ROAD ACCIDENT DATA ANALYSIS

Project Report submitted in Partial
fulfillment of the requirement for the
award of Degree of

B. Tech. Computer Science and Engineering

Submitted by

Japmann Kaur Banga	**20BRS1048**
Aman Kumar Singh	**20BCE1022**

Under the guidance of
Dr. Brindha

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

November 2022

BONAFIDE CERTIFICATE

Certified that this project report titled " **Road Accident Data Analysis**" is the bonafide work of "**Aman Kumar Singh, Japmann Kaur Banga**" who carried out the project work under my supervision in the partial fulfillment of the requirements for the award of the B. Tech Computer Science and Engineering degree.

Dr. Brindha

DECLARATION BY THE STUDENT

I, **Aman Kumar Singh** bearing Reg. No **20BCE1022, Japmann Kaur Banga** bearing Reg No **20BRS1048** hereby declare that this project report entitled (**Road Accident Data Analysis**) has been prepared by me towards the partial fulfillment of the requirement for the award of the B. Tech Computer Science and Engineering (B. Tech) Degree under the guidance of **Dr. Brindha.**

I also declare that this project report is my original work and has not been previously submitted for the award of any Degree, Diploma, Fellowship, or other similar titles.

Japmann Kaur

Aman

Japmann Kaur Banga

20BRS1048

Aman Kumar Singh

20BCE1022

Place: Vellore Institute of Technology, Chennai

Date: 10/11/2022

CONTENTS

ABSTRACT

The current rate of population growth is a major concern and is on the rise. Traffic is heavy during peak hours and can lead to traffic accidents. Many fatalities are due to reckless driving, annoying reflections from other vehicles, etc. Such accidents cause loss of life and property. This white paper describes multilevel models such as predictive and classification algorithms for analyzing accident severity. In addition, this will help minimize losses and comply with established road safety measures. Prediction algorithms are used to predict the occurrence of traffic accidents, and classification algorithms are used to classify the severity of traffic accidents into fatal, serious, and minor injuries.

The paper reviews road traffic accident data analysis and visualization in the R programming environment. One of the key objectives in accident data analysis to identify the main factors associated with a road and traffic accident. The aim is to show how to extract meaningful data from the raw database and visualize it.

The results revealed the hour wise, day wise, month wise and year wise plots which allowed us to observe how road traffic accidents change in timescale. Visualization and data analysis of road traffic accidents led to conclusions which would assist in reducing the number of accidents.

OBJECTIVES/SCOPE OF STUDY

The rapid and unplanned process of urbanization has caused an unprecedented revolution in global automotive growth. The alarming increase in morbidity and mortality from road traffic accidents (RTI) in recent decades has become a major concern worldwide. Motor vehicle crashes currently rank sixth in the list of burdens of illness, and by 2025 he is projected to be third. Across India in 2021, he will kill more than 1.55,000 people in road accidents, according to data from the National Crime Records Service. That's an average of 426 deaths per day and 18 per hour, the highest death toll ever recorded.

The growing number of road and traffic accidents poses a challenge to transportation systems. It is not only concerned with health issues, but also with the economic burden on society. As a result, it is critical for the safety analysis to conduct a comparative study of road accidents in order to identify the factors that cause an accident to occur, so that preventive measures can be implemented to reduce the accident rate and severity of accident consequences. To identify the various factors associated with road accidents, we will be coming up with a comparative study of road accidents by incorporsting the R language for data manipulation and graph based clustering algorithms in Python for data visualisation and identifying influential factors.

PROBLEM STATEMENT

Road accidents are one of the most important factors influencing untimely death and economic loss of public and private property. Road safety is a term that refers to the planning and implementation of specific strategies to prevent road and traffic accidents. Road accident data analysis is a critical tool for identifying various factors associated with traffic accidents and can aid in lowering the accident rate.

Accurate analysis is required to respond to the sheer number of traffic accidents in one place. This analysis is performed in more detail to determine the severity of traffic accidents using supervised learning techniques such as machine learning algorithms. Accident severity is categorized as fatality, serious injury, minor injury, and motor accident.

CONTRIBUTIONS

Our contribution to the project is, we tried to take many different types of datasets for this project, like weather-caused accident datasets, time-based accident datasets, fault type-based accident datasets, and vehicle-type collision datasets. And we tried to merge all these different datasets to derive many conclusions and we derived so many conclusions like at what time of period, most accidents occur, what type of vehicles are causing or have more contribution in causing the road accidents. We also applied a neural network algorithm to predict the severity of accidents in weather conditions and our trained model is giving the output with good accuracy.

From past surveys, it is found that, due to urbanization, the standard of living of people has upgraded, and hence has boosted an increase in both population and vehicle. Every person owns one or more than one vehicles nowadays, which has increased. This has led to the increased commotion on the roads. Transportation is a fundamental part of our lives, as every person needs a vehicle at some point in the day. People going to schools, workplaces, recreational or shopping places need vehicles.

We are aiming to identify critical parameters such as weather conditions, light conditions, which are typically important factors in finding accident severity. We have analyzed several datasets collected from different resources and the performance of different ML algorithms. In this study, we have discussed the severity of accidents which will assist the researchers to work in road safety measures and minimizing casualties.

INTRODUCTION TO DOMAIN

Heterogeneity in road accident data is highly undesirable and unavoidable. The major disadvantage of heterogeneity of road accident data is that certain relationships may remain hidden such as certain accident factors associated with particular vehicle type may not be significant in entire data set the enormity of the effect of certain accident related factors may be different for various conditions severity levels for an accident contributing factors may be different for different accident types. In order to get more accurate results this heterogeneity of road accident data must be removed to deal with this heterogeneous nature of road accident data, divide the data into groups based on some exogenous attributes e.g. accident location, road condition, cause of accident and analyzed every group separately to identify several influential factors associated with road accidents in each group. However, this method is unrealistic as grouping the data based on certain attributes may result in less important groups. Some subgroups can have large numbers of samples and some can have very low numbers of samples thus restricting the application of accident severity models.

Data analysis is the process of examining, cleaning, transforming, and modeling data with the goal of discovering useful information, drawing conclusions, and supporting decision-making. There are multiple aspects and approaches to data analysis, including different techniques with different names and used in different fields of business, science and social sciences. In today's business world, data analytics play a role in helping organizations make more scientific decisions and operate more effectively.

LITERATURE REVIEW

A study [1] used Hadoop to analyze large amounts of data based on various criteria for predicting traffic accidents. Compared to other methods, Hadoop has proven to be the most efficient method for analyzing big data. The algorithms used in this document are CCMF and TCAMP, which effectively analyze datasets to predict traffic accident risk. The proposed algorithm attempts to predict the risk of traffic accidents using vast amounts of data on vehicle behavior and conditions favorable to traffic accidents.

This paper [2] focuses on finding and predicting traffic accident patterns based on severity, road type, accident type, climate, accident time, etc. The method of finding interesting and useful patterns from spatial databases is called spatial data mining. The spatio-temporal algorithm finds hidden patterns easier than traditional data mining techniques.
Spatial data is data that contains locations on the surface of the earth. Two spatio-temporal clustering algorithms are used to find patterns. DBSCAN or density-based spatial clustering algorithms and grid-based algorithms for noisy applications. On the other hand, DBSCAN data must have data points and constrained thresholds, whereas grid-based clustering algorithms require multidimensional data structures. Grid-based algorithms such as STING and CLIQUE have been found to provide more accurate results with faster processing times.

In [3], traffic accidents are inferred from heterogeneous data. A vast amount of heterogeneous data, including accident data and GPS recordings, was collected to investigate how vehicle mobility affects traffic accidents. This data is analyzed to build a Stack Denoise Auto-Encoder model that examines human locomotion characteristics to predict crash risk. The preparedness model can be used to simulate real-time accident risk and warn people of potential accidents to ensure safer routes and travel.

Anupama McCaret. In Al [4] he analyzed an accident data set of the last few years to predict traffic accidents. The approach proposed in this paper involves merging machine learning algorithms such as Bayes net, j48 graft and j48 decision tree in the data mining process, and studies the performance of algorithms in predicting accidents. Therefore, we have found that combining such algorithms yields better results than using a single algorithm. The results obtained help predict traffic accidents and contribute to their prevention and control.

This paper [5] used two predictive models to analyze historical and current accident data to predict the number of accidents that will occur this year. Multiple Linear regression and artificial neural networks were used for predictive analysis. After the analysis is performed, we conclude that the regression model had a larger error in the predicted values. On the other hand, predictions from artificial neural network analysis were more accurate and had fewer errors. Therefore, ANN has proven to be a better method for making his predictions of accidents.

The problem addressed in the paper [6] is to predict the number of accidents that occur on intersections and other roads and to find roads with high accident risk. Algorithms were used on heterogeneous data to determine traffic risk. An algorithm integrated into the framework is Advanced Function-Based Nonnegative Matrix Factorization (FNMF). This framework succeeds in predicting traffic risk on each street and intersection more accurately than existing algorithms and methods. Two clusters were defined to separate the risk locations. One cluster has high risk roads for accidents and another cluster has high risk vehicle collisions.

In this paper [7], high-density accident areas were identified. To this end, we used GIS and the KDE method to investigate spatial patterns of injury-related traffic accidents. It also uses the K-Means clustering method to create a taxonomy of London and UK traffic accident hotspots. Five groups and 15 clusters were created based on collision and attribute data. These clusters have been discussed in road safety campaigns and evaluated in terms of robustness and potential applications.

In this paper [8], they examined traffic accident data from Spain to determine the severity of injuries in urban novice drivers. We used the Information Root Node Variation (IRNV) method (based on a decision tree) to obtain a rule set that provides useful information about the most probable cause of death in accidents involving inexperienced drivers in urban areas. did. This method is based on the decision tree classifier. These rules provide useful knowledge to prevent this type of accident.

In this article [9], the zones, streets and specific times of CDMX where the highest number of his traffic accidents occurred in 2016 were determined. A database has been built that compiles pieces of information from the Waze social network. Using the Knowledge Discovery Database (KDD) methodology, we discovered patterns in incident reports. An expectation-maximization (EM) algorithm was used to obtain an ideal number of clusters for the data, and k-means was used for the clustering procedure.

This paper [10] states that spatial data is related to speckled data. For example, the demographic data break has mottled and gapped hard edges. Comparative studies show that the approaches used with geospatial data ecologically exploited these attributes.

SUMMARY:

S.No.	Title	Models Used	Summary	Authors
1.	A Data Mining Framework to Analyze Road Accident Data using Map Reduce Methods CCMF and TCAMP Algorithms	CCFM and TCAMP	The authors have used Hadoop to analyze the large data based on various criteria to predict road accidents, in order to raise precaution alarms for the same. On comparing Hadoop with other methods, it has proved to be the most efficient method for big data analysis.	S. Nagendra Babu, J. Jebamalar Tamilselvi
2.	ACCIDENT PREDICTION BASED ON ACCIDENT TYPES USING SPATIO TEMPORAL CLUSTERING ALGORITHMS	DBSCAN, STING and CLIQUE	This paper focuses on finding and predicting road accident patterns based on the severity, road type, accident type, climate, hour of accident etc. The method of finding interesting and useful patterns from spatial databases is termed as spatial data mining. Spatiotemporal algorithms are able to locate hidden patterns more easily than traditional data mining techniques.	Dara Anitha Kumari, Dr. A. Govardhan
3.	Learning Deep Representation from Big and Heterogeneous Data for Traffic Accident Inference	Stack denoise autoencoder model	Huge amount of heterogeneous data containing accident data and GPS records have been collected, to check how vehicle mobility affects the road accidents. On analyzing this data a Stack denoise autoencoder model is prepared which studies the features in human mobility to predict accident risks	Quanjun Chen, Xuan Song, Harutoshi Yamada, Ryosuke Shibasaki
4.	A Radical Approach to Forecast the Road Accident Using Data Mining Technique	Bayes Net, j48 graft and j48 decision tree	It has been thus noticed that the combination of such algorithms render better results than a single algorithm used. The results obtained would support in forecasting road traffic accidents and hence prevention and control can be provided.	Anupama Makkar, Harpreet Singh Gill

5.	Study of Road Accident Prediction Model at Accident Blackspot Area: A Case Study at Selangor	Multiple Linear Regression , Artificial Neural Networks	After conducting the analysis it is concluded that the predicted values from the regression model had greater errors. While the predictions made from Artificial Neural Network analysis were more accurate having less errors. Hence ANN was proved to be a better methodology to make predictions for accidents.	Haikal Aiman Hartika, Mohd Zakwan Ramli, Muhamad Zaihafiz Zainal Abidin, Mohd Hafiz Zawawi
6.	Traffic Risk Mining From Heterogeneous Road Statistics	advanced feature based non-negative matrix factorization (FNMF).	This framework is successful in predicting the traffic risks at any road or intersection more accurately than the existing algorithms or methods. Two clusters were defined that segregated the risk locations, in which one cluster had larger roads with accident risks, and other clusters having higher risk of vehicle collision. Risky locations were ranked based on the results of the clusters.	Koichi Moriya, Shin Matsushima, Kenji Yamanishi
7.	Kernel density estimation and Kmeans clustering to profile road accident hotspots	Geographical Information System, Kernel Density Estimation, K-means Clustering	In this paper the high-density areas of accidents have been identified. For this GIS and KDE methodologies have been used to study the spatial patterns of injury related road accidents. K Means clustering methodology is used for creating a classification of road accident hotspots in London and the UK. Five groups and 15 clusters were created based on collision and attribute data. These clusters are discussed and evaluated according to their robustness and potential uses in road safety campaigning.	Tessa K. Anderson
8.	Decision Tree Ensemble Method for Analyzing Traffic Accidents of Novice Drivers in Urban Areas	Decision Tree	In this paper they have surveyed the Spanish road accident data and found the injury severity involving novice drivers in urban areas. The information root node variation (IRNV) method (based on decision trees) was used to get a rule set that provides useful	Serafín MoralGarcía , Javier G. Castellano , Carlos J. Mantas , Alfonso Montella and

			information about the most probable causes of fatalities in accidents involving inexperienced drivers in urban areas. This method is based on the decision tree classifier. These rules provide useful knowledge in order to prevent these kinds of accidents.	Joaquín Abellán
9.	Road Traffic Accidents Analysis in Mexico City through Crowdsourcing Data and Data Mining Techniques	KDD, EM, KMeans	In this article the zones, roads and specific time in the CDMX in which the largest number of road traffic accidents are concentrated during 2016 has been identified. A database compiling information obtained from the social network known as Waze is built. The methodology Discovery of knowledge in the database (KDD) for the discovery of patterns in the accident reports was used. The Maximization of Expectations (EM) algorithm was used to obtain the number ideal of clusters for the data and k-means was used for the grouping method.	Gabriela V. Angeles Perez, Jose Castillejos Lopez, Araceli L. Reyes Cabello, Emilio Bravo Grajales, Adriana Perez Espinosa, Jose L. Quiroz Fabian
10.	Twenty years and counting with ADIE: Spatial Analysis by Distance Indices software and review of its adoption and use	Ia, index of aggregation, Patch and gap cluster indices and Red blue Plot	It stated that the spatial data deals with the patchy data. For example, in population data the destruction is having hard edges which are patchy and have gaps in it. The comparative studies say that the approach used in spatially referenced data has ecologically utilized these attributes.	Linton Winder1, Colin Alexander2, Georgianne Griffiths3, John Holland4, Chris Woolley5, Joe Perry6

EXISTING METHODOLOGIES AND LIMITATIONS SUMMARY

The seriousness of traffic accidents have become a major problem worldwide, especially in developing countries. Understanding the major contributing factors can help address the severity of traffic accidents. There are various methods existing to estimate the contributing factors.

A hybrid approach of K-Means and Random Forest (RF) was developed to obtain the most important traffic accident variables. K-means extracts hidden information from traffic accident data and creates new functions in the training set. The distance between each cluster and the connecting line of k1 and k9 is calculated and chosen as the k-max value. k is the optimal value for splitting the training set. RF is used for severity prediction classification of accidents.

Other methodologies include multiple logistic regression and the pattern recognition type of artificial neural network (ANN) as a machine learning solution are used to recognize the most influential variables on the severity of accidents and the superior approach for accident prediction.

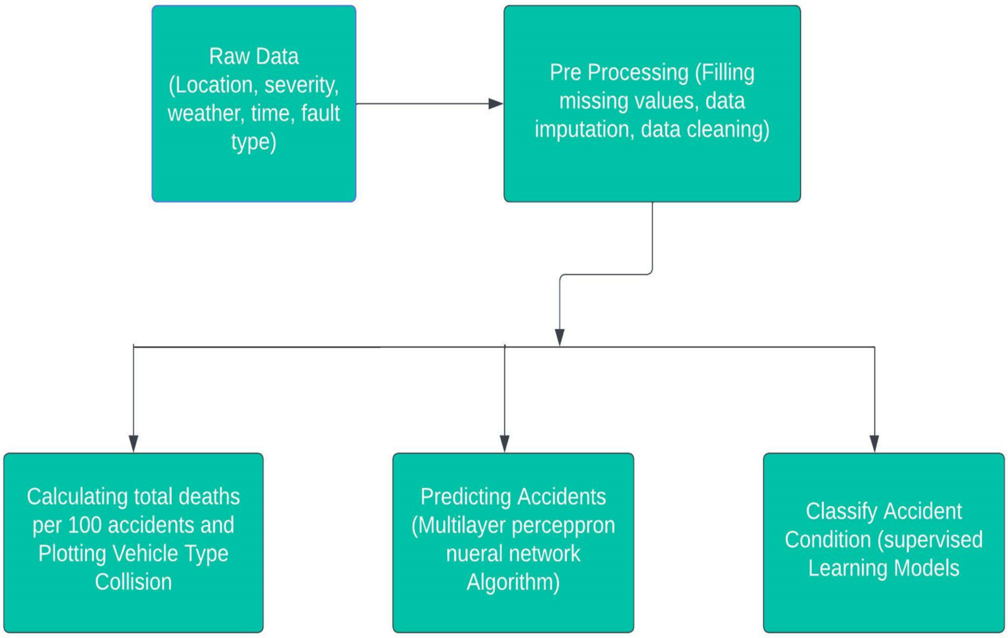
Limitations:

- The First limitation being the lack of availability of data-sets for Indian Road Accidents. There are no data-sets for Indian Road Accidents whereas we can find numerous data sets for countries like the USA, the UK, etc. If the data sets are found then one can easily train the data and can help in reducing fatalities in road accidents.
- Most current accident analysis techniques are event-based and do not adequately capture the dynamic complexity and non-linear interactions that characterize accidents in complex systems.
- Estimation methods vary from country to country. The composition of different vehicle fleets can lead to bias in mortality estimates and comparisons. Significantly higher than in Europe, which is misleading, as the risk of death while riding a motorcycle or moped is much higher than while driving a car.

PROPOSED PROJECT WORK IN DETAIL

1. Workflow

The following image describes the proposed framework that is being followed in the project.


Raw Data (Location, severity, weather, time, fault type)
Pre Processing (Filling missing values, data imputation, data cleaning)
Calculating total deaths per 100 accidents and Plotting Vehicle Type Collision
Predicting Accidents (Multilayer perceppron nueral network Algorithm)
Classify Accident Condition (supervised Learning Models

In the work first we preprocessed the data then we found the total deaths per 100 accidents, rush hour from data visualization and finally classified the accident severity using supervised learning algorithms based on the weather conditions.

First we extracted 4 datasets from different sites of India with data from 2019 to 2021, in R we used the "xlsx" package to extract the data in the RStudio and in python we used panda library to extract data in Google Colab.

After extracting the raw data we pre processed the data, in preprocessing we did data cleaning in R using MICE and in python, we replaced the na value with zero and in the timestamps data, we extracted the year, day, hour, month information from the whole time data and added it to the dataset. After that by using our fault type data, we visualized the data by making a pie chart of all types of vehicle collision fault type and we got to know driver's fault is the main cause in the road accident analysis.

In the time series dataset, we analyzed and visualized that the period of 3pm-6pm is the deadliest rush hour period in road data accidents. Like this we visualized weather causes and vehicle collision type in the data visualization part. After that we tried to classify the accidents caused by weather conditions and trained our model with 89% accuracy using python keras.

2. Modules/Algorithm Phases

A. Data Pre-Processing:

In the data pre processing step, we analyze the data and find the missing values, inconsistency in the dataset and outliers etc. In R we used the `dplyr` package to know about NA values and missing values. For the imputation of data, in R we used the “mice” package. Python's numpy class is used to handle missing data. Unnecessary columns can lead to deviations from correct predictions, so some columns are discarded to refine the data set for better predictions. In this way, the features required to perform the predictive function are selected. Categorical data is also encoded into numerical data for computation and fitting in machine learning models. This process uses the LabelEncoder class from scikit learn.

Code:

For dataset in R:

```
install.packages("dplyr")
library("dplyr")
(is.na(Mydata)).sum()
datamiss = data2[is.na(Mydata)]
datamiss
install.packages("mice")
library(mice)
dataframeData = as.data.frame(datamiss)
dataframeData
md.pattern(dataframeData)
mice_imputes=mice(dataframeData,m=7,method="pmm")
```

For Dataset in Google Colab:

```
import pandas as pd
import numpy as np
import sklearn
df=pd.read_csv("C:/Users/kaurb/Downloads/Mydata4.csv", parse_dates =
['Start_Time','End_Time'])
df.head()
df.columns
df.isnull().sum()
df1=df
from datetime import datetime
db = pd.DataFrame()
db["year"] = df1['Start_Time'].dt.year
db["month"] = df1['Start_Time'].dt.month
db["day"] = df1['Start_Time'].dt.day
db["hour"] = df1['Start_Time'].dt.hour
df1 = df1.drop('Start_Time', axis=1)
df1=pd.concat([db,df1], axis=1)
df1.columns
```

```
df1.isnull().sum()
df1.shape
df1=df1.dropna(how='any', subset=['City', 'Zipcode',
'Weather_Condition', 'Sunrise_Sunset', 'Civil_Twilight',
'Wind_Direction', 'Nautical_Twilight', 'Astronomical_Twilight'])
df1.shape
df1.isnull().sum()
from sklearn.linear_model import LinearRegression
model=LinearRegression()
df1.year=df1.year.astype('float64')
df1.month=df1.month.astype('float64')
df1.day=df1.day.astype('float64')
df1.hour=df1.hour.astype('float64')
d=df1
def regression_fill(df1, y):
 print("doing for ", y)
 df2=df1[df1[y].isnull()]
 df1=df1.dropna(subset=[y])
 model.fit(df1[['year', 'month', 'day', 'hour', 'Start_Lat',
'Start_Lng']], df1[y])
 Y=model.predict(df2[['year', 'month', 'day', 'hour', 'Start_Lat',
'Start_Lng']])
 df2=df2.drop([y], axis=1)
 df2[y]=Y
 df1=pd.concat([df1, df2])
 return df1
targets=['Temperature(F)', 'Wind_Chill(F)', 'Humidity(%)',
'Pressure(in)',
'Visibility(mi)', 'Wind_Speed(mph)']
for t in targets:
 df1=regression_fill(df1, t)
df1['Precipitation(in)'].fillna(0, inplace=True)
df1.isnull().sum()
df1.dtypes
from sklearn.preprocessing import LabelEncoder
encoder=LabelEncoder()
for c in df1.columns:
 if df1[c].dtype=='object' or df1[c].dtype=='bool':
 df1[c]=encoder.fit_transform(df1[c]).astype('int32')
df1.head()
df1.dtypes
df1=df1.sort_values(by=['ID']).reset_index(drop=True)
df1.shape
df1.head()
df1.to_csv("Mydata6.csv")
```

B. Total Deaths per 100 Accidents and Plotting Vehicle Type Collision:

We find this metric is to analyze the severity of accidents and how many people are surviving per 100 accidents in India. And by plotting Vehicle Type Collisions, we can get to know what type of vehicle is causing the highest number of accidents and particularly for each vehicle type we can try to conclude the result.

Code:

Total Deaths per 100 Accidents:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.pylab import rcParams
rcParams['figure.figsize'] = 15, 6
dataAcc = pd.read_csv("Mydata1.csv")
dataAcc.head(n=10)
totalaccidents = {}

totalaccidents["2018"] = dataAcc.loc[36,'Number of Total Road
Accidents of Two-Wheelers - 2018']
totalaccidents["2018"] += dataAcc.loc[36,'Number of Total
Road Accidents of Auto-Rickshaws - 2018']
totalaccidents["2018"] += dataAcc.loc[36,'Number of Total
Road Accidents of Cars, Jeeps, Taxis - 2018']
totalaccidents["2018"] += dataAcc.loc[36,'Number of Total
Road Accidents of Buses - 2018']
totalaccidents["2018"] += dataAcc.loc[36,'Number of Total
Road Accidents of Trucks, Tempos, MAVs, Tractors - 2018']
totalaccidents["2018"] += dataAcc.loc[36,'Number of Total
Road Accidents of Other Motor Vehicles - 2018']
totalaccidents["2018"] += dataAcc.loc[36,'Number of Total
Road Accidents of Other Vehicles/Objects - 2018']

totalaccidents["2019"] = dataAcc.loc[36,'Two-Wheelers -
Number of Road Accidents-Total - 2019']
totalaccidents["2019"] += dataAcc.loc[36,'Auto-Rickshaws -
Number of Road Accidents-Total - 2019']
totalaccidents["2019"] += dataAcc.loc[36,'Cars, Jeeps,Taxis -
Number of Road Accidents - Total - 2019']
totalaccidents["2019"] += dataAcc.loc[36,'Buses - Number of
Road Accidents - Total - 2019']
totalaccidents["2019"] += dataAcc.loc[36,'Trucks,
Tempos,MAVs,Tractors - Number of Road Accidents - Total -
2019']
totalaccidents["2019"] += dataAcc.loc[36,'Other Motor
Vehicles - Number of Road Accidents - Total - 2019']
```

```
totalaccidents["2019"] += dataAcc.loc[36,'Other
Vehicles/Objects - Number of Road Accidents - Total - 2019']

totalaccidents["2020"] = dataAcc.loc[36,'Motor Cycle/ Scooter
- Number of Road Accidents - Total - 2020']
totalaccidents["2020"] += dataAcc.loc[36,'Moped/Scootty -
Number of Road Accidents - Total - 2020']
totalaccidents["2020"] += dataAcc.loc[36,'Auto rickshaw -
Number of Road Accidents - Total - 2020']
totalaccidents["2020"] += dataAcc.loc[36,'Tempo - Number of
Road Accidents - Total - 2020']
totalaccidents["2020"] += dataAcc.loc[36,'E-Rickshaw - Number
of Road Accidents - Total - 2020']
totalaccidents["2020"] += dataAcc.loc[36,'Motor Car - Number
of Road Accidents - Total - 2020']
totalaccidents["2020"] += dataAcc.loc[36,'Jeep - Number of
Road Accidents - Total - 2020']
totalaccidents["2020"] += dataAcc.loc[36,'Taxi - Number of
Road Accidents - Total - 2020']
totalaccidents["2020"] += dataAcc.loc[36,'Bus - Number of
Road Accidents - Total - 2020']
totalaccidents["2020"] += dataAcc.loc[36,'Truck/Lorry -
Number of Road Accidents - Total - 2020']
totalaccidents["2020"] += dataAcc.loc[36,'Articulated
Vehicle/Trolly - Number of Road Accidents - Fatal - 2020']
totalaccidents["2020"] += dataAcc.loc[36,'Tractor - Number of
Road Accidents - Total - 2020']
totalaccidents["2020"] += dataAcc.loc[36,'Other Motor
Vehicles - Number of Road Accidents - Total - 2020']
totalkilled = {}

totalkilled["2018"] = dataAcc.loc[36,'Number of Persons
Killed from accidents of Two-Wheelers - 2018']
totalkilled["2018"] += dataAcc.loc[36,'Number of Persons
Killed from accidents of Auto-Rickshaws - 2018']
totalkilled["2018"] += dataAcc.loc[36,'Number of Persons
Killed from accidents of Cars, Jeeps, Taxis - 2018']
totalkilled["2018"] += dataAcc.loc[36,'Number of Persons
Killed from accidents of Buses - 2018']
totalkilled["2018"] += dataAcc.loc[36,'Number of Persons
Killed from accidents of Trucks, Tempos, MAVs, Tractors -
2018']
totalkilled["2018"] += dataAcc.loc[36,'Number of Persons
Killed from accidents of Other Motor Vehicles - 2018']
totalkilled["2018"] += dataAcc.loc[36,'Number of Persons
Killed from accidents of Other Vehicles/Objects - 2018']
```

```
totalkilled["2019"] = dataAcc.loc[36,'Two-Wheelers - Number
of Persons-Killed - 2019']
totalkilled["2019"] += dataAcc.loc[36,'Auto-Rickshaws -
Number of Persons-Killed - 2019']
totalkilled["2019"] += dataAcc.loc[36,'Cars, Jeeps,Taxis -
Number of Persons Killed - 2019']
totalkilled["2019"] += dataAcc.loc[36,'Buses - Number of
Persons - Killed - 2019']
totalkilled["2019"] +=
dataAcc.loc[36,'Trucks,Tempos,MAVs,Tractors - Number of
Persons - Killed - 2019']
totalkilled["2019"] += dataAcc.loc[36,'Other Motor Vehicles -
Number of Persons - Killed - 2019']
totalkilled["2019"] += dataAcc.loc[36,'Other Vehicles/Objects
- Number of Persons - Killed - 2019']

totalkilled["2020"] = dataAcc.loc[36,'Motor Cycle/ Scooter -
Number of Persons - Killed - 2020']
totalkilled["2020"] += dataAcc.loc[36,'Moped/Scootty - Number
of Persons - Killed - 2020']
totalkilled["2020"] += dataAcc.loc[36,'Auto rickshaw - Number
of Persons - Killed - 2020']
totalkilled["2020"] += dataAcc.loc[36,'Tempo - Number of
Persons - Killed - 2020']
totalkilled["2020"] += dataAcc.loc[36,'E-Rickshaw - Number of
Persons - Killed - 2020']
totalkilled["2020"] += dataAcc.loc[36,'Motor Car - Number of
Persons - Killed - 2020']
totalkilled["2020"] += dataAcc.loc[36,'Jeep - Number of
Persons - Killed - 2020']
totalkilled["2020"] += dataAcc.loc[36,'Taxi - Number of
Persons - Killed - 2020']
totalkilled["2020"] += dataAcc.loc[36,'Bus - Number of
Persons - Killed - 2020']
totalkilled["2020"] += dataAcc.loc[36,'Truck/Lorry - Number
of Persons - Killed - 2020']
totalkilled["2020"] += dataAcc.loc[36,'Articulated
Vehicle/Trolly - Number of Persons - Killed - 2020']
totalkilled["2020"] += dataAcc.loc[36,'Tractor - Number of
Persons - Killed - 2020']
totalkilled["2020"] += dataAcc.loc[36,'Other Motor Vehicles -
Number of Persons - Killed - 2020']
severity = {}
for i in totalkilled:
    severity[i] = (totalkilled[i]/totalaccidents[i])*100

plt.figure(figsize=(10,5))
plt.plot([2018,2019,2020],list(severity.values()))
plt.xticks([2018,2019,2020])
plt.yticks([28,29,30,31])
plt.title('Accident Severity Index (Total deaths per 100
accidents)')
plt.xlabel('Year')
plt.ylabel('Accident Severity Index')
plt.show()
```

Vehicle Type Collision:

```
vehicletype = {}
vehicletype["2-Wheeler"] =dataAcc.loc[36,'Motor Cycle/
Scooter - Number of Road Accidents - Total - 2020']
vehicletype["2-Wheeler"] +=dataAcc.loc[36,'Moped/Scootty -
Number of Road Accidents - Total - 2020']

vehicletype["3-Wheeler"] =dataAcc.loc[36,'Auto rickshaw -
Number of Road Accidents - Total - 2020']
vehicletype["3-Wheeler"] +=dataAcc.loc[36,'Tempo - Number of
Road Accidents - Total - 2020']
vehicletype["3-Wheeler"] +=dataAcc.loc[36,'E-Rickshaw -
Number of Road Accidents - Total - 2020']


vehicletype["4-Wheeler"] =dataAcc.loc[36,'Motor Car - Number
of Road Accidents - Total - 2020']
#plot vehicle-type

plt.figure(figsize=(10,8))
plt.pie(list(vehicletype.values()),labels=list(vehicletype.ke
ys()),autopct='%1.2f%%')
plt.axis('equal')
plt.xlabel('Types of Vehicles Involved in Accidents in 2020')

centre_circle = plt.Circle((0,0),0.70,fc='white')
fig = plt.gcf()
fig.gca().add_artist(centre_circle)

plt.show()
vehicletype2 = {}
vehicletype2["2-Wheeler"] = dataAcc.loc[36,'Number of Persons
Killed from accidents of Two-Wheelers - 2018']
vehicletype2["3-Wheeler"]  = dataAcc.loc[36,'Number of
Persons Killed from accidents of Auto-Rickshaws - 2018']
vehicletype2["4-Wheeler"] = dataAcc.loc[36,'Number of Persons
Killed from accidents of Cars, Jeeps, Taxis - 2018']
vehicletype2["Heavy Vehicle"] = dataAcc.loc[36,'Number of
Persons Killed from accidents of Buses - 2018']
```

```
vehicletype2["Heavy Vehicle"] += dataAcc.loc[36,'Number of
Persons Killed from accidents of Trucks, Tempos, MAVs,
Tractors - 2018']
vehicletype2["Other Vehicle"] = dataAcc.loc[36,'Number of
Persons Killed from accidents of Other Motor Vehicles -
2018']
vehicletype2["Other Vehicle"] += dataAcc.loc[36,'Number of
Persons Killed from accidents of Other Vehicles/Objects -
2018']
plt.figure(figsize=(10,8))
plt.pie(list(vehicletype2.values()),labels=list(vehicletype2.
keys()),autopct='%1.2f%%')
plt.axis('equal')
plt.xlabel('Types of Vehicles Involved in Accidents in 2018')

centre_circle = plt.Circle((0,0),0.70,fc='white')
fig = plt.gcf()
fig.gca().add_artist(centre_circle)

plt.show()

vehicletype["4-Wheeler"] +=dataAcc.loc[36,'Jeep - Number of
Road Accidents - Total - 2020']
vehicletype["4-Wheeler"] +=dataAcc.loc[36,'Taxi - Number of
Road Accidents - Total - 2020']

vehicletype["Heavy Vehicle"] =dataAcc.loc[36,'Bus - Number of
Road Accidents - Total - 2020']
vehicletype["Heavy Vehicle"] +=dataAcc.loc[36,'Truck/Lorry -
Number of Road Accidents - Total - 2020']
vehicletype["Heavy Vehicle"] +=dataAcc.loc[36,'Articulated
Vehicle/Trolly - Number of Road Accidents - Total - 2020']
vehicletype["Heavy Vehicle"] +=dataAcc.loc[36,'Tractor -
Number of Road Accidents - Total - 2020']

vehicletype["Other Vehicle"] =dataAcc.loc[36,'Other Motor
Vehicles - Number of Road Accidents - Total - 2020']
```

C. Finding Rush Hours (Data Visualization):

Rush hour is when accidents are most likely to occur. Rush hour was localized using visualization techniques. I used the seaborn and matplotlib libraries to visualize the data and identify peak traffic times. The bar chart used the number of accidents that occurred and the time in 24-hour terms to find rush hours. The results obtained in this way are satisfactory and accurate.

Code:

```
import pandas as pd
import seaborn as sns
Mydata2 = pd.read_excel("Accident_Dataset.xlsx")
Mydata2.to_csv ("Mydata2.csv", index = None,header=True)
df = pd.read_csv("Mydata2.csv")
df.head(n=10)
X = df.iloc[:,3:].values
X_n = df[['LGA_NAME','TOTAL_PERSONS']]
X_n = X_n.iloc[:,:].values
import matplotlib.pyplot as plt
plt.scatter(X_n[:,0],X_n[:,1])
plt.xlabel('LGA_NAME')
plt.ylabel('Toatal_Accidents')
plt.show()
time = df['ACCIDENT_TIME']
j=0
t=[]
for i in time:
   t.append(int(i[0:2]))
   j+=1

import math
j=0
for i in t:
    t[j] = i//3
    j+=1

deathpertime={}
for i,j in zip(X_n[:,1],t):
    if j in deathpertime:
        deathpertime[j] += i
    else:
        deathpertime[j] = i
val=[]
for i in sorted(deathpertime):
    val.append(deathpertime[i])
```

```
label =
['12am-3am','3am-6am','6am-9am','9am-12pm','12pm-3pm','3pm-6p
m','6pm-9pm','9pm-12am']
plt.figure(figsize=(10,5))
plt.bar([0,1,2,3,4,5,6,7],val)
plt.xticks([0,1,2,3,4,5,6,7],label)
plt.xlabel('Time (3-hour period)')
plt.ylabel('Number of accidents')
plt.show()
plt.pie(val,labels=label,autopct='%1.1f%%')
plt.axis('equal')
plt.xlabel('Percentage of Accidents per 3-hour period')
plt.show()
```

Other Plots:

```
import numpy as np
X1 = df.iloc[:,:].values
X1 = np.delete(X1,2,axis=1)
X1 = np.delete(X1,1,axis=1)
X1 = np.delete(X1,0,axis=1)
names = (df.columns.values)
names = names[3:]
sns.countplot(x='LIGHT_CONDITION', hue='SEVERITY',data=df)
plt.show()
sns.countplot(x='SEVERITY',hue='ALCOHOLTIME',data=df)
plt.show()
sns.countplot(x='LIGHT_CONDITION',hue='ALCOHOLTIME',data=df)
plt.show()
sns.countplot(x='SEVERITY',hue='SPEED_ZONE',data=df)
plt.show()
sns.countplot(x='POLICE_ATTEND',hue='SPEED_ZONE',data=df)
plt.show()
sns.countplot(x='FEMALES',hue='SEVERITY',data=df)
plt.show()
```

D. Classify the accident severity based on weather conditions:

For classification, we used the Perceptron multilayer neural network with supervised learning algorithm. The network consists of three fully connected layers with activation function "Sigmoid". For the first layer, i. H. The input layer uses 25 neurons. Second, i. The H. hidden layer uses 50 neurons and the last layer i.H. is the output layer and consists of only five layers. For optimization, we used the SGD (Stochastic Gradient Descent) optimizer. Model has been trained for 10 epochs with a lot size of 32.

Code:

```
import pandas as pd
import numpy as np
import sklearn
from keras.optimizers import SGD, RMSprop, Adadelta, Adagrad,
Adam, Adamax
from keras.models import Sequential
from keras.layers.core import Dense, Activation, Flatten
import keras
from sklearn.model_selection import train_test_split
from keras.utils import to_categorical
df=pd.read_csv('Mydata5.csv')
df.head()
df.columns
data=df[['Astronomical_Twilight', 'Civil_Twilight',
'Humidity(%)', 'Nautical_Twilight', 'Precipitation(in)',
'Pressure(in)', 'Sunrise_Sunset', 'Temperature(F)',
'Visibility(mi)', 'Weather_Condition', 'Wind_Chill(F)',
'Wind_Direction', 'Wind_Speed(mph)', 'Severity']].values
x=data[:, 0: 13]
y=data[:, 13]
xtrain, xtest, ytrain, ytest=train_test_split(x, y,
test_size=.2)
ytrain = to_categorical(ytrain.astype('float32'))
model = Sequential()
model.add(Dense(25, input_dim=13, activation= "sigmoid"))
model.add(Dense(50, activation= "sigmoid"))
model.add(Dense(5, activation="sigmoid"))
model.summary()
opt = SGD()
model.compile(loss='categorical_crossentropy', optimizer=opt,
metrics=['accuracy'])
model.fit(xtrain, ytrain, epochs=10, batch_size=32)
model.save("train2.model")
ytest = to_categorical(ytest.astype('float32'))
score, acc = model.evaluate(xtest, ytest)
print(acc*100)
```

E. Fault Type Based Data Visualization:

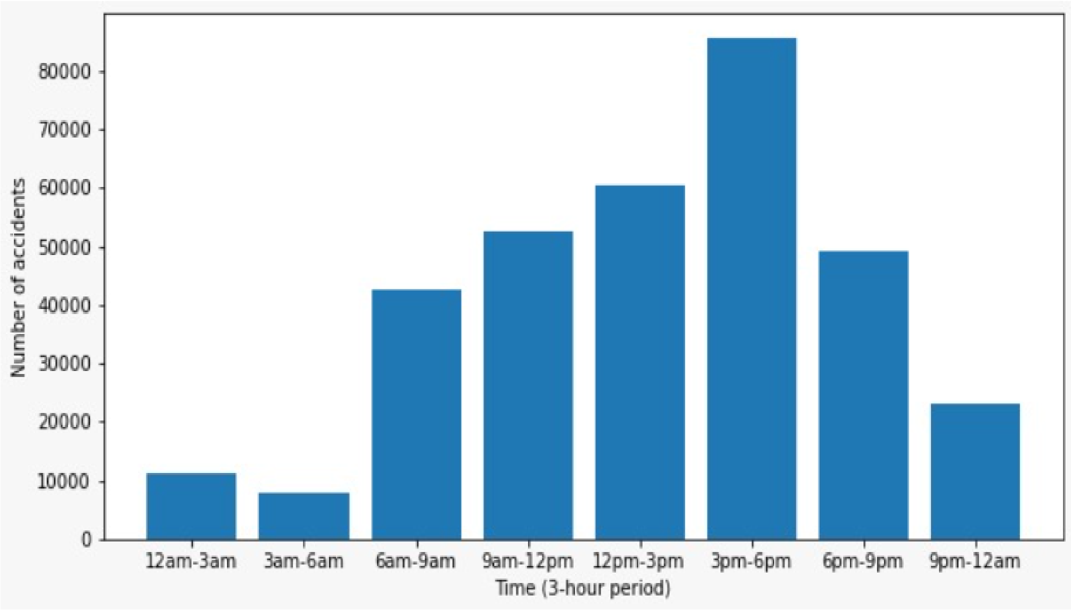
Code:

```
import pandas as pd
import matplotlib.pyplot as plt
FaultType = pd.read_csv("Mydata3.csv")
FaultType = FaultType.drop(FaultType.index[[34,37]])
```

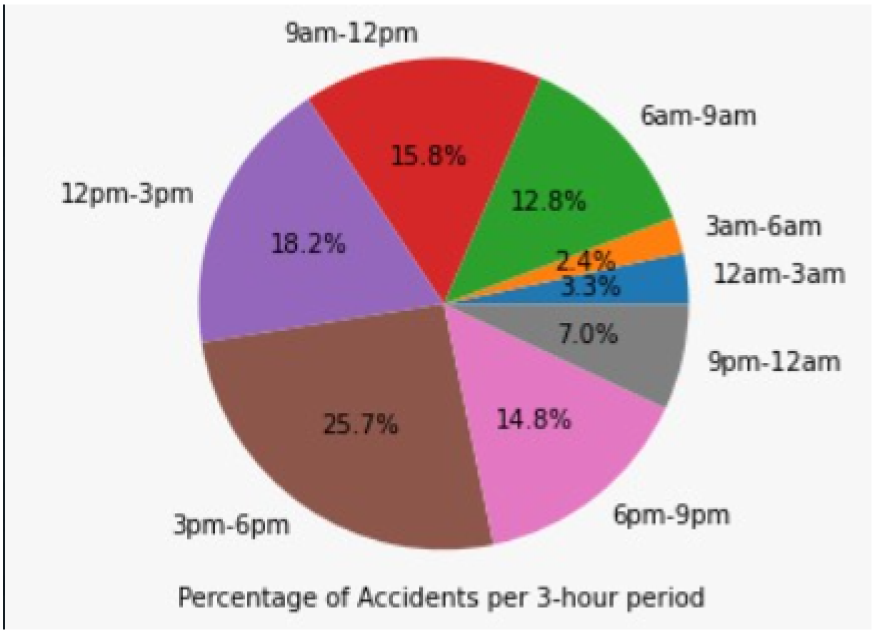
```
faulttype = {}
faulttype["Driver's Fault"] = FaultType.loc[36,'Fault of
Driver-Total No. of Road Accidents - 2019']
faulttype["Cyclist's Fault"] = FaultType.loc[36,'Fault of
Cyclist-Total No. of Road Accidents - 2019']
faulttype["Vehicle Condition"] = FaultType.loc[36,'Defect in
Condition of Motor Vehicle-Total No. of Road Accidents -
2019']
faulttype["Road Condition"] = FaultType.loc[36,'Defect in
Road Condition-Total No. of Road Accidents - 2019']
faulttype["Weather Condition"] = FaultType.loc[36,'Weather
Condition-Total No. of Road Accidents - 2019']
faulttype["Passenger's Fault"] = FaultType.loc[36,'Fault of
Passenger-Total No. of Road Accidents - 2019']
faulttype["Poor Light"] = FaultType.loc[36,'Poor light-Total
No. of Road Accidents - 2019']
faulttype["Stray Animals"] = FaultType.loc[36,'Stray
animals-Total No. of Road Accidents - 2019']
faulttype["Others"] = FaultType.loc[36,'Other causes/ Causes
not known-Total No. of Road Accidents - 2019']
val = list(faulttype.values())
total = sum(val)
for i in range(0,9):
    val[i] = format(val[i]*100/total,'.2f')
plt.figure(figsize=(10,8))
plt.pie(list(faulttype.values()))
plt.axis('equal')
plt.xlabel('Accidents in 2019')

centre_circle = plt.Circle((0,0),0.70,fc='white')
fig = plt.gcf()
fig.gca().add_artist(centre_circle)
label = [list(pair) for pair in
zip(list(faulttype.keys()),val)]
plt.legend(label)
plt.show()
```

RESULTS

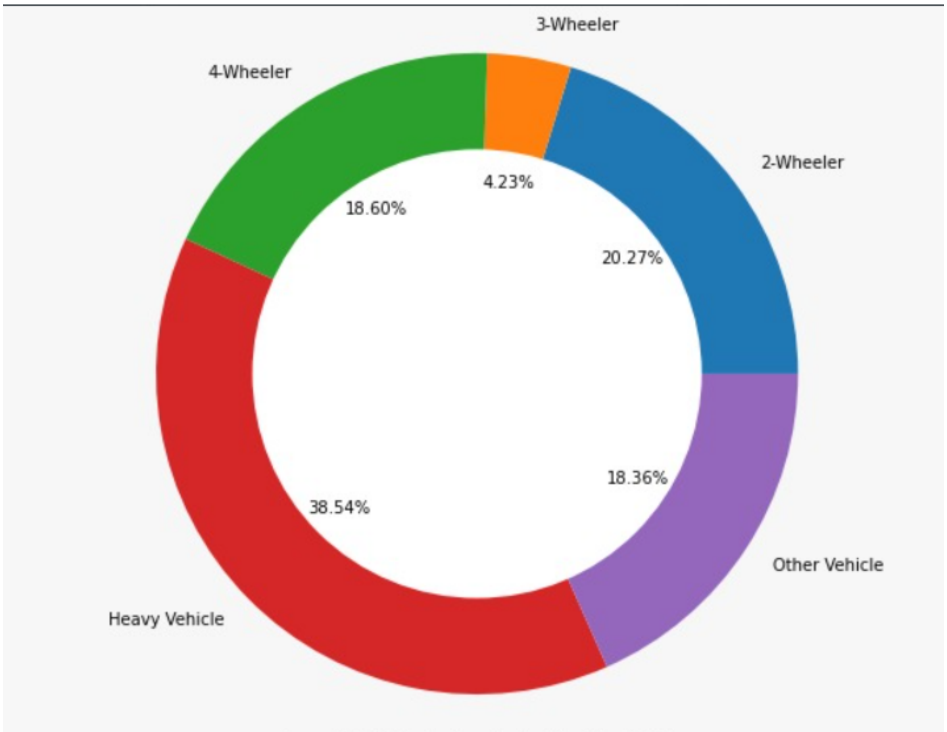
1. Finding Rush Hours:



This image is obtained from the data visualizations. So from here we can see that most of the accidents occur between 12 pm to 3 pm in the noon and 3pm to 6pm in the evening. So we can conclude that the noon 12-3 hours and the evening 15-18 hours are the rush hours for accidents.

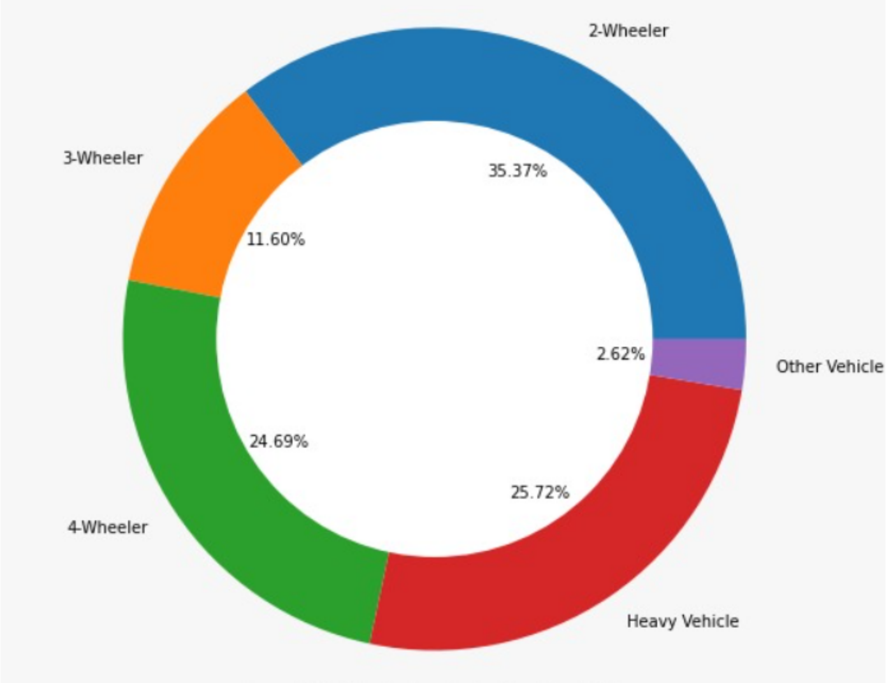


2. Vehicle Type Collision:



Types of Vehicles Involved in Accidents in 2018

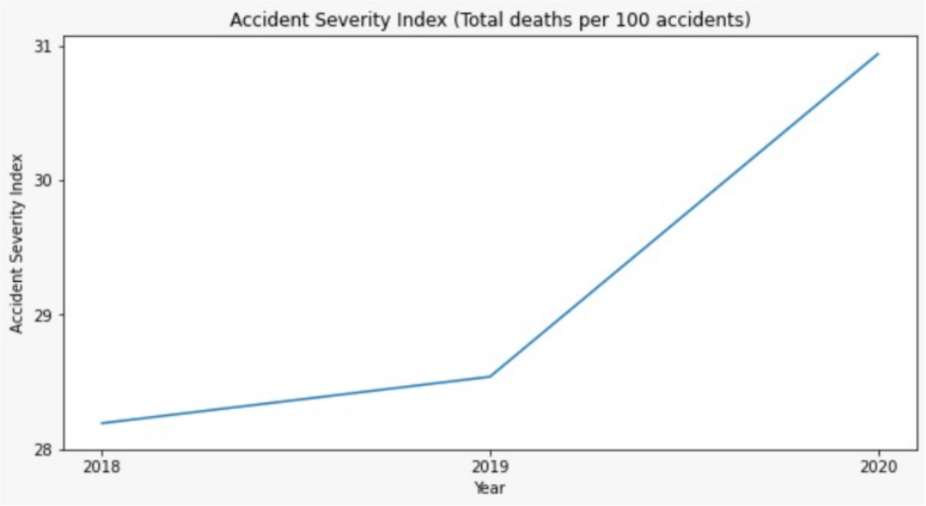
The image is obtained from data visualization. By this result image it is clear that in the year of 2018 the most number of accidents were caused by `Heavy Vehicle` and in the second place the most number of accidents were caused by `2-Wheeler` vehicles.



Types of Vehicles Involved in Accidents in 2020

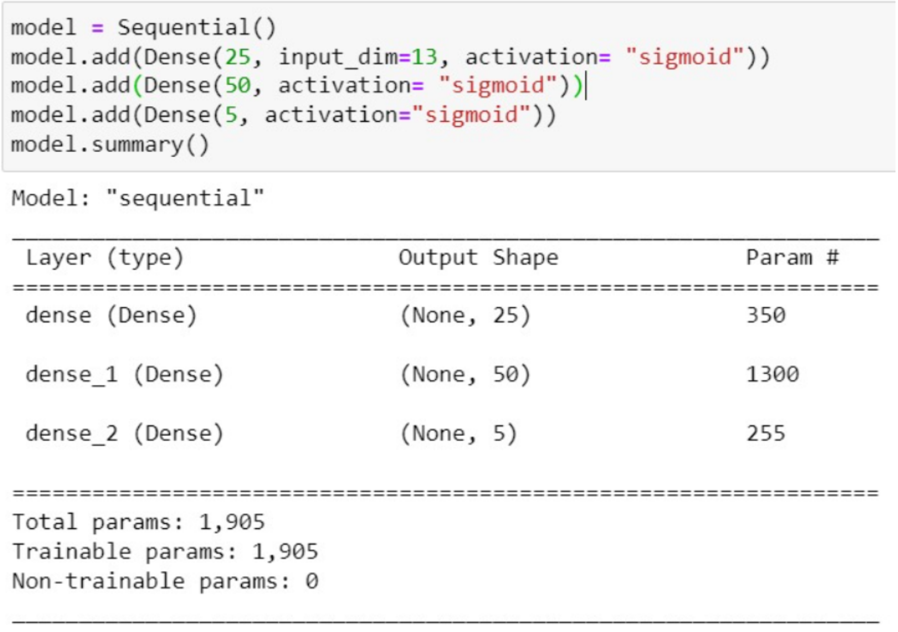
The image is obtained from data visualization. By this result image it is clear that in the year of 2020 the most number of accidents were caused by `2-Wheeler` and in the second place the most number of accidents were caused by `Heavy vehicles` but there is a new entry in this list, `4-Wheeler`.

3. Total Deaths per 100 Accidents:



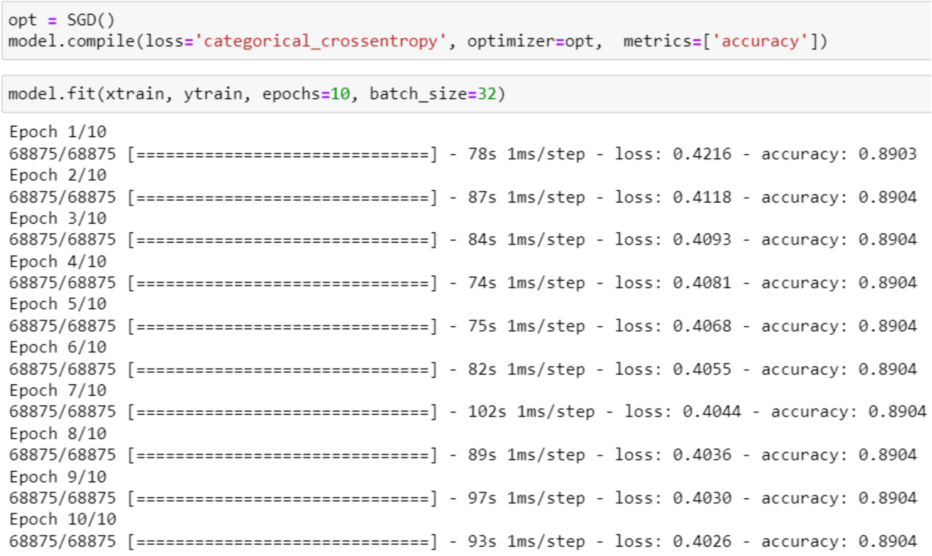
This image is also obtained from data visualization. In this we can see a huge spike in deaths in the year of 2020 per 100 accidents. So we can conclude that in the comparison of 2018 and 2019, there were more deaths in 2020.

4. Multilayer Perceptron Neural Network Algorithm:



```
model = Sequential()
model.add(Dense(25, input_dim=13, activation= "sigmoid"))
model.add(Dense(50, activation= "sigmoid"))
model.add(Dense(5, activation="sigmoid"))
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 25)                350

 dense_1 (Dense)             (None, 50)                1300

 dense_2 (Dense)             (None, 5)                 255

=================================================================
Total params: 1,905
Trainable params: 1,905
Non-trainable params: 0
_________________________________________________________________
```

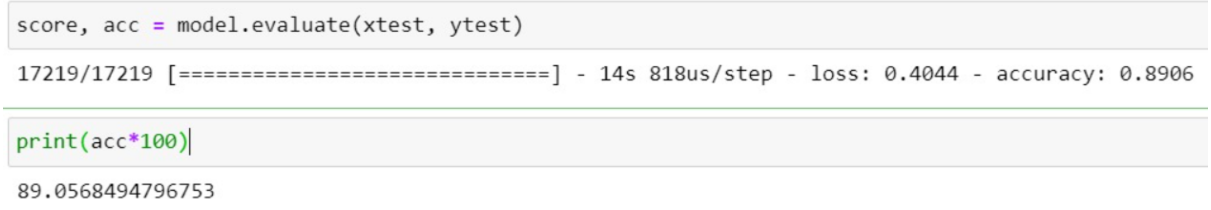


```
opt = SGD()
model.compile(loss='categorical_crossentropy', optimizer=opt,  metrics=['accuracy'])
```

```
model.fit(xtrain, ytrain, epochs=10, batch_size=32)
```

```
Epoch 1/10
68875/68875 [==============================] - 78s 1ms/step - loss: 0.4216 - accuracy: 0.8903
Epoch 2/10
68875/68875 [==============================] - 87s 1ms/step - loss: 0.4118 - accuracy: 0.8904
Epoch 3/10
68875/68875 [==============================] - 84s 1ms/step - loss: 0.4093 - accuracy: 0.8904
Epoch 4/10
68875/68875 [==============================] - 74s 1ms/step - loss: 0.4081 - accuracy: 0.8904
Epoch 5/10
68875/68875 [==============================] - 75s 1ms/step - loss: 0.4068 - accuracy: 0.8904
Epoch 6/10
68875/68875 [==============================] - 82s 1ms/step - loss: 0.4055 - accuracy: 0.8904
Epoch 7/10
68875/68875 [==============================] - 102s 1ms/step - loss: 0.4044 - accuracy: 0.8904
Epoch 8/10
68875/68875 [==============================] - 89s 1ms/step - loss: 0.4036 - accuracy: 0.8904
Epoch 9/10
68875/68875 [==============================] - 97s 1ms/step - loss: 0.4030 - accuracy: 0.8904
Epoch 10/10
68875/68875 [==============================] - 93s 1ms/step - loss: 0.4026 - accuracy: 0.8904
```

For the classification purpose we have used supervised learning algorithm multilayer perceptron neural networks. The network is made of 3 fully connected layers with activation function “sigmoid”. For the first layer i.e. the input layer 25 neurons are used; in the second i.e. in the hidden layer 50 neurons are used and in the final layer i.e. the output layer consists of only 5 neurons. For optimizing we have used SGD (Stochastic Gradient Descent) Optimizer. The model is trained for 10 epochs with a batch size of 32.

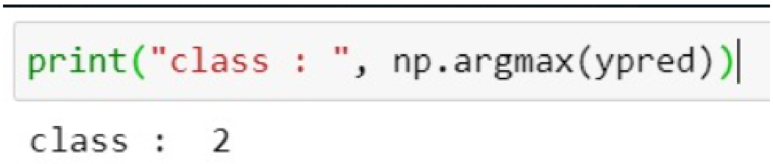


```
score, acc = model.evaluate(xtest, ytest)
```

```
17219/17219 [==============================] - 14s 818us/step - loss: 0.4044 - accuracy: 0.8906
```

```
print(acc*100)
```

```
89.0568494796753
```



```
print("class : ", np.argmax(ypred))
```

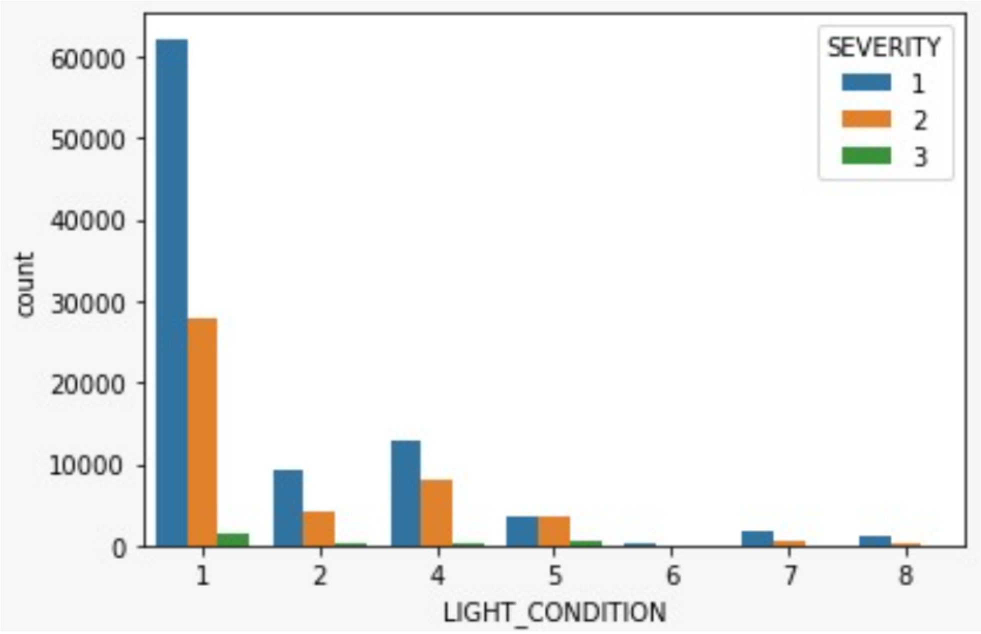
```
class :  2
```

In this we are getting almost 89% accuracy on predicting the severity of the accident based on the various weather conditions. So the model is giving good predictions

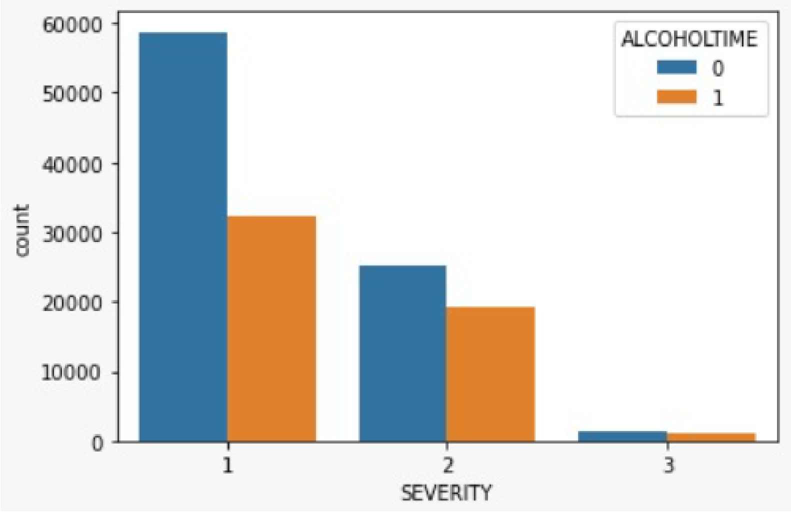
5. Others:

```
import numpy as np
X1 = df.iloc[:,:].values
X1 = np.delete(X1,2,axis=1)
X1 = np.delete(X1,1,axis=1)
X1 = np.delete(X1,0,axis=1)
names = (df.columns.values)
names = names[3:]
```

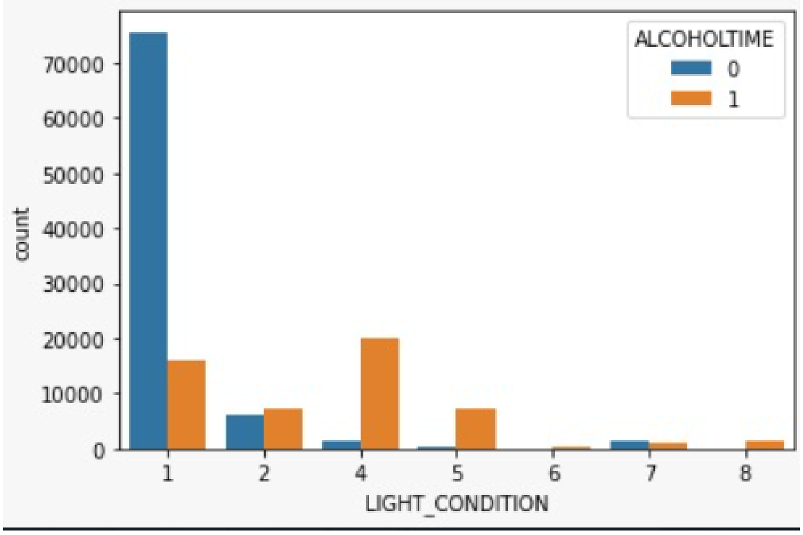
1.
```
sns.countplot(x='LIGHT_CONDITION',
hue='SEVERITY',data=df)
plt.show()
```



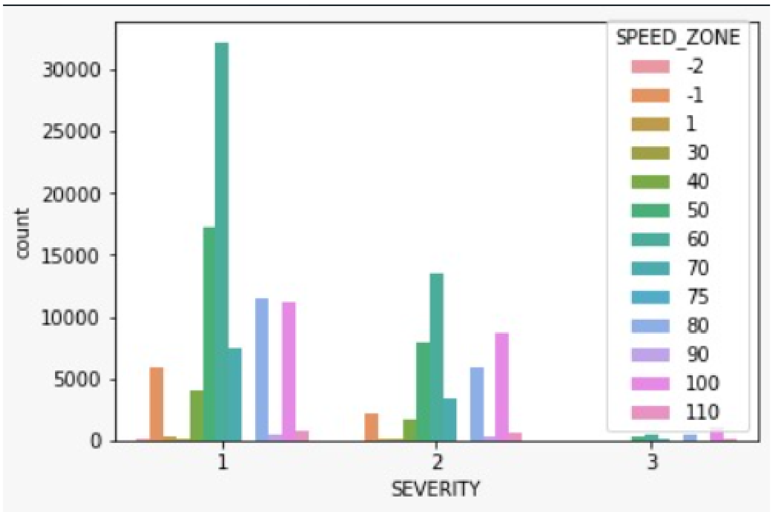
2.
```
sns.countplot(x='SEVERITY',hue='ALCOHOLTIME',data=df)
plt.show()
```



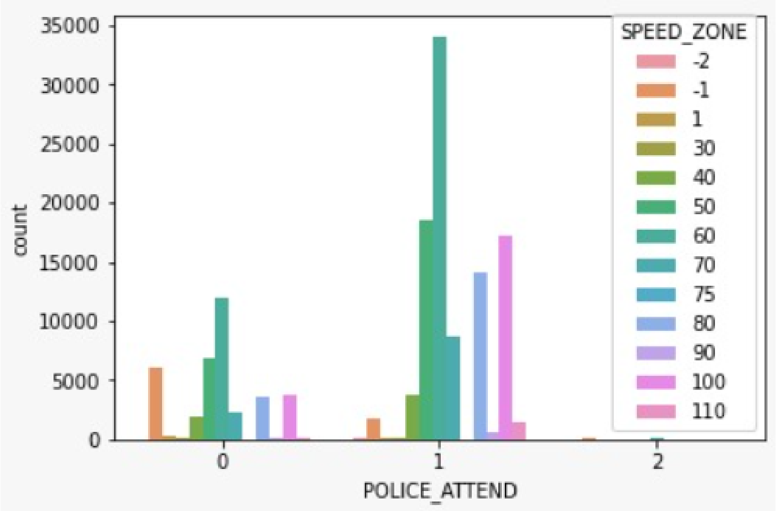
3.
```
sns.countplot(x='LIGHT_CONDITION',hue='ALCOHOLTIME',data
=df)
plt.show()
```



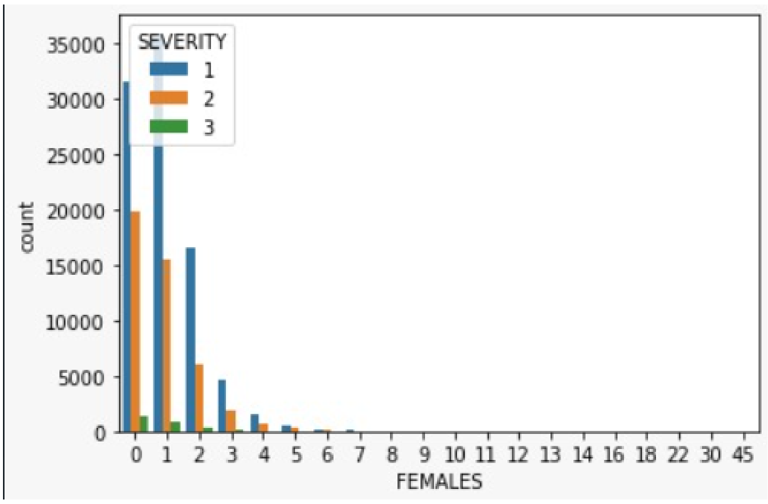
4.
```
sns.countplot(x='SEVERITY',hue='SPEED_ZONE',data=df)
plt.show()
```



5.
```
sns.countplot(x='POLICE_ATTEND',hue='SPEED_ZONE',data=df
)
plt.show()
```

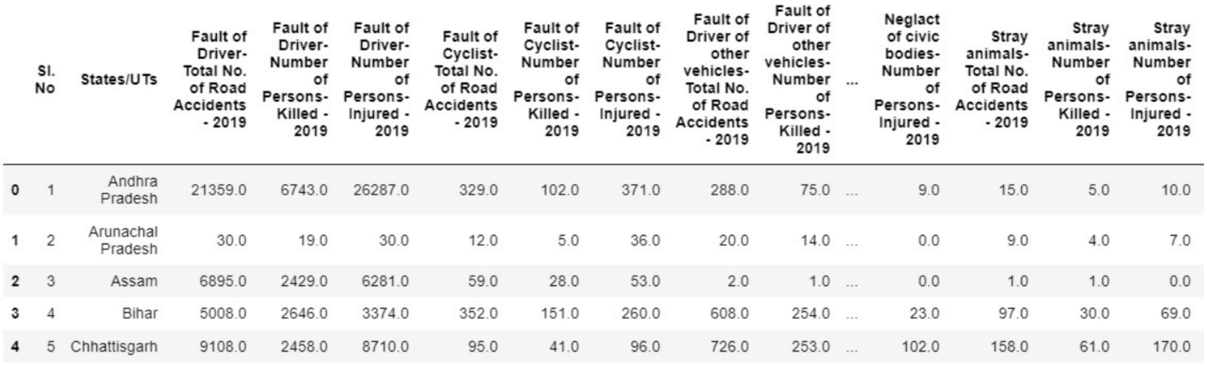


6.
```
sns.countplot(x='FEMALES',hue='SEVERITY',data=df)
plt.show()
```



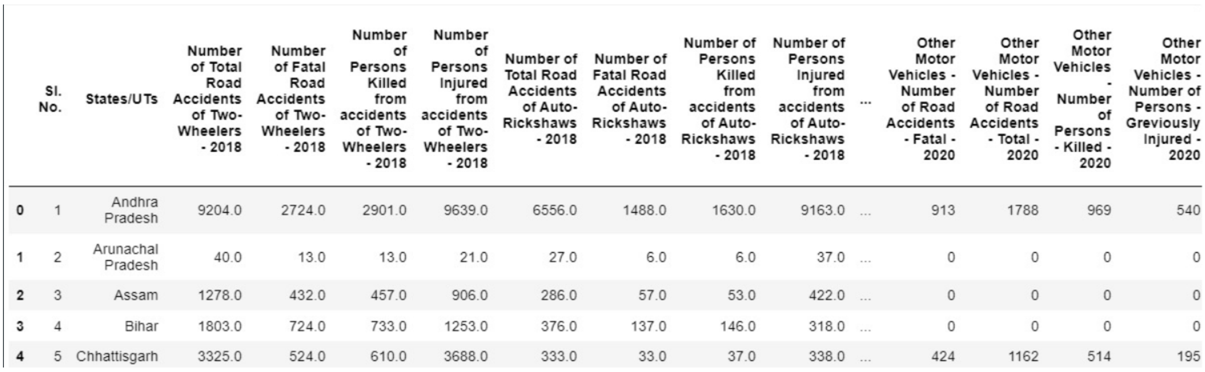
TABLES:

Fault Type



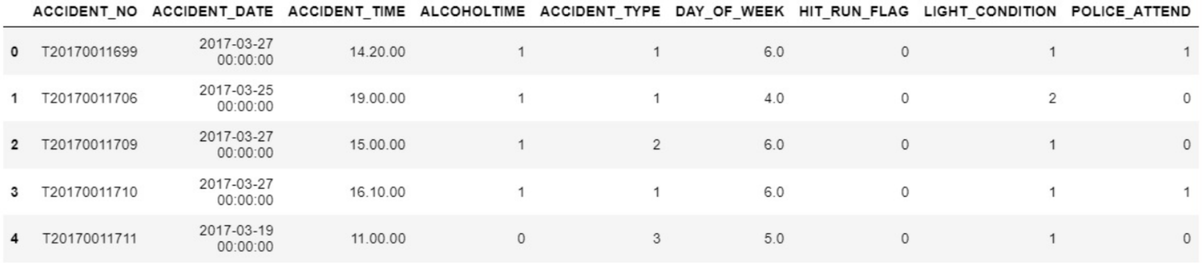
	Sl. No	States/UTs	Fault of Driver-Total No. of Road Accidents - 2019	Fault of Driver-Number of Persons-Killed - 2019	Fault of Driver-Number of Persons-Injured - 2019	Fault of Cyclist-Total No. of Road Accidents - 2019	Fault of Cyclist-Number of Persons-Killed - 2019	Fault of Cyclist-Number of Persons-Injured - 2019	Fault of Driver of other vehicles-Total No. of Road Accidents - 2019	Fault of Driver of other vehicles-Number of Persons-Killed - 2019	...	Neglact of civic bodies-Number of Persons-Injured - 2019	Stray animals-Total No. of Road Accidents - 2019	Stray animals-Number of Persons-Killed - 2019	Stray animals-Number of Persons-Injured - 2019
0	1	Andhra Pradesh	21359.0	6743.0	26287.0	329.0	102.0	371.0	288.0	75.0	...	9.0	15.0	5.0	10.0
1	2	Arunachal Pradesh	30.0	19.0	30.0	12.0	5.0	36.0	20.0	14.0	...	0.0	9.0	4.0	7.0
2	3	Assam	6895.0	2429.0	6281.0	59.0	28.0	53.0	2.0	1.0	...	0.0	1.0	1.0	0.0
3	4	Bihar	5008.0	2646.0	3374.0	352.0	151.0	260.0	608.0	254.0	...	23.0	97.0	30.0	69.0
4	5	Chhattisgarh	9108.0	2458.0	8710.0	95.0	41.0	96.0	726.0	253.0	...	102.0	158.0	61.0	170.0

Accidents



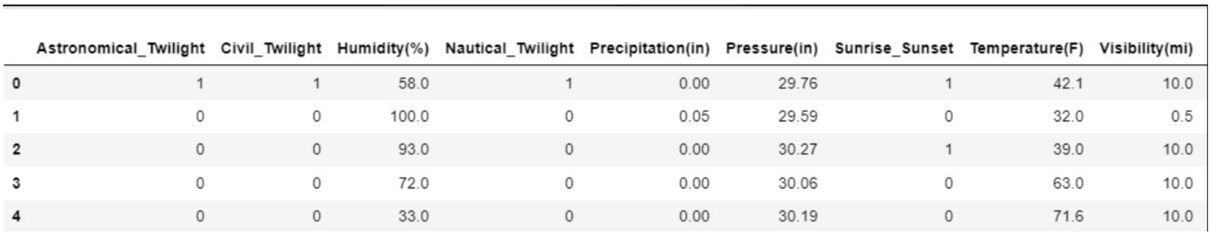
	Sl. No.	States/UTs	Number of Total Road Accidents of Two-Wheelers - 2018	Number of Fatal Road Accidents of Two-Wheelers - 2018	Number of Persons Killed from accidents of Two-Wheelers - 2018	Number of Persons Injured from accidents of Two-Wheelers - 2018	Number of Total Road Accidents of Auto-Rickshaws - 2018	Number of Fatal Road Accidents of Auto-Rickshaws - 2018	Number of Persons Killed from accidents of Auto-Rickshaws - 2018	Number of Persons Injured from accidents of Auto-Rickshaws - 2018	...	Other Motor Vehicles - Number of Road Accidents - Fatal - 2020	Other Motor Vehicles - Number of Road Accidents - Total - 2020	Other Motor Vehicles - Number of Persons - Killed - 2020	Other Motor Vehicles - Number of Persons - Greviously Injured - 2020
0	1	Andhra Pradesh	9204.0	2724.0	2901.0	9639.0	6556.0	1488.0	1630.0	9163.0	...	913	1788	969	540
1	2	Arunachal Pradesh	40.0	13.0	13.0	21.0	27.0	6.0	6.0	37.0	...	0	0	0	0
2	3	Assam	1278.0	432.0	457.0	906.0	286.0	57.0	53.0	422.0	...	0	0	0	0
3	4	Bihar	1803.0	724.0	733.0	1253.0	376.0	137.0	146.0	318.0	...	0	0	0	0
4	5	Chhattisgarh	3325.0	524.0	610.0	3688.0	333.0	33.0	37.0	338.0	...	424	1162	514	195

Time Based

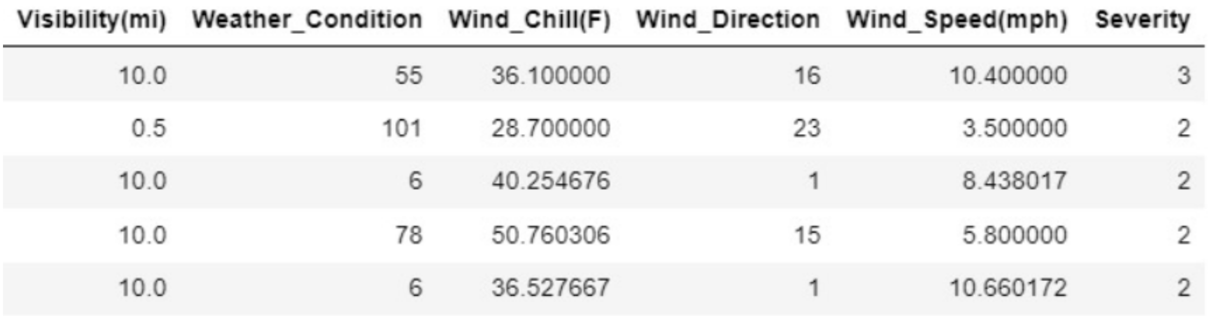


	ACCIDENT_NO	ACCIDENT_DATE	ACCIDENT_TIME	ALCOHOLTIME	ACCIDENT_TYPE	DAY_OF_WEEK	HIT_RUN_FLAG	LIGHT_CONDITION	POLICE_ATTEND
0	T20170011699	2017-03-27 00:00:00	14.20.00	1	1	6.0	0	1	1
1	T20170011706	2017-03-25 00:00:00	19.00.00	1	1	4.0	0	2	0
2	T20170011709	2017-03-27 00:00:00	15.00.00	1	2	6.0	0	1	0
3	T20170011710	2017-03-27 00:00:00	16.10.00	1	1	6.0	0	1	1
4	T20170011711	2017-03-19 00:00:00	11.00.00	0	3	5.0	0	1	0

Weather Condition & Predicted Accident Severity



	Astronomical_Twilight	Civil_Twilight	Humidity(%)	Nautical_Twilight	Precipitation(in)	Pressure(in)	Sunrise_Sunset	Temperature(F)	Visibility(mi)
0	1	1	58.0	1	0.00	29.76	1	42.1	10.0
1	0	0	100.0	0	0.05	29.59	0	32.0	0.5
2	0	0	93.0	0	0.00	30.27	1	39.0	10.0
3	0	0	72.0	0	0.00	30.06	0	63.0	10.0
4	0	0	33.0	0	0.00	30.19	0	71.6	10.0



Visibility(mi)	Weather_Condition	Wind_Chill(F)	Wind_Direction	Wind_Speed(mph)	Severity
10.0	55	36.100000	16	10.400000	3
0.5	101	28.700000	23	3.500000	2
10.0	6	40.254676	1	8.438017	2
10.0	78	50.760306	15	5.800000	2
10.0	6	36.527667	1	10.660172	2

CONCLUSION

So, we can conclude that the rush hours are the times when people come back from the offices and the kids, children come back from the students and the traffic on the road is maximum. Most of the time the heavy vehicles like buses, trucks and the most useful vehicles in daily life like 2 wheelers are the main causes of the road accidents.

From our dataset we also concluded that most of the time (85% times in 2018) it's the driver's fault in a road accident. Also, the severity classification deep learning model is working well with a good accuracy. So, by our work we can predict whether there will be an accident or not and if the accident occurs how severe it may be. So necessary measures should be taken to prevent these accidents.

SCOPE FOR FUTURE WORK

The future plan for this project is to connect an IoT device to a deep learning model using a raspberry pi that takes weather condition input from the environment and feeds it to the deep learning model. You can take steps to avoid accidents.
Therefore, we developed an app that can efficiently predict traffic accidents based on the above factors. There are many reasons to choose neural networks, including: Ability to extract information from incomplete and noisy data. Gain experience and knowledge through self-education and organization. Possibility of very fast optimization. Your aptitude for problems for which algorithmic solutions are difficult or non-existent.

REFERENCES

[1] Athanasios Theofilatos and George, Yannis National "A review of the effect of traffic and weather characteristics on road safety", Accident Analysis and Prevention 72 (2014) 24425. (2014)

[2] 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) Volume2, Issue 7, July 2013.

[3] R.R. Dinu, A. Veeraragavan "Random parameter models for accident prediction on two lane undivided highways in India" , Journal of safetyResearch 42(2011) 39-42, 2011

[4] S. Nagendra Babu, J. Jebamalar Tamilselvi," A Data Mining Framework to Analyze Road Accident Data using Map Reduce Methods CCMF and TCAMP Algorithms" , IJSSST, 2013

[5] Dara Anitha Kumari, Dr. A. Govardhan," ACCIDENT PREDICTION BASED ON ACCIDENT TYPES USING SPATIO TEMPORAL CLUSTERING ALGORITHMS",

[6] Quanjun Chen, Xuan Song, Harutoshi Yamada, Ryosuke Shibasaki," Learning Deep Representation from Big and Heterogeneous Data for Traffic Accident Inference", Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence

[7] Anupama Makkar, Harpreet Singh Gill," A Radical Approach to Forecast the Road Accident Using Data Mining Technique", International Journal of Innovative Science and Research Technology ISSN No: - 2456 – 2165 Volume 2, Issue 8,2017

[8] Haikal Aiman Hartika, Mohd Zakwan Ramli, Muhamad Zaihafiz Zainal Abidin, Mohd Hafiz Zawawi," Study of Road Accident Prediction Model at Accident Blackspot Area: A Case Study at Selangor", International Journal of Scientific Research in Science, Engineering and Technology, 2017

[9] Koichi Moriya, Shin Matsushima, Kenji Yamanishi," Traffic Risk Mining From Heterogeneous Road Statistics", IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, VOL. 19, NO,2018

[10] "Linton Winder1, Colin Alexander2, Georgianne Griffiths3, John Holland4, Chris Woolley5, Joe Perry6" Twenty years and counting with ADIE: Spatial Analysis by Distance Indices software and review of its adoption and use,2019

[11] Tessa K. Anderson, "Kernel density estimation and K-means clustering to profile road accident hotspots", 2008

[12] Serafín Moral-García , Javier G. Castellano , Carlos J. Mantas , Alfonso Montella and Joaquín Abellán , “Decision Tree Ensemble Method for Analyzing Traffic Accidents of Novice Drivers in Urban Areas”, 2019

[13] Gabriela V. Angeles Perez, Jose Castillejos Lopez, Araceli L. Reyes Cabello, Emilio Bravo Grajales, Adriana Perez Espinosa, Jose L. Quiroz Fabian, “Road Traffic Accidents Analysis in Mexico City through Crowdsourcing Data and Data Mining Techniques”, 2018