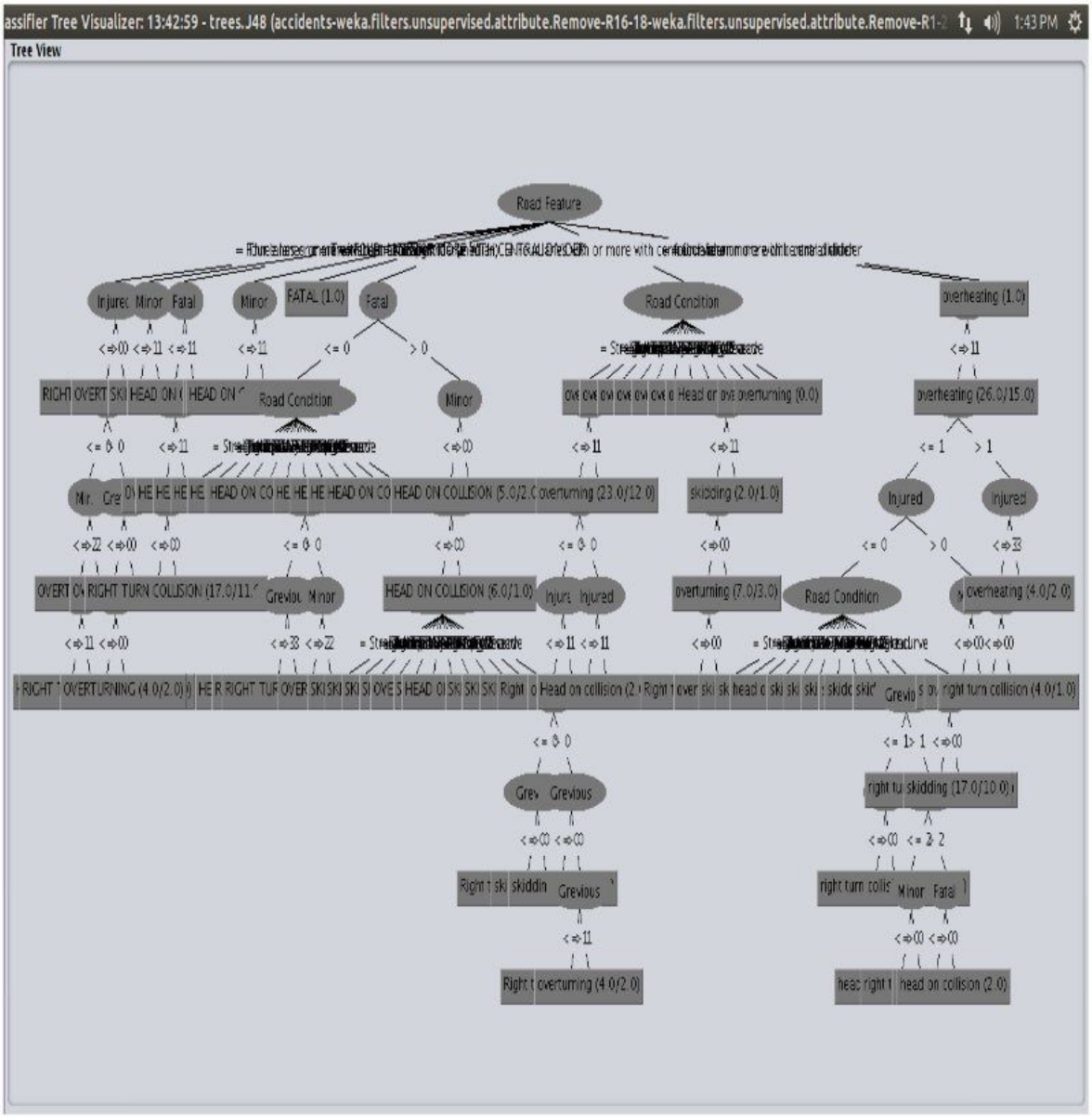


FIGURE 3.4.2: DECISION TREE INDUCTION

3.5 VISUALIZATION WITH APACHE ZEPELLIN

Nature of Accident Vs Grievous:

Nature Of Accident Vs Grevious FINISHED

```
%sql
select grevious, count(1) value
from accident
where nature="${nature=OVERTURNING,OVERTURNING|FATAL|SKIDDING|HEAD ON COLLISION|RIGHT TURN
    COLLISION|COLLISSION BRUSH/SIDE WIPE|REAR END COLLISION}"
group by grevious
order by grevious
```

nature: SKIDDING

grevious	value
0	48
1	13
2	11
3	4
4	2

FIGURE 3.5.1: TABLE SHOWING THE NUMBER OF GREVIOUS ACCIDENTS FOR GREVIOUS VALUES WHEN NATURE IS SKIDDING

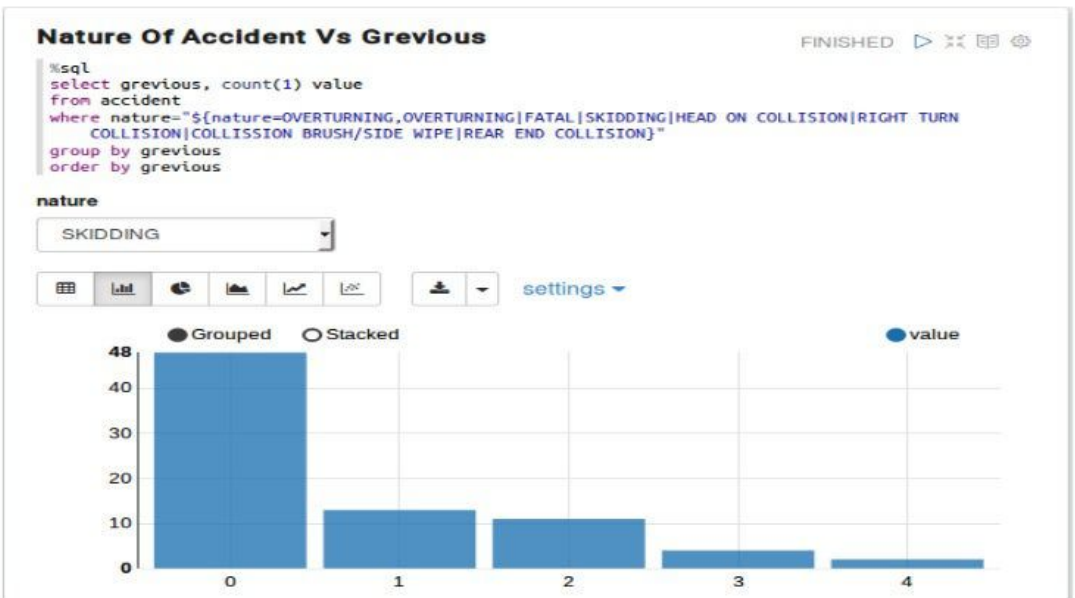


FIGURE 3.5.2: BAR GRAPH SHOWING THE NUMBER OF GREVIOUS ACCIDENTS ON Y-AXIS FOR GREVIOUS VALUES ON X-AXIS WHEN NATURE IS SKIDDING

Classification Vs Minor:

Classification Vs Minor FINISHED

```
%sql
select minor, count(1) value
from accident
where classification="${classification=MINOR INJURED,MINOR INJURED|FATAL}"
group by minor
order by minor
```

classification

MINOR INJURED

minor	value
0	2
1	54
2	22
3	6
6	1
7	1

FIGURE 3.5.3: TABLE SHOWING THE NUMBER OF MINOR ACCIDENTS FOR MINOR VALUES WHEN CLASSIFICATION IS MINOR INJURED

FIGURE 3.5.4: BAR GRAPH SHOWING THE NUMBER OF MINOR ACCIDENTS ON Y-AXIS FOR MINOR VALUES ON X-AXIS WHEN CLASSIFICATION IS MINOR INJURED

Causes Vs Classification Vs Minor:

Causes Vs Classification Vs Minor FINISHED

```
%sql
select minor, count(1) value
from accident
where causes="${causes=OVERSPEEDING,OVERSPEEDING|DRUNKEN}"
and classification="${classification=MINOR INJURED,MINOR INJURED|FATAL}"
group by minor
order by minor
```

causes: OVERSPEEDING classification: MINOR INJURED

minor	value
0	2
1	14
2	1
3	1
7	1

FIGURE 3.5.5: TABLE SHOWING NUMBER OF MINOR ACCIDENTS FOR MINOR VALUES WHEN CAUSES IS OVERSPEEDING & CLASSIFICATION IS MINOR INJURED

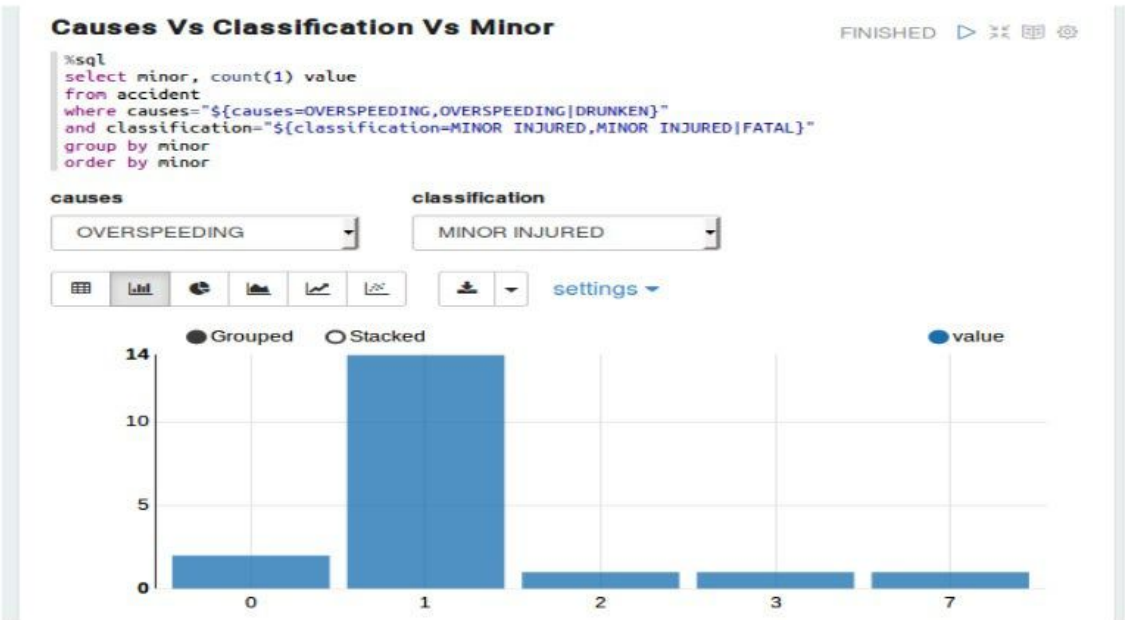
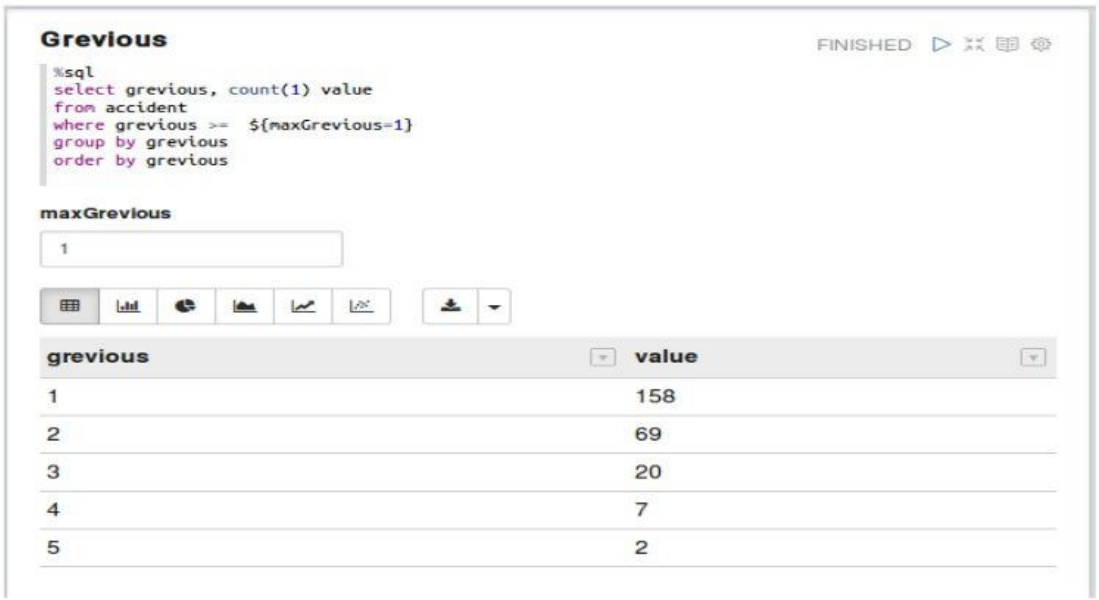


FIGURE 3.5.6: BAR GRAPH SHOWING THE NUMBER OF MINOR ACCIDENTS ON Y-AXIS FOR MINOR VALUES ON X-AXIS WHEN CAUSES IS OVERSPEEEDING CLASSIFICATION IS MINOR INJURED

Grevious:



grevious	value
1	158
2	69
3	20
4	7
5	2

FIGURE 3.5.7: TABLE SHOWING NUMBER OF GREVIOUS ACCIDENTS FOR GREVIOUS VALUES GREATER THAN 1

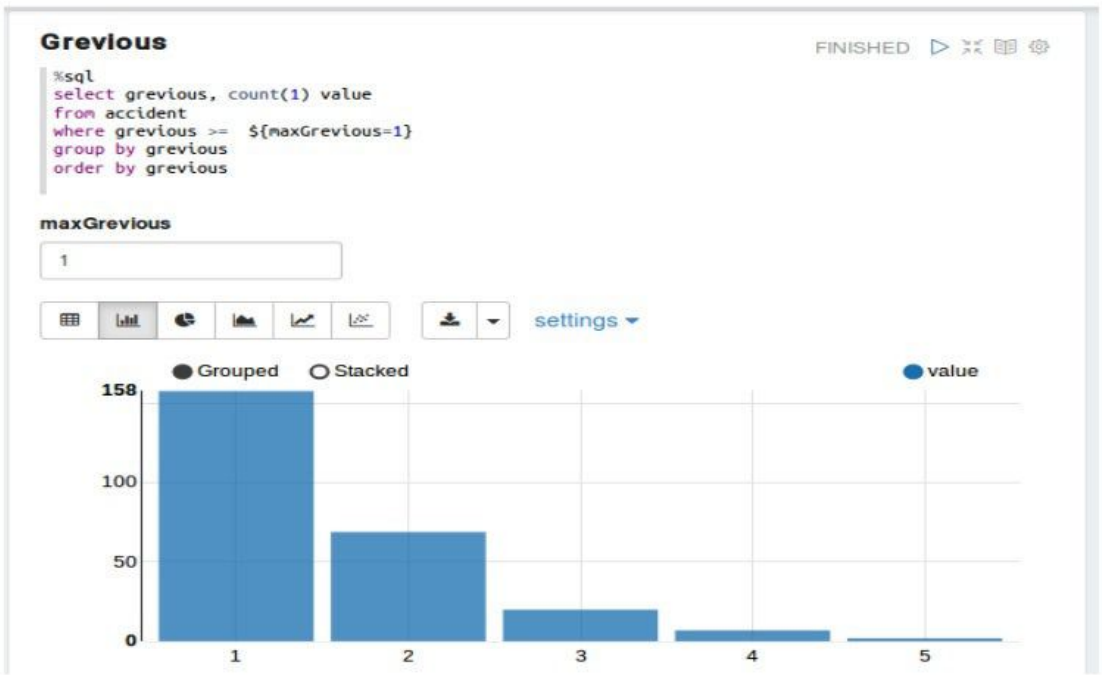
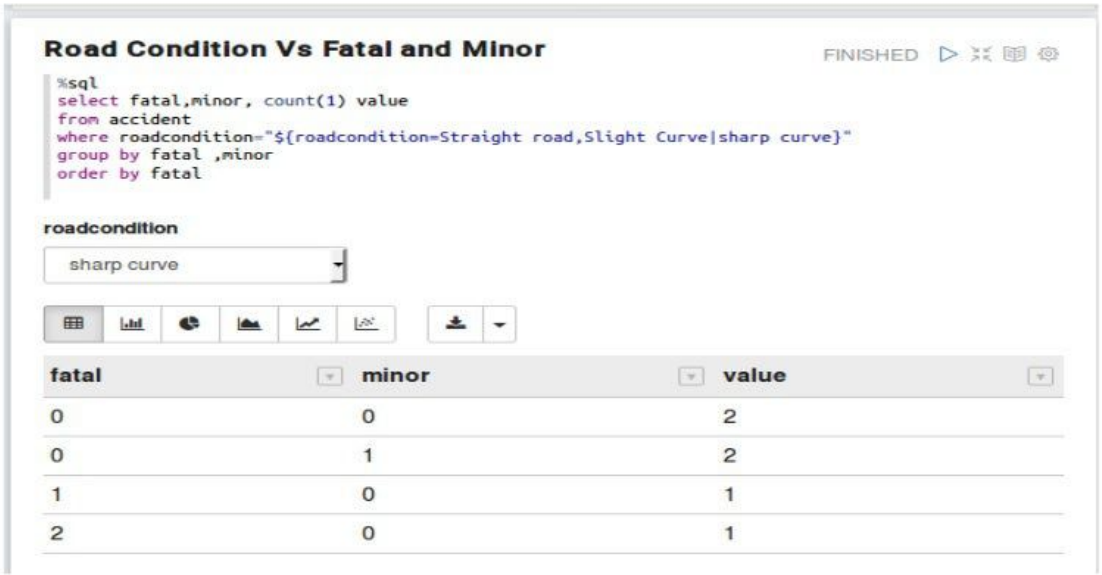


FIGURE 3.5.8: BAR GRAPH SHOWING NUMBER OF GREVIOUS ACCIDENTS FOR GREVIOUS VALUES GREATER THAN 1

Road Condition Vs Fatal and Minor:



fatal	minor	value
0	0	2
0	1	2
1	0	1
2	0	1

FIGURE 3.5.9: TABLE SHOWING NUMBER FATAL AND MINOR ACCIDENTS WHEN ROAD CONDITION IS SHARP CURVE

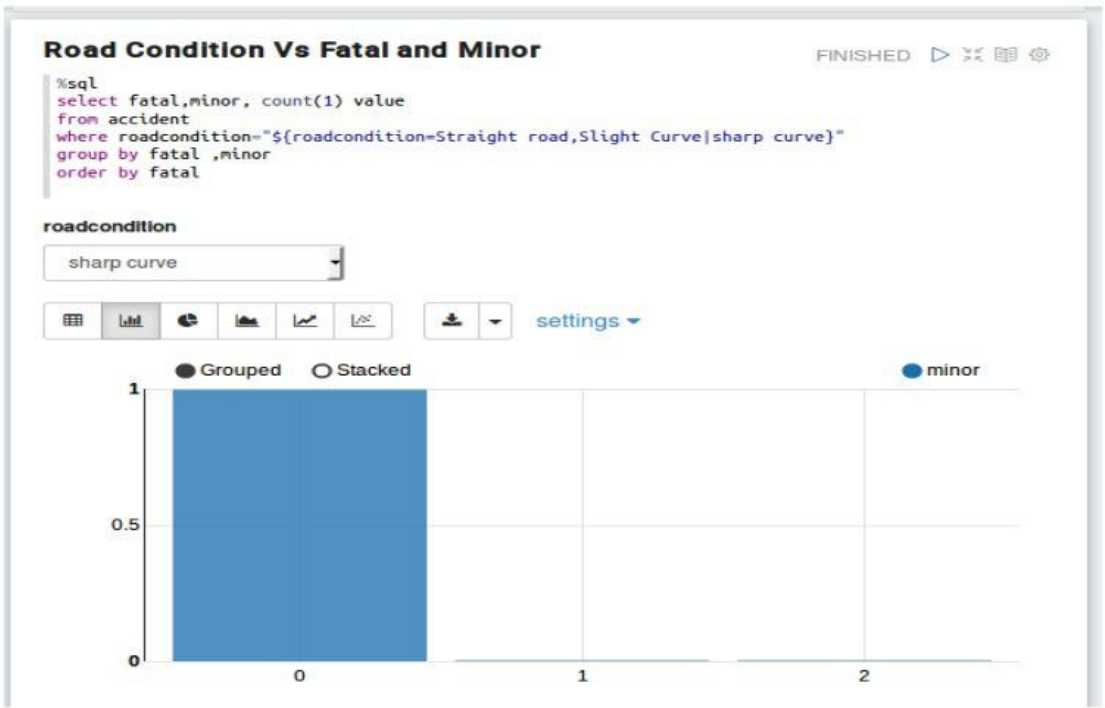


FIGURE 3.5.10: BAR GRAPH SHOWING NUMBER FATAL AND MINOR ACCIDENTS WHEN ROAD CONDITION IS SHARP CURVE

Road Feature Vs Grevious:

Road Feature Vs Grevious FINISHED

```
%sql
select grevious, count(1) value
from accident
where roadfeature="${roadfeature=FOUR  LANES OR MORE WITH CENTRAL DIVIDER,FOUR  LANES OR MORE
    WITH CENTRAL DIVIDER|hump}"
group by grevious
order by grevious
```

roadfeature

hump

grevious	value
0	3
1	2
2	1

FIGURE 3.5.11: TABLE SHOWING NUMBER GREVIOUS ACCIDENTS WHEN ROAD FEATURE IS HUMP

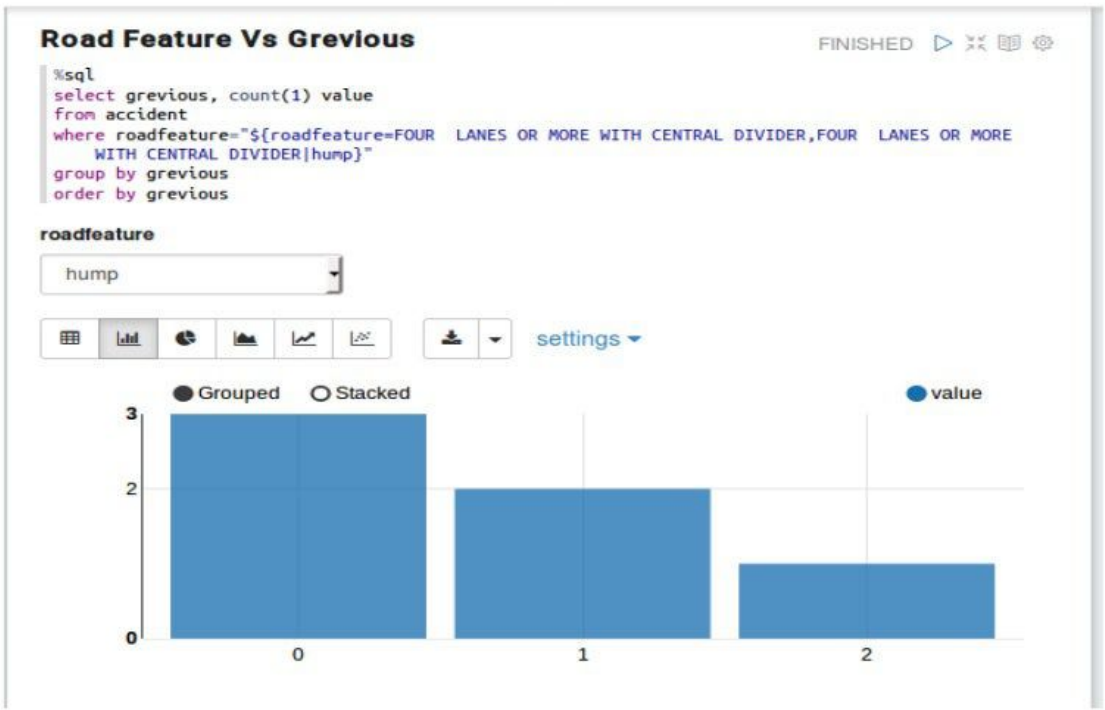


FIGURE 3.5.12: BAR GRAPH SHOWING NUMBER GREVIOUS ACCIDENTS IN Y-AXIS FOR GREVIOUS VALUES IN X-AXIS WHEN ROAD FEATURE IS HUMP

Road Feature Vs Minor:

Road Feature Vs Minor FINISHED

```
%sql
select minor, count(1) value
from accident
where roadfeature="${roadfeature=FOUR  LANES OR MORE WITH CENTRAL DIVIDER,Two lanes|hump}"
group by minor
order by minor
```

roadfeature

hump

minor	value
0	3
1	1
2	1
3	1

FIGURE 3.5.13: TABLE SHOWING NUMBER OF MINOR ACCIDENTS WHEN ROAD FEATURE IS HUMP

FIGURE 3.5.14: BAR GRAPH SHOWING NUMBER OF MINOR ACCIDENTS IN Y-AXIS FOR MINOR VALUES IN X-AXIS WHEN ROAD FEATURE IS HUMP

Fatal vs accident location and accident time:

Fatal Vs Accident Location and Accident Time FINISHED

```
%sql
select location, acctime, fatal, count(fatal)
from accident
where fatal>0
group by location, acctime, fatal
order by location
```

location	acctime	fatal	count(fatal)
	10:00 AM	1	1
1+800 RHS	3:00 PM	1	1
10+000 RHS	11:15 PM	1	1
11+300 RHS	4:00 PM	1	1
11+300 RHS	6:00 AM	2	1
12+000	10:00 PM	1	1
12+400 LHS	6:30 AM	1	1
12+700 RHS	6:22 PM	2	1

Took 16 sec. Last updated by user1 at June 21 2017, 1:03:13 PM. (outdated)

FIGURE 3.5.15 : TABLE SHOWING NUMBER OF FATAL ACCIDENTS IN A LOCATION AT A TIME

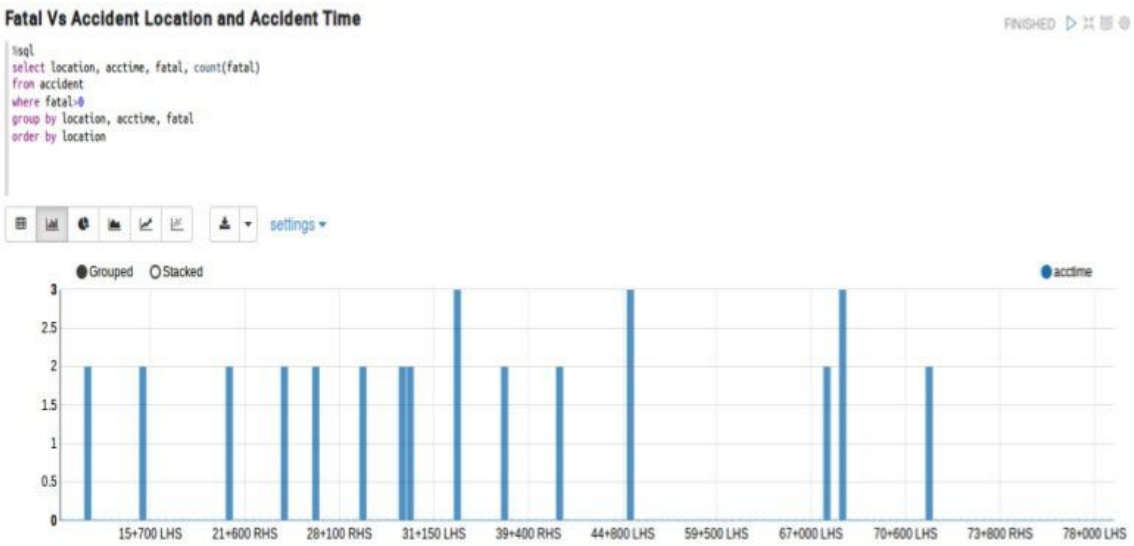


FIGURE 3.5.16 : BAR GRAPH SHOWING NUMBER OF FATAL ACCIDENTS IN A LOCATION AT A TIME

CHAPTER 4

TEST CASES

Testing is a type of program change life cycle in the midst of which the application is rehearsed with the final objective of finding oversights. Structure checking truly a movement having variety of tests of different tests whose fundamental part is mainly to totally hone all PC based systems. But every test has another reason, work needed to affirm which every structure parts are suitably fused and also performed distributed limits. This testing methodology is considered as truly done to guarantee that the thing decisively does similarly what ought to. The testing is the last affirmation and also the acknowledgement functions inside for the affiliation. In the midst of testing considering noteworthy activities that are centered around for the testing and the modification of the source code. Here testing stage taking after targets or proficient.

- Affirmation the way of the errand.
- Find and avoid any type of remaining slip-ups from past stages.
- Acknowledge the items in order to get the response for the primary issues.
- Give an operational resolute nature of the needed structure.

4.1 Software Test Environment

Preparation of the test data accept an essential part in the structure testing. In the wake of setting up the test data, the structure under study is taken a stab for using the tested data. In the case of testing data, it should depend on the previous test data result and comparison should do with the previous.

4.2 Test Case and Procedures

The test cases results are the key variables to the achievement of any system. Test cases are fundamental for the productive execution of structure. The test procedures will be summoning frameworks which with everything taken into account will do the going with:

- Compile and join reference program (if correlated).
- Execute reference program with test data.
- Request and association test driver.

4.3 Unit test cases

In this section, it incorporates the general system is attempted autonomously. Unit testing are mainly focuses onto check attempts even for the smallest unit of programming framework in all the module. In this case its generally called as module testing. Modules in the system are attempted autonomously. In this testing is done with its own programming style. Unit testing mainly hones particular roots in the modules control structure in order to ensure complete extension and greatest mix-up acknowledgement.

4.4 Test Cases:

TABLE 4.4.1: Skidding

Test Case No.	1
Name of Test Case	Skidding
Feature Being Tested	Level of injury
Expected Output	Number of people injured
Actual Output	Graph and no. of people affected
Remarks	Pass

TABLE 4.4.2: Major injury

Test Case No.	2
Name of Test Case	Major injury
Feature Being Tested	Classification
Expected Output	No. of people affected
Actual Output	Graph and no. of people
Remarks	Pass

TABLE 4.4.3: Over Speeding

Test Case No.	3
Name of Test Case	Over speeding
Feature Being Tested	Cause
Expected Output	No. of people affected
Actual Output	Graph and no. of people
Remarks	Pass

TABLE 4.4.4: Hump

Test Case No.	4
Name of Test Case	Hump
Feature Being Tested	Road condition
Expected Output	No. of people affected
Actual Output	Graph and no. of people
Remarks	Pass

TABLE 4.4.5: Sharp curve

Test Case No.	5
Name of Test Case	Sharp curve
Feature Being Tested	Road condition
Expected Output	No. of people affected
Actual Output	Graph and no. of people
Remarks	Pass

TABLE 4.4.6: Four lanes or more with central divider

Test Case No.	6
Name of Test Case	Four lanes or more with central divider
Feature Being Tested	Road condition
Expected Output	No. of people affected
Actual Output	Graph and no. of people
Remarks	Pass

CHAPTER 5

CONCLUSION AND FUTURE SCOPE

In this paper, a framework is proposed for analyzing accident patterns for different types of accidents on the road which makes use of clustering and decision tree algorithm. The study uses 1,208 accidents that have occurred around K.R.Puram Traffic police station limits from 2012 to 2016. Clustering algorithm finds cluster based on attributes accident type, road type, causes ,road feature etc. J48 decision tree algorithm have been applied on each cluster as well as on EDS to induce decision tree. More information can be identified if more features are available that is associated with an accident.

The tree generated is pruned to large extent due to memory restrictions and varied type of data. Further room for improvement exists by adding more clusters to the distributed processing module & using more user friendly visualizations. The analysis can be used to develop preventive measures using Image Processing techniques for the vehicles violating traffic rules or for the vehicles that match many attributes in this project. Preventive measure can be developed in the locations which are more prone to accidents found from the analysis.

BIBLIOGRAPHY

1. Sachin Kumar , Durga Toshniwal ,"Analyzing Road Accident Data Using Association Rule Mining, International Conference on Computing, Communication and Security (ICCCS)", IEEE 2015.

2. An Shi,Zhang Tao, Zhang Xinming, Wang Jian, "Evolution of Traffic Flow Analysis under Accidents on Highways Using Temporal Data Mining", Fifth International Conference on Intelligent Systems Design and Engineering Applications,2014.

3. Eyad Abdullah, Ahmed Emam, "Traffic Accidents Analyzer Using Big Data", International Conference On Computational Science and Computational Intelligence, 2015.

4. Seoung-hun Park ,Young-guk Ha, "Large Imbalance Data Classification Based on MapReduce for Traffic Accident Prediction", Eighth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing,2014.

5. Lokesh Hebbani, "Road Safety Scenario in India Problems & Solutions", 5th Foundation Day Lecture CiSTUP, IISC January 10, 2014.

6. Costabilea. J., Walla, J., Vecovskia, V & Baileya, "The rapid deployment of an effective road safety counter measure through a smart phone application- The story of Speed Adviser", Proceedings of the Australasian Road Safety Research, Policing & Education Conference November,2014.