IBM Innovation Center for Education

PHEMESOFT Software That Matters
TRUSTED DELIVERY PARTNER for IBM ICE

UNIVERSITY OF PETROLEUM

AND

ENERGY STUDIES

BACHELOR OF TECHNOLOGY

Computer Science & Engineering

(specialization in Artificial Intelligence And Machine Learning)

A DATA MINING FRAMEWORK TO ANALYZE ROAD ACCIDENT DATA

PROJECT REPORT

Version No.	1.00
Authorized by	PIYUSH MALVIYA PRIYAL GUPTA SAKSHAM GARG RAHUL DHANOLA

Industry Mentor:	**Guided By:**
Sumit Shukla	Sujoy Chatterjee

1. Background

In developed as well as developing countries, infrastructure development is one of the major investments by the government, while the safety of passengers on roads is of utmost importance. A road optimization during the construction or the 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. Due to road accidents, a large number of lives are lost. From an analysis, it has been estimated that every year over 3,00,000 people die and 10 to 15 million people are injured due to road accidents in the entire world. There are a lot of vehicles driving on the roadway every day, and traffic accidents could happen at any time anywhere. Some accident involves a fatality, which means people die in that accident. As human beings, we all want to avoid accidents and stay safe. Traffic accidents have now earned india a dubious distinction; with nearly 140,000 deaths annually, the country has overtaken china to top the world in road fatalities. India is the only country in the world with more than 15 fatalities and 53 injuries every hour due to road crashes. The roads of india have not abated their contribution to traffic accident fatalities. The accident rate in india has been on an increase ever since the start of the century. Data mining analyses can help identify the major causes and help the transport authorities in improving safety requirements.

1.1 Aim

Our main aim is to study the states and the union territories of India against the contributing causes to facilitating road safety in the country. We are focused on taking the aid of clustering to group similar objects of this dataset to group regions based on vulnerability. The major problem in the analysis of accident data is its Heterogeneous nature. Thus, heterogeneity must be considered during the analysis of the data.

Road accident analysis aims to investigate the main factors that characterize an accident to understand patterns of behaviours and, consequently, to identify the appropriate countermeasures to adopt to avoid the accident. In this, we will use different data mining algorithms to analyze the data.

Different algorithms are applied to group the accident locations into clusters and mining techniques are used to characterize the locations. Most states of traffic management and information systems focus on data analysis. Python and Jupyter notebooks are mainly used. Since Python has a large number of libraries and packages, it has a very large ecosystem.

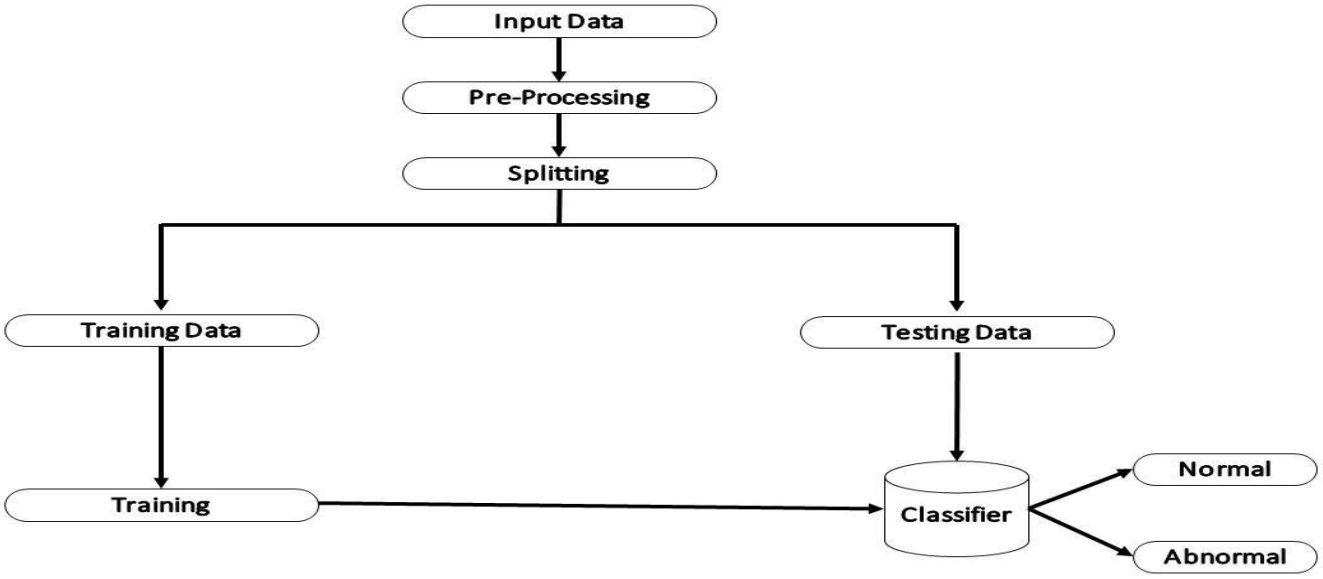
Python is used both in data scraping and in developing the server. Jupyter notebook is an open-source and web-based interactive environment for making notebook documents. The primary Jupyter online application and Jupyter python web server are the substances required for making a notebook

1.2 Technologies

In this, we will use different data mining algorithms to analyze the data. Different algorithms are applied to group the accident locations into clusters and mining techniques are used to characterize the locations. Most states of traffic management and information systems focus on data analysis. Python and Jupyter notebooks are mainly used.

Since Python has a large number of libraries and packages, it has a very large ecosystem. Python is used both in data scraping and in developing the server. Jupyter notebook is an open-source and web-based interactive environment for making notebook documents. The primary Jupyter online application and Jupyter python web server are the substances required for making a notebook.

1.3 Hardware Architecture



- **Data preprocessing:** Data preprocessing is one of the important tasks in data mining. Data preprocessing mainly deals with removing noise, handling missing values, and removing irrelevant attributes to make the data ready for analysis. In this step, we aim to preprocess the accident data to make it appropriate for the analysis.

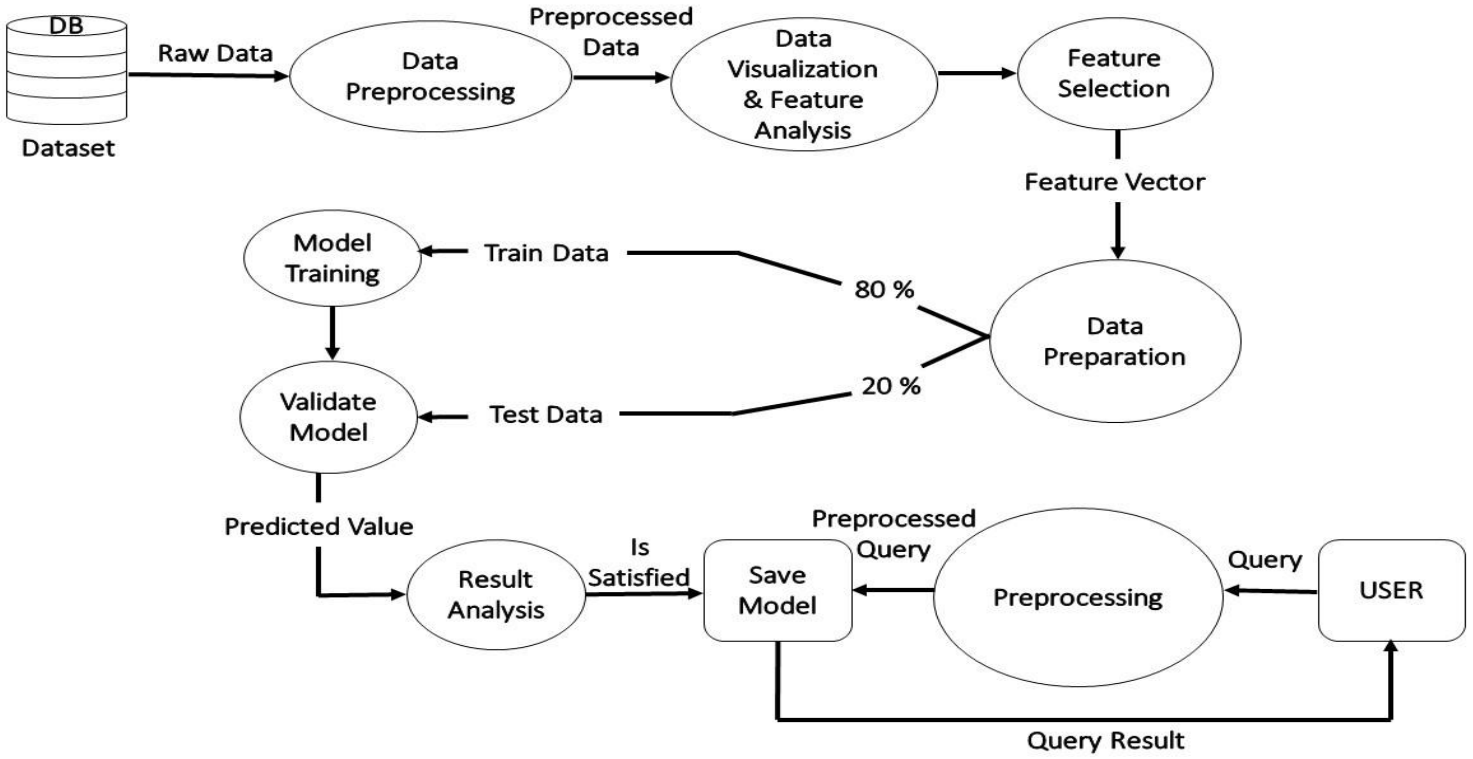
- **Clustering algorithm:** There are several clustering algorithms in the literature. The objective of the clustering algorithm is to divide the data into different clusters or groups such that the objects within a group are similar to each other whereas objects in other clusters are different from each other. K means & Decision trees have been used in road accident analysis.

- **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 the 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. Further association rules are generated from the frequent item sets and strong rules based on interestingness measures are taken for the analysis.

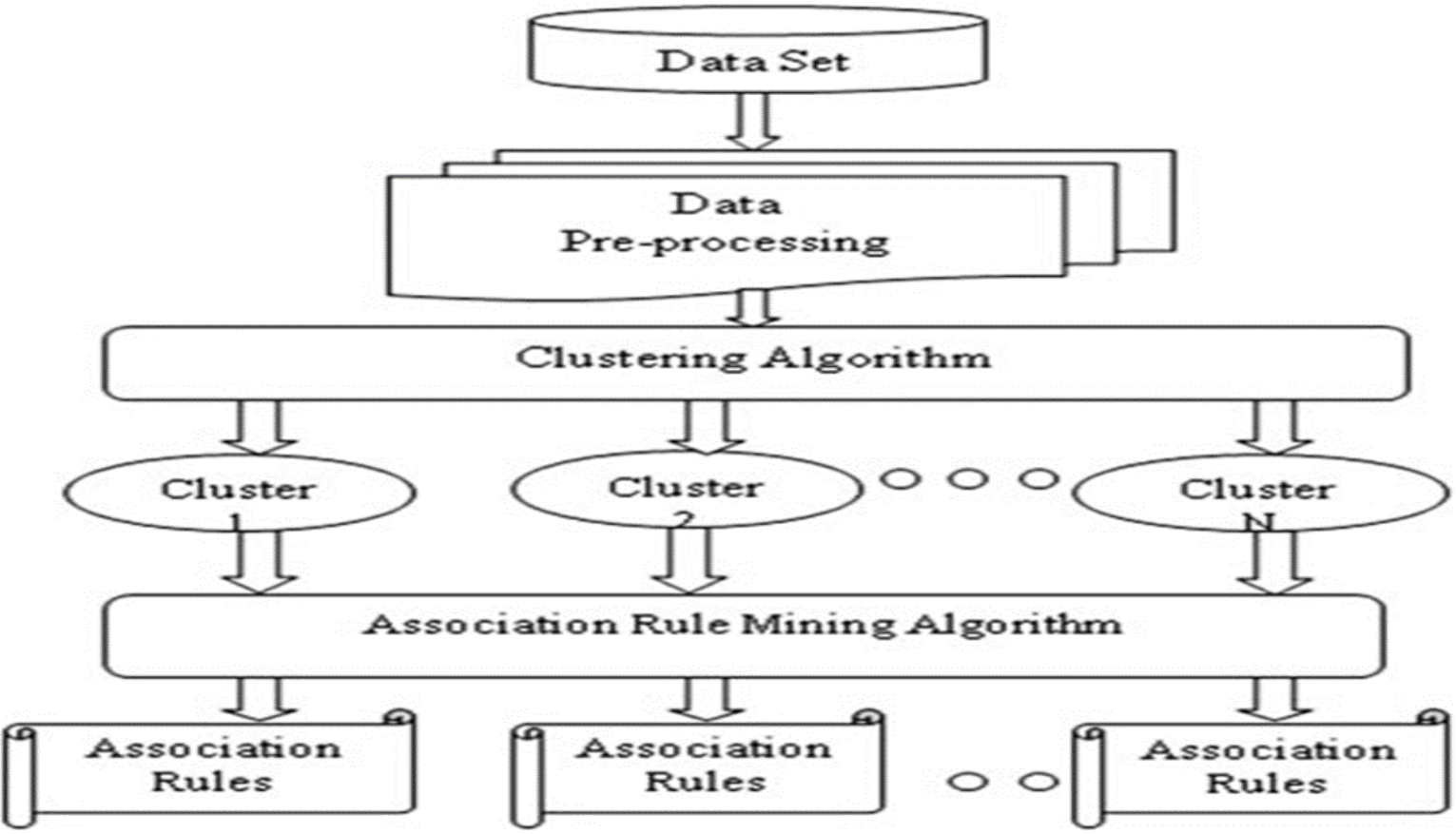
1.4 Software Architecture

This project can run on commodity hardware. The first part is the training phase which takes 10-15 mins and the second part is the testing part which only takes a few seconds to perform classification and calculate accuracy.

2. System



This system provides a well-analyzed data mining framework which indicates the accidents and their causes according to the authentic data obtained from data.gov.in. In this, we will use different data mining algorithms to analyze the data. Different algorithms are applied to group the accident locations into clusters and mining techniques are used to characterize the locations.



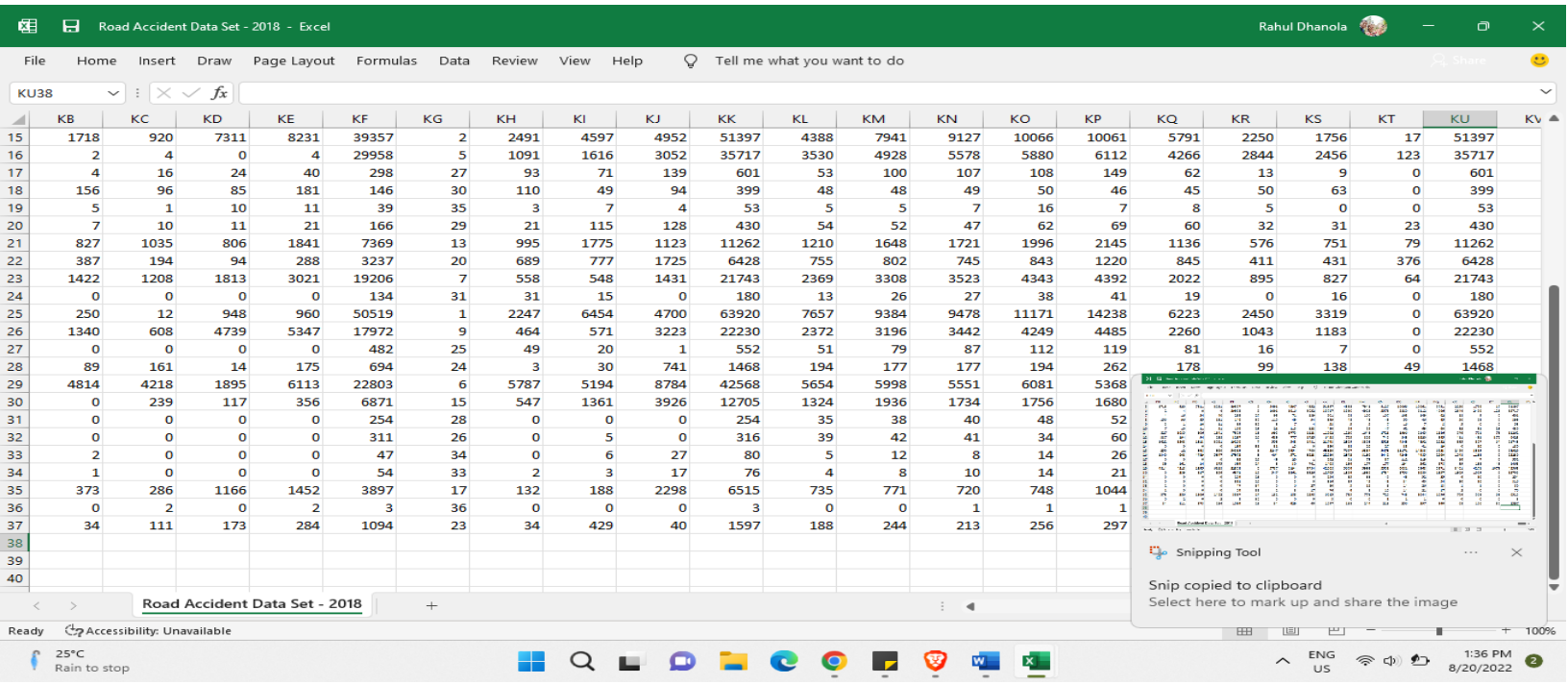
2.1 Requirements

2.1.1 Functional Requirement

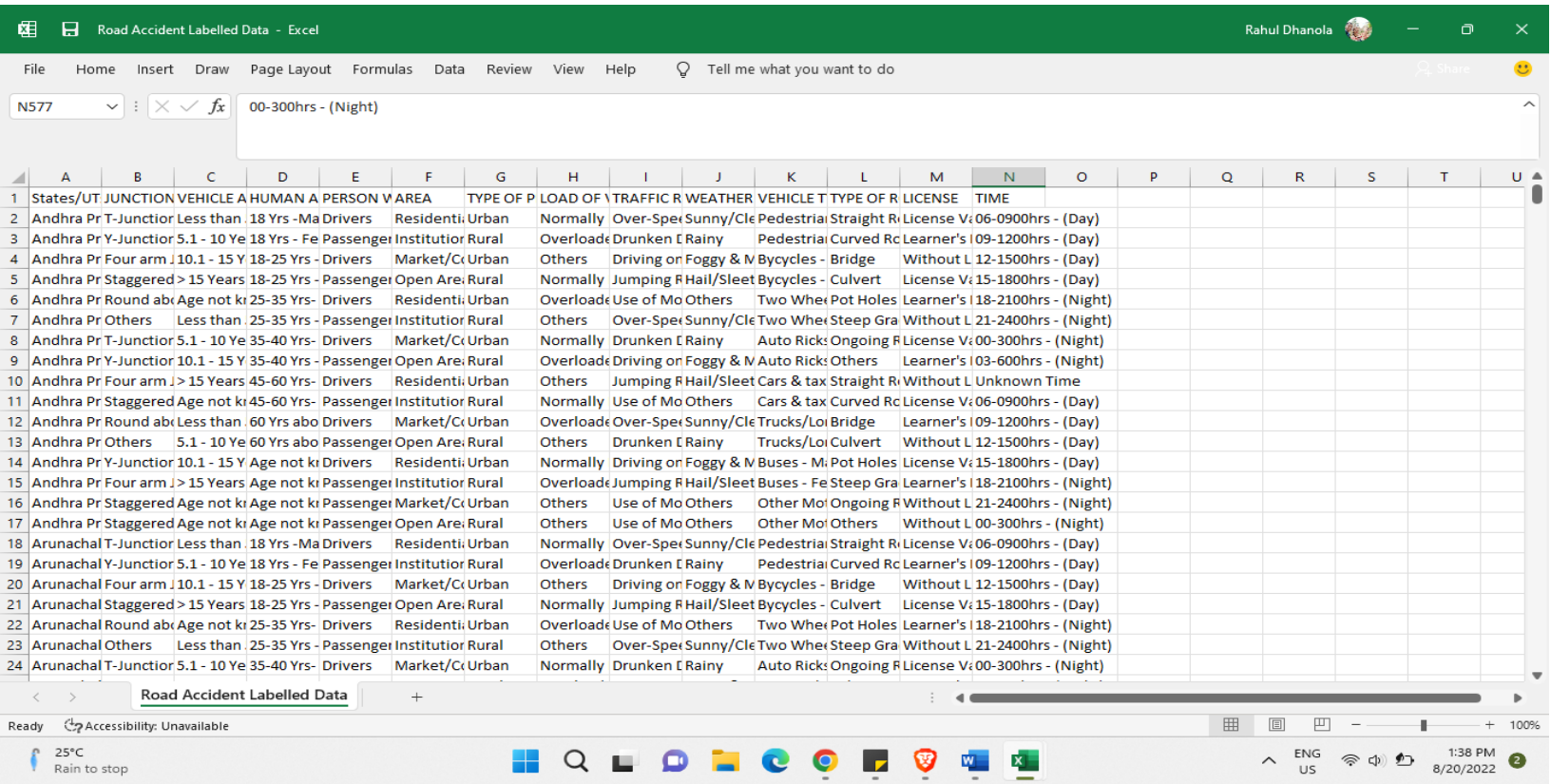
Type	Name
Programming Language with Version	Python 3.5 & above
Operating System	Windows 7 & above. Linux based OS Mac OS

2.1.2 User Requirement: The user should have the required data before implementing the project.

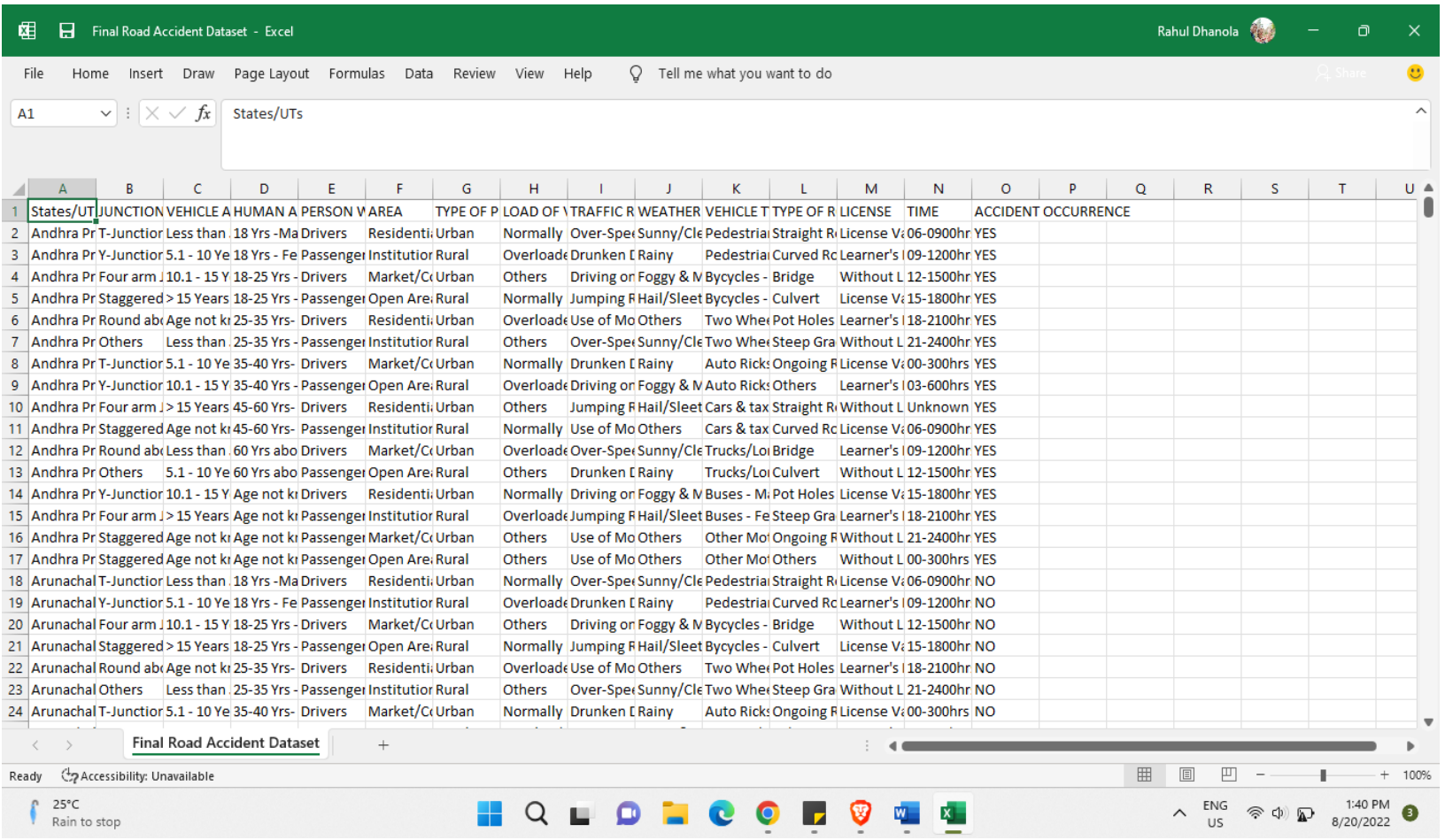
Dataset 1: Road Accident Data Set – 2018



Dataset 2: Road Accident Labelled Data



Dataset 3: Final Road Accident Data



States/UT	JUNCTION	VEHICLE A	HUMAN A	PERSON V	AREA	TYPE OF P	LOAD OF V	TRAFFIC R	WEATHER	VEHICLE T	TYPE OF R	LICENSE	TIME	ACCIDENT OCCURRENCE
Andhra Pr	T-Junctior	Less than	18 Yrs -Ma	Drivers	Residenti	Urban	Normally	Over-Spe	Sunny/Cle	Pedestria	Straight R	License Va	06-0900hr	YES
Andhra Pr	Y-Junctior	5.1 - 10 Ye	18 Yrs - Fe	Passenger	Institutior	Rural	Overloade	Drunken D	Rainy	Pedestria	Curved Ro	Learner's	09-1200hr	YES
Andhra Pr	Four arm J	10.1 - 15 Y	18-25 Yrs -	Drivers	Market/Co	Urban	Others	Driving or	Foggy & M	Bycycles -	Bridge	Without L	12-1500hr	YES
Andhra Pr	Staggered	> 15 Years	18-25 Yrs -	Passenger	Open Area	Rural	Normally	Jumping F	Hail/Sleet	Bycycles -	Culvert	License Va	15-1800hr	YES
Andhra Pr	Round abo	Age not kr	25-35 Yrs-	Drivers	Residenti	Urban	Overloade	Use of Mo	Others	Two Whee	Pot Holes	Learner's	18-2100hr	YES
Andhra Pr	Others	Less than	25-35 Yrs -	Passenger	Institutior	Rural	Others	Over-Spe	Sunny/Cle	Two Whee	Steep Gra	Without L	21-2400hr	YES
Andhra Pr	T-Junctior	5.1 - 10 Ye	35-40 Yrs-	Drivers	Market/Co	Urban	Normally	Drunken D	Rainy	Auto Ricks	Ongoing R	License Va	00-300hrs	YES
Andhra Pr	Y-Junctior	10.1 - 15 Y	35-40 Yrs -	Passenger	Open Area	Rural	Overloade	Driving or	Foggy & M	Auto Ricks	Others	Learner's	03-600hrs	YES
Andhra Pr	Four arm J	> 15 Years	45-60 Yrs-	Drivers	Residenti	Urban	Others	Jumping F	Hail/Sleet	Cars & tax	Straight R	Without L	Unknown	YES
Andhra Pr	Staggered	Age not kr	45-60 Yrs-	Passenger	Institutior	Rural	Normally	Use of Mo	Others	Cars & tax	Curved Ro	License Va	06-0900hr	YES
Andhra Pr	Round abo	Less than	60 Yrs abo	Drivers	Market/Co	Urban	Overloade	Over-Spe	Sunny/Cle	Trucks/Lo	Bridge	Learner's	09-1200hr	YES
Andhra Pr	Others	5.1 - 10 Ye	60 Yrs abo	Passenger	Open Area	Rural	Others	Drunken D	Rainy	Trucks/Lo	Culvert	Without L	12-1500hr	YES
Andhra Pr	Y-Junctior	10.1 - 15 Y	Age not kr	Drivers	Residenti	Urban	Normally	Driving or	Foggy & M	Buses - Mi	Pot Holes	License Va	15-1800hr	YES
Andhra Pr	Four arm J	> 15 Years	Age not kr	Passenger	Institutior	Rural	Overloade	Jumping F	Hail/Sleet	Buses - Fe	Steep Gra	Learner's	18-2100hr	YES
Andhra Pr	Staggered	Age not kr	Age not kr	Passenger	Market/Co	Urban	Others	Use of Mo	Others	Other Mo	Ongoing R	Without L	21-2400hr	YES
Andhra Pr	Staggered	Age not kr	Age not kr	Passenger	Open Area	Rural	Others	Use of Mo	Others	Other Mo	Others	Without L	00-300hrs	YES
Arunachal	T-Junctior	Less than	18 Yrs -Ma	Drivers	Residenti	Urban	Normally	Over-Spe	Sunny/Cle	Pedestria	Straight R	License Va	06-0900hr	NO
Arunachal	Y-Junctior	5.1 - 10 Ye	18 Yrs - Fe	Passenger	Institutior	Rural	Overloade	Drunken D	Rainy	Pedestria	Curved Ro	Learner's	09-1200hr	NO
Arunachal	Four arm J	10.1 - 15 Y	18-25 Yrs -	Drivers	Market/Co	Urban	Others	Driving or	Foggy & M	Bycycles -	Bridge	Without L	12-1500hr	NO
Arunachal	Staggered	> 15 Years	18-25 Yrs -	Passenger	Open Area	Rural	Normally	Jumping F	Hail/Sleet	Bycycles -	Culvert	License Va	15-1800hr	NO
Arunachal	Round abo	Age not kr	25-35 Yrs-	Drivers	Residenti	Urban	Overloade	Use of Mo	Others	Two Whee	Pot Holes	Learner's	18-2100hr	NO
Arunachal	Others	Less than	25-35 Yrs -	Passenger	Institutior	Rural	Others	Over-Spe	Sunny/Cle	Two Whee	Steep Gra	Without L	21-2400hr	NO
Arunachal	T-Junctior	5.1 - 10 Ye	35-40 Yrs-	Drivers	Market/Co	Urban	Normally	Drunken D	Rainy	Auto Ricks	Ongoing R	License Va	00-300hrs	NO

2.1.3 Environmental Requirement:

Type	Name
System Architecture	32 – bit or 64 – bit
Memory	8 GB 2400 MHz DDR4
Storage	500 GB SATA3 2.4 HDD
Central Processing Unit (CPU)	Intel or AMD - 2 GHz or Faster
Graphical Processor Unit (GPU)	2 GB Nvidia Graphic Processor

2.2 Implementation

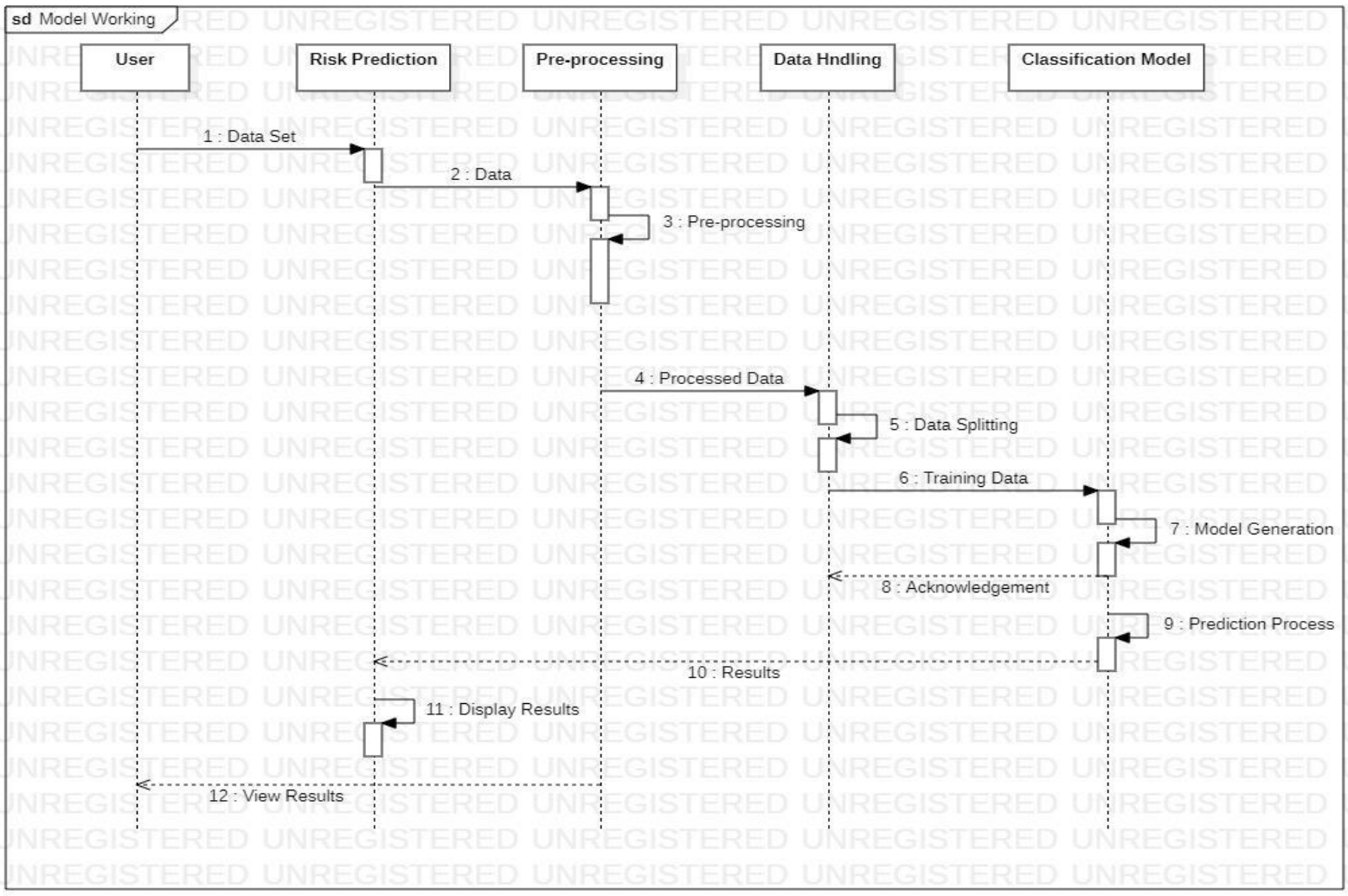


Fig: Sequence Diagram of System

2.3 Testing

We will develop and train the machine learning model from scratch and subsequently, a testing framework will also be made for the classification module. We will use many classification datasets to test the model. We will also document the performance and accuracy of our model.

Test Data1:

In [103_
```
testData1 = {
    'States/UTs': ['Kerala'],
    'JUNCTION': ['Four arm Junction'],
    'VEHICLE AGE': ['> 15 Years'],
    'HUMAN AGE AND SEX': ['45-60 Yrs- Male'],
    'PERSON WITHOUT SAFETY PRECAUTIONS': ['Drivers'],
    'AREA': ['Residential Area'],
    'TYPE OF PLACE': ['Urban'],
    'LOAD OF VEHICLE': ['Others'],
    'TRAFFIC RULES VIOLATION': ['Jumping Red Light'],
    'WEATHER': ['Hail/Sleet'],
    'VEHICLE TYPE AND SEX': ['Cars & taxies Vans & LMV - Male'],
    'TYPE OF ROAD': ['Straight Road'],
    'LICENSE': ['Without Licence'],
    'TIME': ['Unknown Time']
}
```

In [104_
```
# Now the test values are compared with the Column Codes and store them
for col in testData1:
    code = [columnCodes[''.join(testData1[col])]]
    testData1[col] = code
print(testData1)

testDataFrame = pd.DataFrame.from_dict(testData1)
print(testDataFrame)
```

```
{'States/UTs': [17], 'JUNCTION': [0], 'VEHICLE AGE': [2], 'HUMAN AGE AND SEX': [9], 'PERSON WITHOUT SAFETY PREC
AUTIONS': [0], 'AREA': [3], 'TYPE OF PLACE': [1], 'LOAD OF VEHICLE': [1], 'TRAFFIC RULES VIOLATION': [2], 'WEAT
HER': [1], 'VEHICLE TYPE AND SEX': [7], 'TYPE OF ROAD': [7], 'LICENSE': [2], 'TIME': [8]}
   States/UTs  JUNCTION  VEHICLE AGE  HUMAN AGE AND SEX  \
0          17         0            2                  9

   PERSON WITHOUT SAFETY PRECAUTIONS  AREA  TYPE OF PLACE  LOAD OF VEHICLE  \
0                                  0     3              1                1

   TRAFFIC RULES VIOLATION  WEATHER  VEHICLE TYPE AND SEX  TYPE OF ROAD  \
0                        2        1                     7             7

   LICENSE  TIME
0        2     8
```

In [105_
```
predictionValue = trainedModel.predict(testDataFrame)
if (predictionValue == 1):
    print("Yes, there is a chance of Accident")
else:
    print("No, it is safe")
```

```
Yes, there is a chance of Accident
```

Test Data2:

In [106_
```
testData2 = {
    'States/UTs': ['Chhattisgarh'],
    'JUNCTION': ['Staggered Junction'],
    'VEHICLE AGE': ['> 15 Years'],
    'HUMAN AGE AND SEX': ['45-60 Yrs- Male'],
    'PERSON WITHOUT SAFETY PRECAUTIONS': ['Passengers'],
    'AREA': ['Residential Area'],
    'TYPE OF PLACE': ['Rural'],
    'LOAD OF VEHICLE': ['Others'],
    'TRAFFIC RULES VIOLATION': ['Jumping Red Light'],
    'WEATHER': ['Hail/Sleet'],
    'VEHICLE TYPE AND SEX': ['Other Motor Vehicles - Female'],
    'TYPE OF ROAD': ['Others'],
    'LICENSE': ['Without Licence'],
    'TIME': ['Unknown Time']
}
```

In [107_
```
# Now the test values are compared with the Column Codes and store them
for col in testData2:
    code = [columnCodes[''.join(testData2[col])]]
    testData2[col] = code
print(testData2)
testDataFrame = pd.DataFrame.from_dict(testData2)
```

```
{'States/UTs': [6], 'JUNCTION': [3], 'VEHICLE AGE': [2], 'HUMAN AGE AND SEX': [9], 'PERSON WITHOUT SAFETY PRECA
UTIONS': [1], 'AREA': [3], 'TYPE OF PLACE': [0], 'LOAD OF VEHICLE': [1], 'TRAFFIC RULES VIOLATION': [2], 'WEATH
ER': [1], 'VEHICLE TYPE AND SEX': [8], 'TYPE OF ROAD': [1], 'LICENSE': [2], 'TIME': [8]}
```

In [108_
```
predictionValue = trainedModel.predict(testDataFrame)
if (predictionValue == 1):
    print("Yes, there is a chance of Accident")
else:
    print("No, it is safe")
```

```
Yes, there is a chance of Accident
```

Test Data3:

In [109_
```
testData3 = {
    'States/UTs': ['Puducherry'],
    'JUNCTION': ['Staggered Junction'],
    'VEHICLE AGE': ['> 15 Years'],
    'HUMAN AGE AND SEX': ['45-60 Yrs- Male'],
    'PERSON WITHOUT SAFETY PRECAUTIONS': ['Passengers'],
    'AREA': ['Residential Area'],
    'TYPE OF PLACE': ['Rural'],
    'LOAD OF VEHICLE': ['Others'],
    'TRAFFIC RULES VIOLATION': ['Jumping Red Light'],
    'WEATHER': ['Hail/Sleet'],
    'VEHICLE TYPE AND SEX': ['Other Motor Vehicles - Female'],
    'TYPE OF ROAD': ['Others'],
    'LICENSE': ['Without Licence'],
    'TIME': ['Unknown Time']
}
```

In [110_
```
# Now the test values are compared with the Column Codes and store them
for col in testData3:
    code = [columnCodes[''.join(testData3[col])]]
    testData3[col] = code
print(testData3)
testDataFrame = pd.DataFrame.from_dict(testData3)
```

```
{'States/UTs': [26], 'JUNCTION': [3], 'VEHICLE AGE': [2], 'HUMAN AGE AND SEX': [9], 'PERSON WITHOUT SAFETY PREC
AUTIONS': [1], 'AREA': [3], 'TYPE OF PLACE': [0], 'LOAD OF VEHICLE': [1], 'TRAFFIC RULES VIOLATION': [2], 'WEAT
HER': [1], 'VEHICLE TYPE AND SEX': [8], 'TYPE OF ROAD': [1], 'LICENSE': [2], 'TIME': [8]}
```

In [111_
```
predictionValue = trainedModel.predict(testDataFrame)
if (predictionValue == 1):
    print("Yes, there is a chance of Accident")
else:
    print("No, it is safe")
```

```
No, it is safe
```

Test Data4:

In [112_
```
testData4 = {
    'States/UTs': ['Lakshadweep'],
    'JUNCTION': ['Staggered Junction'],
    'VEHICLE AGE': ['10.1 - 15 Years'],
    'HUMAN AGE AND SEX': ['45-60 Yrs- Male'],
    'PERSON WITHOUT SAFETY PRECAUTIONS': ['Drivers'],
    'AREA': ['Institutional Area'],
    'TYPE OF PLACE': ['Rural'],
    'LOAD OF VEHICLE': ['Others'],
    'TRAFFIC RULES VIOLATION': ['Jumping Red Light'],
    'WEATHER': ['Others'],
    'VEHICLE TYPE AND SEX': ['Other Motor Vehicles - Female'],
    'TYPE OF ROAD': ['Others'],
    'LICENSE': ['License Valid Permanent'],
    'TIME': ['00-300hrs - (Night)']
}
```

In [113_
```
# Now the test values are compared with the Column Codes and store them
for col in testData4:
    code = [columnCodes[''.join(testData4[col])]]
    testData4[col] = code
print(testData4)
testDataFrame = pd.DataFrame.from_dict(testData4)
```

```
{'States/UTs': [18], 'JUNCTION': [3], 'VEHICLE AGE': [0], 'HUMAN AGE AND SEX': [9], 'PERSON WITHOUT SAFETY PREC
AUTIONS': [0], 'AREA': [0], 'TYPE OF PLACE': [0], 'LOAD OF VEHICLE': [1], 'TRAFFIC RULES VIOLATION': [2], 'WEAT
HER': [1], 'VEHICLE TYPE AND SEX': [8], 'TYPE OF ROAD': [1], 'LICENSE': [1], 'TIME': [0]}
```

In [114_
```
predictionValue = trainedModel.predict(testDataFrame)
if (predictionValue == 1):
    print("Yes, there is a chance of Accident")
else:
    print("No, it is safe")
```

```
No, it is safe
```

2.4 Evaluation

We will develop and train the machine learning model from scratch and subsequently, a testing framework will also be made for the classification module. We will use many classification datasets to test the model. We will also document the performance and accuracy of our model.

Whenever we build Machine Learning models, we need some form of metric used for the measurement of the goodness of the model. Bear in mind that the "goodness" of the model could have multiple interpretations, but generally when we speak of it in a Machine Learning context, we are talking of the measure of a model's performance on new instances that weren't a part of the training data.

Some common intrinsic metrics to evaluate the system are as follows:

- **Accuracy**

  The accuracy of a Machine Learning classification algorithm is one way to measure how often the algorithm classifies a data point correctly. Accuracy is the number of correctly predicted data points out of all the data points.

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ prediction}$$

- **Precision**

  Precision (also called positive predictive value) is the fraction of relevant instances among the retrieved instances

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$= \frac{True\ Positive}{Total\ predicted\ positive}$$

- **Recall**

  Recall measures the proportion of actual positive labels correctly identified by the model.

$$Recall = \frac{True\ Positive}{True\,Positive + False\,Negative}$$

$$= \frac{True\ Positive}{Total\ Actual\ positive}$$

3 Snapshots of the Project

Snapshot 1:

A Data Mining Framework To Analyze Road Accident Data - Road Accident Data Analysis

Introduction

Road accidents are uncertain and unpredictable incidents and their analysis requires the knowledge of the factors affecting them.Fatalities and injuries resulting from road traffic accidents are a major and growing public health problem in India. Every week nearly 2,650 people get killed and 9,000 get injured due to traffic accidents.

Traffic accidents have now earned India a dubious distinction; with nearly 140,000 deaths annually, the country has overtaken China to top the world in road fatalities. India is the only country in the world which faces more than 15 fatalities and 53 injuries every hour as a consequence of road crashes.

The major problem in the analysis of accident data is its Heterogeneous nature. Thus, heterogeneity must be considered during analysis of the data. Road accident analysis aims to investigate the main factors that characterize an accident to understand patterns or behaviors and, consequently, to identify the appropriate countermeasures to adopt to avoid the accident.

Importing the Libraries

In [1]:
```
# Importing the necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plot
```

Reading the Dataset

In [2]:
```
# Reading the csv file
data = pd.read_csv(
    '../input/road-accident-analysis-Data/Road Accident Analysis Data/Analysis/Road Accident Data Set - 2018.csv',
    index_col=0)
```

Snapshot 2:

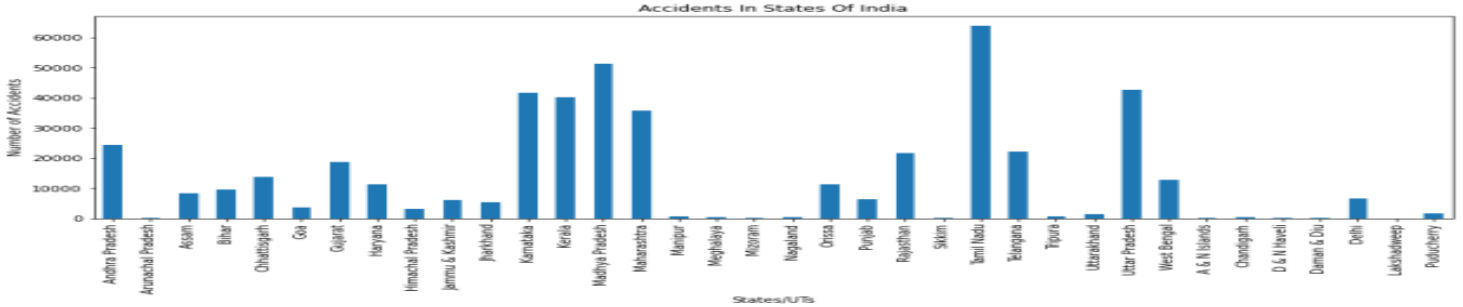
Exploratory Data Analysis (EDA)

In statistics, exploratory data analysis is an approach of analyzing data sets to summarize their main characteristics, often using statistical graphics and other data visualization methods. A statistical model can be used or not, but primarily EDA is for seeing what the data can tell us beyond the formal modeling or hypothesis testing task.

Total Number Of Accidents Occured across The Indian States

In [8]:
```
# Total Number Of Accidents Occured across The Indian States
data['Total Accidents'].plot.bar(figsize=(15, 5),
                                 title="Accidents In States Of India")
plot.ylabel('Number of Accidents')
```

Out[8]:
```
Text(0, 0.5, 'Number of Accidents')
```



Snapshot 3:

Model - 2 => Using Random Forest

In [99]:
```
RF = RandomForestClassifier()
RF.fit(accidentData_train, target_train)
target_pred_rf = RF.predict(accidentData_test)
print('Accuracy score: {0:0.2f}'.format(
    metrics.accuracy_score(target_test, target_pred_rf)))
```

```
Accuracy score: 0.57
```

We have achieved a accuracy of 55%.

Model - 3 => Using Decision Tree

In [100…
```
DT = DecisionTreeClassifier()
DT.fit(accidentData_train, target_train)
target_pred_dt = DT.predict(accidentData_test)
print('Accuracy score: {0:0.2f}'.format(
    metrics.accuracy_score(target_test, target_pred_dt)))
```

```
Accuracy score: 0.91
```

We have achieved a accuracy of 94%.

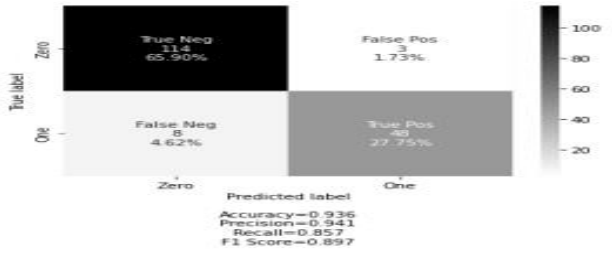
Model - 4 => Using Decision Tree - Using AdaBoostClassifier

In [101…
```
# Create adaboost classifer object
adaboostClassifier = AdaBoostClassifier()

# Train Adaboost Classifer
trainedModel = adaboostClassifier.fit(accidentData_train, target_train)

# Predict the response for test dataset
target_pred_ada = trainedModel.predict(accidentData_test)

# Printing the Accuracy
print("Accuracy: {0:0.2f}".format(
    metrics.accuracy_score(target_test, target_pred_ada)))
```

```
Accuracy: 0.94
```

We have achieved a accuracy of 95%.

Snapshot 4:



Evaluating The Model

Whenever we build Machine Learning models, we need some form of metric used for the measurement of the goodness of the model. Bear in mind that the "goodness" of the model could have multiple interpretations, but generally when we speak of it in a Machine Learning context, we are talking of the measure of a model's performance on new instances that weren't a part of the training data.

Predictive Model: Evaluation

$$\text{Accuracy} = \frac{tp + tn}{tp + tn + fp + fn}$$

predictive result / classification		actual result / classification: yes	actual result / classification: no
	yes	**tp** (true positive)	**fp** (false positive) ← Type 1 error
	no	**fn** (false negative)	**tn** (true negative)

$$\text{Precision} = \frac{tp}{tp + fp} \qquad \text{Recall} = \frac{tp}{tp + fn}$$

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \qquad \text{True Negative Rate} = \frac{tn}{tn + fp}$$

4 Conclusions

This framework brings us into an analysis of road accident data which indicates to us the major causes and conditions of an accident where it is important to understand the basic probabilities where we get the maximum accident depending upon the region as well as the factors like over speeding, drink and drive and other major and minute factors to inculcate the accident and also to improve or prevent them from happening in all situations depending on weather conditions and regions where the accidents take place majorly.

This framework also helps to analyze the fatal rates to indicate the level of the cause and accidents, using clustering and algorithms we can conclude the probabilities and accuracy of the model that what will be the expected rate of an accident to take place and further what can be done to prevent and take effective measure to ensure the safety and health of a driver or a suspect of the accident, this also brings us an understanding of how is it important to inculcate the best way to prevent an accident on whatever the condition is like a natural cause of weather or traffic timings and regions and localities also this brings us to a better way to drive and be safe.

This project can help an organization which can implement and measure better equipment and all the aspects to prevent and help suspect to save his/her life in any minimal or major accident taking place across the Indian region also this can help a government department of road safety to understand and analysis root cause and their solutions to take immediate action plan to suppose the betterment of each citizens road safety.

5 Further Development Or Research

In this project further, there can be analyzed based on all parameters including datasets of weather conditions, road health and safety measures, road traffic, traffic signals and all other geographical situations.

The optimistic approach for the framework can result in an analysis that can help build an integrated solution to prevent major accidents that can be very useful to save lives as it can indicate factors and aspects that can improve the road safety measures and their use in a global level to ensure the better road traffic and management, also this can lead to the betterment of government policies on road safety and also in constructing the roads based on the better placements of every equipment to ensure the better approach and safety as well as precautions.

6 References

[1] A. Priyanka and K. Sathiyakumari, "A comparative study of classification algorithm using accident data", International Journal of Computer Science & Engineering Technology (IJCSET)

[2] Karlaftis M, Tarko A. Heterogeneity considerations in accident modelling. Accid Anal Prev. 1998;30(4):425–33.

[3]Article(online)Available:https://www.hindustantimes.com/mumbai-news/india-had-most-deaths-in-road-accidents-in-2019-report/story-pikRXxsS4hptNVvf6J2g9O.html

[4]Road Traffic Accidents in India: Issues and Challenges, Sanjay Kumar Singh https://www.sciencedirect.com/science/article/pii/S2352146517307913

[5] Ng KS, Hung WT, Wong WG. An algorithm for assessing the risk of traffic accidents. J Saf Res. 2002;33:387–410.

[6] T. Dipo, I Akomolafe and Akinbola Olutayo, "Using Data Mining Technique to Predict Cause of Accident and Accident Prone Locations on Highways", American Journal of Database Theory and Application 2012

[7] Article by S. Nagendra Babu “A Data Mining Framework to Analyze Road Accident Data using Map Reduce Methods CCMF and TCAMP Algorithms”