A Project Report

on

PREDICTION OF TRAFFIC ACCIDENTS BASED ON HUMAN INTERACTION AND MACHINE LEARNING

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

Bachelor of Technology in Computer Science and Engineering

GALGOTIAS UNIVERSITY

(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

Under the supervision of
Mr. HIMANSHU SHARMA
Assistant Professor
Department of Computer Science and Engineering

Submitted By

19SCSE1010218 - ATUL ANAND
19SCSE1010812 - URISHITA PANDITA

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GALGOTIAS UNIVERSITY, GREATER NOIDA, INDIA

DECEMBER - 2021

CANDIDATE'S DECLARATION

I/We hereby certify that the work which is being presented in the thesis/project/dissertation, entitled **"PREDICTION OF TRAFFIC ACCIDENTS BASED ON HUMAN INTERACTION AND MACHINE LEARNING"** in partial fulfillment of the requirements for the award of the Project review submitted in the School of Computing Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the period of month, Year to Month and Year, under the supervision of Name... Designation, Department of Computer Science and Engineering/Computer Application and Information and Science, of School of Computing Science and Engineering , Galgotias University, Greater Noida

The matter presented in the thesis/project/dissertation has not been submitted by me/us for the award of any other degree of this or any other places.

Atul Anand

19SCSE1010218

Urishita Pandita

19SCSE1010812

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Prof. Himanshu Sharma

Assistant Professor/SCSE

Galgotias University

ACKNOWLEDGMENT

I am sincerely thankful to Galgotias University, Greater Noida for providing me with the opportunity to write a research paper on the topic **“PREDICTION OF TRAFFIC ACCIDENTS BASED ON HUMAN INTERACTION AND MACHINE LEARNING”.**

We would like to thank our guide **Mr. Himanshu Sharma** for guiding us in every stage of this project. Without his support, it would have been very difficult for me to prepare such a meaningful and interesting project.

This paper has helped me a lot learn about machine learning and deep learning and I hope it helps people to have a basic understanding of deepfake in pictures and its detection.

Atul Anand – 19SCSE1010218

Urishita Pandita - 19SCSE1010812

ABSTRACT

With the exponentially increasing number of vehicles, road safety is a matter of huge concern. Road accidents kill 1.2 million people every year. In 2017, there have been 2367 accidents with injuries reported in Hyderabad alone. It causes loss of lives and economical damage, due to which is a serious concern which needs to be solved.

We have used Machine Learning algorithms to predict the severity of an accident occurring at a particular location and time. Factors like speed limit, age, weather, vehicle type, light conditions and day of the week have been used as parameters for training the model. We have used the road accident data provided by the government of the UK from 2005-2015. The dataset has 1.2 million records of which 80% is used to train the model and 20% to test it. We have chosen Random Forest for our Machine Learning model as it showed the highest accuracy of 86.86%. User data at a specific time will be used to predict the severity of a road accident at the given location. The severity metrics are 1= Fatal, 2= Serious, 3= Slight.

We have used Machine Learning tools such as Python, Scikit-Learn, Numpy, Matplotlib etc. and Google colab is used. The OpenWeatherMap Api is used to get the weather and light conditions at a particular time based on the location of the user. The TextLocalApi is used to send a sms to the police containing the location coordinates of the user and the accident severity predicted for that location. GeoLocation Api is used to take the GPS coordinates of the user.

We have created a web app for user input and output display and a notification is sent to the police to take preventive measures. The model is trained and tested on Google colab. The front end takes the input from the user and sends it to the backend where the Machine Learning model is deployed. The model will run with the input data and predict the severity of an accident occurring at the respective location of the user.

This model will play an important role in planning and management of traffic and would help us reduce a lot of road accidents in the future.

LIST OF FIGURES

LIST OF TABLES

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

According to the death statistics released by the World Health Organization, the number of traffic accidents occurring annually in the world is alarming. The traffic accidents killed 1.2 million people each year and 50 million people were injured. Approximately 3,300 people were killed and 137,000 people were injured every day. Direct economic losses of 43 billion dollars, the frequent occurrence of traffic accidents directly threaten human life and property safety.

Road accident prediction is one of the most important research areas in traffic safety. The occurrence of road traffic accidents is mainly affected by geometric characteristics of road, traffic flow, characteristics of drivers and environment of road. Many studies have been conducted to predict accident frequencies and analyze the characteristics of traffic accidents, including studies on hazardous location/hot spot identification, accident injury-severities analysis, and accident duration analysis. Some studies focus on the mechanism of accidents. Other factors include weather and light conditions of the road.

Lee et al [1] created a probabilistic demonstrate relating noteworthy crash antecedents to changes in crash potential. Abdel [10] built a past crash expectation show with the coordinated case-control calculated relapse procedure. No particular approach accessible for the activity police to foresee which zone is mischance inclined at a particular time. The activity mishap forecast plays an imperative part within the coordinates arranging and administration of activity, the reason which with much arbitrariness around the activity mishap incorporate a few nonlinear components, such as individuals, car, street, climate and so on. A traditional way of linear analyses cannot reveal the real situation since the noise pollution and amount of data are too little, because the result of prediction cannot be satisfactory. It has a 7.8% lower accuracy than the proposed model.

1.1 Objective

Machine Learning algorithms can process a large number of classification parameters and are able to obtain useful patterns. It can process huge amounts of data efficiently and can be scalable. In computer science and related fields, artificial neural networks are computational models that simulate the central nervous system of the animal (especially the brain), allowing the machine to learn and identify information like the human brain.

1.2 Problem Definition

With the exponentially increasing number of vehicles, road safety is a matter of huge concern.
Road accidents kill 1.2 million people every year.
Road crashes cost $518 billion globally, costing individual countries from 1-2% of their economy. In 2017, there have been 2367 accidents with injuries reported in Hyderabad alone.
Steps are being taken to combat this issue but they have been ineffective.

1.3 Existing System

No specific approach available for the traffic police to predict which area is accident prone at a specific time. The traditional Back propagation network has defects. It has a 17% lower accuracy than the proposed model.
We propose the use of a machine learning technique. Machine learning has the ability to model complex non-linear phenomena.

1.4 Proposed System

An ML powered web app which predicts accidents severity based on the current conditions.
It is trained with 1.6 million accident records over 2005-2015. More data means greater accuracy. The purpose of such a model is to be able to predict which conditions will be more prone to accidents, and therefore take preventive measures.
We will even try to locate more precisely future accidents in order to provide faster care and precaution service. According to the predicted severity, a message will be sent to the traffic police to take preventive measures.

1.5 Organization of Report

To provide a platform i.e a web app for taking user input at a particular time and predict severity of an accident at a location beforehand and take precaution.

● Literature Survey discusses the literature survey of this project which includes an insight into the core part of our project along with the technologies used.
● The System Architecture part deals with the design of our proposed system. The Implementation part deals with the implementation of our system which discusses the algorithms used in building our system.
● The Result section displays our results and discussions through a series of screenshots. The final part talks about the conclusions and the future scope of our project.

CHAPTER 2

LITERATURE SURVEY

A literature survey in a software development process is a most significant part as it shows the various analyses and research made in the field of your interest including substantive findings, as well as theoretical and methodological contributions to a particular topic. It is the foremost vital part of the report because it gives you a course within the zone of your inquire about; it makes a difference in setting up the objectives for the investigation. The reason is to communicate to the reader what information and concepts have been set up on a theme, and what their qualities and shortcomings are. Table 1: Literature Review

S. No.	**Title**	**Author**	**Year**	**Objectives**
1	A Model of Traffic Accident Prediction Based on Convolutional Neural Network	Lu Wenqi Luo Dongyu Yan Menghua	2017	To predict the traffic accident severity by using convolution neural networks.
2	The Traffic Accident Prediction Based on Neural Network	Fu Huilin, Zhou Yucai	2017	Traditional ways of linear analyses cannot reveal the real situation; the result of prediction is not satisfactory. Compares traditional BP networks with its proposed solution.

3	Evolutionary Cross Validation	Thineswaran Gunasegaran Y u- N Cheah	2017	This paper proposes an evolutionary cross validation algorithm for identifying optimal folds in a dataset to improve predictive modeling accuracy
4	On the Selection of Decision Trees in Random Forests	Simon Bernard, Laurent Heutte and Sebastien Adam	2017	This paper presents a study on the Random Forest (RF) family of ensemble methods.

According to the death statistics released by the World Health Organization, the number of traffic accidents occurring annually in the world is alarming. The traffic accidents killed 1.2 million people each year and 50 million people were injured. Approximately 3,300 people were killed and 137,000 people were injured every day. Direct economic losses of 43 billion dollars, the frequent occurrence of traffic accidents directly threaten human life and property safety. Road traffic accident prediction is one of the important research contents of traffic safety. The occurrence of road traffic accidents is mainly affected by geometric characteristics of road, traffic flow, characteristics of drivers and environment of road [1-2]. Many studies have been conducted to predict accident frequencies and analyze the characteristics of traffic accidents, including studies on hazardous location/hot spot identification [3], accident injury-severities analysis [4], and accident duration analysis [5]. Some studies focus on the mechanism of accidents. Karlaftis et al [6] used hierarchical tree-based regression to revisit the relationship between rural road geometric characteristics, accident rates and their prediction. Lee et al [1] create a probabilistic show relating noteworthy crash antecedents to changes in crash potential. Abdel [10] built a past crash expectation show with the coordinated case-control calculated relapse method. In recent years, deep learning as a new machine learning method began to be highly regarded by researchers and business people. The deep learning theory explains the text, images and sounds, which is widely used in the field of text, image and speech recognition, and neural network technology as a highly efficient deep learning technique has been widely used in traffic accident prediction. Compared with the traditional learning structure, deep learning has the ability to model complex non- linear phenomena using distributed and hierarchical feature representation.

2.1 PREDICTION FACTORS

The data comes from the government website www.data.gov.uk. UK police forces collect the accident data using the form called Stats19. The data consists of all kinds of vehicle collisions from 2005 to 2015. Every column of the dataset is in numerical format. A supporting document to understand each numerical category in accidents dataset is provided on the www.data.gov.uk website.

Table 1. *Prediction Factors.*

Day of Week: Numeric: 1 for Sunday, 2 for Monday, and so on.
Latitude and Longitude
Light Conditions: Day, night, street lights or not.
Weather Conditions: Wind, rain, snow, fog.
Vehicle Type: Pedal cycle, Motorcycle, Car
Road Surface Conditions: Wet, snow, ice, flood.
Speed Limit: 60 mph, 70 mph
<u>Output</u>
Accident Severity: 1 = Fatal, 2 = Serious, 3 = Slight

CATEGORY AND MEANING OF WEATHER AND LIGHT CLASSIFICATION FACTORS

Table 3. *Weather Conditions*

code	label
1	Fine no high winds
2	Raining no high winds
3	Snowing no high winds
4	Fine + high winds
5	Raining + high winds
6	Snowing + high winds
7	Fog or mist
8	Other
9	Unknown
-1	Data missing or out of range

Table 4. *Light Conditions*

code	label
1	Daylight
4	Darkness - lights lit
5	Darkness - lights unlit
6	Darkness - no lighting
7	Darkness - lighting unknown
-1	Data missing or out of range

Road Surface Conditions and Gender of Driver and Vehicle Type

Table 5. *Road Conditions*

code	label
1	Dry
2	Wet or damp
3	Snow
4	Frost or ice
5	Flood over 3cm. deep
6	Oil or diesel
7	Mud
-1	Data missing or out of range

Table 6. *Gender*

code	label
1	Male
2	Female
3	Not known
-1	Data missing

Table 7. *Vehicle Type*

code	label
1	Pedal cycle
2	Motorcycle 50cc and under
3	Motorcycle 125cc and under
4	Motorcycle over 125cc and up to 500cc
5	Motorcycle over 500cc
8	Taxi/Private hire car
9	Car
10	Minibus (8 - 16 passenger seats)
11	Bus or coach (17 or more pass seats)
16	Ridden horse
17	Agricultural vehicle
18	Tram
19	Van / Goods 3.5 tonnes mgw or under
20	Goods over 3.5t. and under 7.5t
21	Goods 7.5 tonnes mgw and over
22	Mobility scooter
23	Electric motorcycle
90	Other vehicle
97	Motorcycle - unknown cc
98	Goods vehicle - unknown weight
-1	Data missing or out of range

Table 8. *Day of The Week*

code	label
1	Sunday
2	Monday
3	Tuesday
4	Wednesday
5	Thursday
6	Friday
7	Saturday

2.2 Validation

Machine learning, especially supervised learning techniques such as classification and regression, require training data to build a model. Training data consists of labeled data, i.e. datasets that are complete with the target value together with input feature vectors. A good classification or regression model can be built if a significant amount of training data is supplied during the training process. This is followed by the validation process where test data is fed into the trained model to evaluate its predictive accuracy. It is important to test the model properly with enough test data so that the model would yield accurate predictions in the production environment.

Unfortunately, scarcity of data often prompts machine learning practitioners to split the dataset in hand into two subsets, namely training data and test data. These subsets emerge from splitting the original dataset according to a certain ratio such as 80:20 or 60:40, with the bigger proportion making up the training data subset. Training and validating a model using a single train-test split (a.k.a. holdout method) would not yield significant predictive accuracy due to bias. Bias in this case means that in a single train-test split, data points could be clustered in such a way that one cluster gets stuck in the training set and another cluster gets stuck in the test set. Such a situation leads to bias in the train-test split, thus adversely affecting the predictive accuracy of a model.

Therefore, it is important to utilize several unique splits of training and test data to build an accurate model. Cross validation utilizes several train-test splits, a.k.a folds and this technique enables the machine learning model to be trained with less bias because all different clusters of data points get to be chosen as training data in different folds. This helps to reduce bias in training the model.

2.3 Decision Trees and Random Forests

A choice tree may be a flowchart-like structure in which each inside hub speaks to a "test" on a property, each department speaks to the result of the test, and each leaf hub speaks to a lesson name (choice taken after computing all qualities). The ways from root to leaf speak to classification rules. A decision tree consists of three types of nodes:

Decision nodes – represented as squares

Chance nodes – represented as circle

End nodes – represented as triangles

One purpose of Machine Learning is to design high performance classification systems from a set of representative samples of a population of data. An efficient way to tackle this kind of problem is to combine an ensemble of individual classifiers to form a unique classification system, called Classifier Ensemble. This ability is often defined through the diversity property. Although there is no agreed definition for diversity [6], this concept is usually recognized to be one of the most important characteristics for the improvement of the generalization performance in an ensemble of classifiers [7]. One can define it as the ability of the individual classifiers of an ensemble to agree mainly on good predictions and to disagree on prediction errors.

Random Forest (RF) family of ensemble methods. In a "classical" RF induction process a fixed number of randomized decision trees are inducted to form an ensemble. This kind of algorithm presents two main drawbacks: (i) the number of trees has to be fixed a priori (ii) the interpretability and analysis capacities offered by decision tree classifiers are lost due to the randomization principle. This kind of process in which trees are independently added to the ensemble, offers no guarantee that all those trees will cooperate effectively in the same committee.

CHAPTER 3

METHODOLOGY

We have developed a web app for our model. It consists of four components:

Front-End: Users input for the prediction factors are taken and sent to the backend server.

Back-End: The model is deployed here and the input data is fed into the Machine Learning model.

Machine Learning Model: We have used decision tree, random forest and logistic regression. Random Forest algorithm showed the highest accuracy of 86.86% and hence chosen for our model. The model runs and predicts the severity. The severity metrics are 1= Fatal, 2= Serious, 3= Slight. The output is sent back to the front-end and displayed to the user.

A sms containing the location coordinates and the severity of accident is sent to the police so that it can take preventive measures at the location.

3.1 System Design

Describes the data flow in a diagrammatic representation.

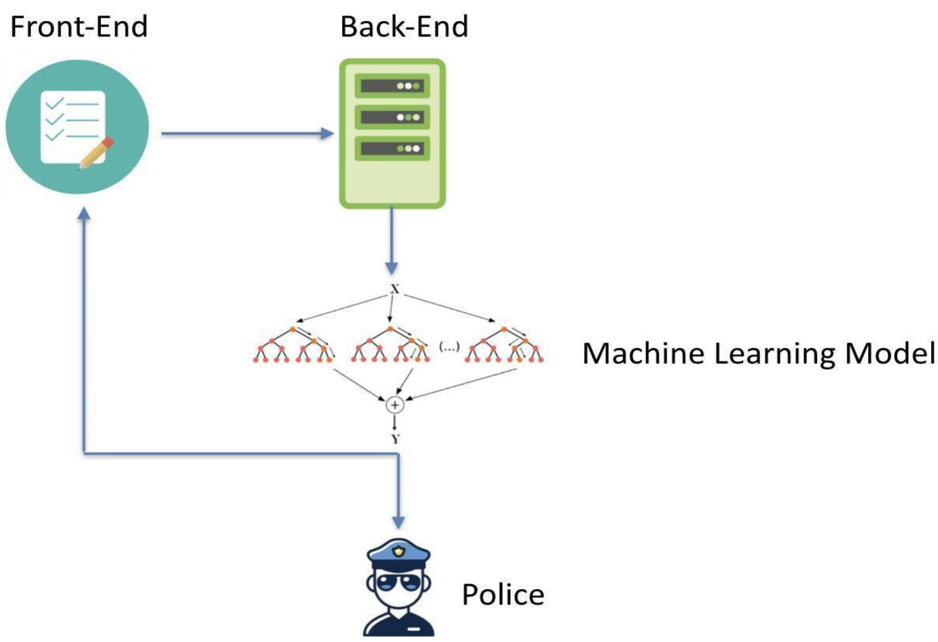


Figure 3.1 System model

3.2 Modules

1. **The Virtual Machine:** It has the trained and tested Machine learning algorithm implemented. The frontend and backend server are deployed on it.
2. **The front end (User):** Geolocation Api takes the location of the user and sends it to the OpenWeatherMap Api which sends geographical conditions. User input is taken for other parameters like age, sex etc. Users can view the heatmap of the accidents in the country.
3. **The back end (Admin):** The server is created and maintained. The input details are given to the model and severity is predicted. The severity can be sent as a message or email to the police to take preventive measures.
4. **Machine Learning Algorithm:** Classification Algorithms decision tree, random forest and logistic regression have been implemented. Random forest has shown the highest accuracy with 86% and has been selected as the model for the web app.

3.3 Technologies Used

3.3.1 Python

Python could be a broadly utilized general-purpose programming language. It was first planned by Guido van Rossum in 1991 and created by the Python Program Foundation. It was primarily created for accentuation on code coherence, and its sentence structure permits software engineers to specific concepts in less lines of code. Python highlights an energetic sort framework and programmed memory management. This underpins different programming ideal models, counting object-oriented, basic, utilitarian and procedural. It too incorporates a comprehensive standard library. It is the world's fastest developing and most well-known programming dialect utilized by computer program engineers, investigators, information researchers, and machine learning engineers alike. It is utilized by locales like YouTube and Dropbox.

It underpins utilitarian and organized programming strategies as well as OOP.
It tends to be used as a pre-arranged vernacular or can be arranged to byte-code for building immense applications. It gives especially undeniable level vivacious data sorts and supports fiery

sort checking. It reinforces customized garbage assortment

3.3.2 Numpy

NumPy is the elemental bundle for logical computing with Python. It contains among other things:

- An effective N-dimensional cluster object
- Advanced functions
- Apparatuses for joining C/C++ and Fortran code
- Valuable straight variable-based math, Fourier change, and arbitrary number capabilities

Other than its self-evident logical employments, NumPy can too be utilized as a proficient multi- dimensional holder of bland data. Subjective data-types can be characterized. This permits NumPy to consistently and quickly coordinated with a wide assortment of databases. Utilizing NumPy in Python gives usefulness comparable to MATLAB since they are both deciphered, and they both permit the client to type in quick programs as long as most operations work on clusters or networks rather than scalars. SciPy may be a library that adds more MATLAB-like usefulness and Matplotlib could be a plotting bundle that gives MATLAB-like plotting functionality. NumPy is authorized beneath the BSD permit, empowering reuse with few restrictions. Python ties of the broadly utilized computer vision library OpenCV utilize NumPy clusters to store and work on data. Since pictures with numerous channels are basically spoken to as three- dimensional clusters, ordering, cutting or concealing with other clusters are exceptionally productive ways to get to particular pixels of an image.

3.3.3 Google collab

Colaboratory (moreover known as Colab) may be a free Jupyter note pad environment that runs within the cloud and stores its note pads on Google Drive. Colaboratory begun as a portion of Extend Jupyter, but the advancement was inevitably taken over by Google [21]. Jupyter is a nonprofit organization made to "create open- source program, open-standards, and administrations for intelligently computing over handfuls of programming languages Spun-off from IPython in

2014 by Fernando Pérez, Extend Jupyter bolsters execution situations in a few dozen languages. Venture Jupyter has created and upheld the intuitively computing items Jupyter Notebook, Jupyter Center, and Jupyter lab, the next-generation adaptation of Jupyter Notebook. Jupyter Scratch pad (once IPython Note pads) may be a web-based intelligently computational environment for making Jupyter note pads documents. The "note pad" term can colloquially make reference to numerous distinctive substances, basically the Jupyter web application, Jupyter Python web server, or Jupyter archive organize depending on setting. A Jupyter Notebook archive could be a JSON record, taking after a versioned pattern, and containing an requested list of input/output cells which can contain code, content (utilizing Markdown), arithmetic, plots and wealthy media, ordinarily finishing with the “.ipynb” extension. It is utilized to run asset seriously tasks.

3.3.4 Scikit-learn

Scikit-learn could be a free program AI library for the Python programming language.[3] It features diverse grouping, backslide and bunching estimations counting reinforce vector machines, sporadic forests, point supporting, k-means and DBSCAN, and is laid out to interoperate with the Python mathematical and sensible libraries NumPy and SciP.

- Straightforward and effective devices for information mining and information analysis
- Available to everyone, and reusable in different contexts
- Completed on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

3.3.5 API

Application Programming Interface (API) In essential terms, APIs fair permit applications to communicate with one another and information to one another.
APIs used are:

1. The Geolocation API returns an area and precision sweep based on data around cell towers and WiFi hubs that the portable client can detect. This record portrays the convention utilized to send this information to the server and to return a reaction to the client. Communication is done over HTTPS utilizing POST. Both ask and reaction are designed as JSON, and the substance sort of both is application/json.

2. Weather API: Given by OpenWeatherMap, you've got to get current climate information, 5- and 16-day figures, UV File, discuss contamination, climate conditions etc.
3. Sms API: Given by Content Nearby. Can be effortlessly coordinated with any application and can be utilized to begin sending SMS in minutes.

3.3.6 SSH Client

Secure Shell (SSH) could be a cryptographic organized convention for working to arrange administrations safely over an unsecured network. Commonplace applications incorporate farther command-line login and farther command execution, but any arrangement benefit can be secured with SSH. SSH gives a secure channel over an unsecured arrangement in a client–server engineering, interfacing an SSH client application with an SSH server. The convention detail recognizes between two major forms, alluded to as SSH-1 and SSH-2. The standard TCP harbour for SSH is 22. SSH is for the most part utilized to get to Unix-like working frameworks, but it can moreover be utilized on Windows. Windows 10 employments OpenSSH as its default SSH client

SSH was planned as a substitution for Telnet and for unsecured inaccessible shell conventions such as the Berkeley rlogin, rsh, and rexec protocols. Those conventions send data, strikingly passwords, in plaintext, rendering them vulnerable to interferences and divulgence utilizing bundle analysis. The encryption utilized by SSH is aiming to supply secrecy and astuteness of information over an unsecured organize, such as the Web, in spite of the fact that records spilled by Edward Snowden show that the National Security Organization can now and then decode SSH, permitting them to studied the substance of SSH sessions.

3.4 Diagrammatic Representation

3.4.1 Data flow diagram

A data flow diagram (DFD) maps out the stream of data for any handle or system. It employs characterized images like rectangles, circles and bolts, additionally brief content names, to appear information inputs, yields, capacity focuses and the courses between each destination. Information flowcharts can run from basic, indeed hand-drawn handle outlines, to in-depth, multi-level DFDs that burrow dynamically more profoundly into how the information is handled. They can be utilized to analyze an existing framework or show an unused one. Like all the most excellent charts and charts, a DFD can regularly outwardly “say” things that would be difficult to clarify in words, and they work for both specialized and nontechnical groups of onlookers, from designer to CEO. That’s why DFDs stay so well-known after all these a long time. Whereas they work well for information stream program and frameworks, they are less pertinent these days to visualizing intelligently, real-time or database-oriented computer program or systems.

DFD level 0

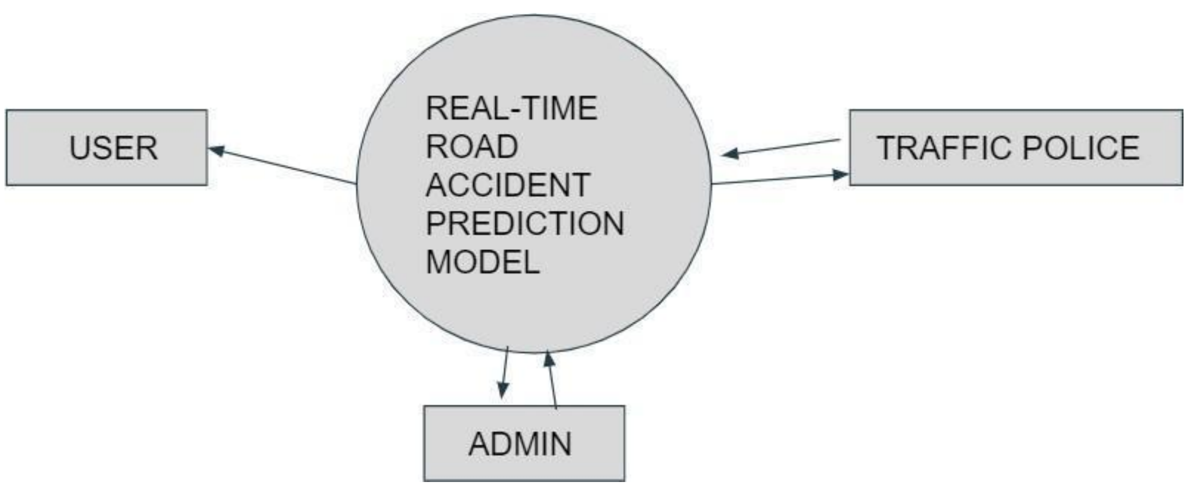


Figure 4.1.1 DFD level 0

- Admin: is responsible for building the ML model and maintaining it.
- User: the person who views the output.
- Traffic police: they take respective action according to the output predicted by the

ML model.

3.4.1.2 DFD level 1

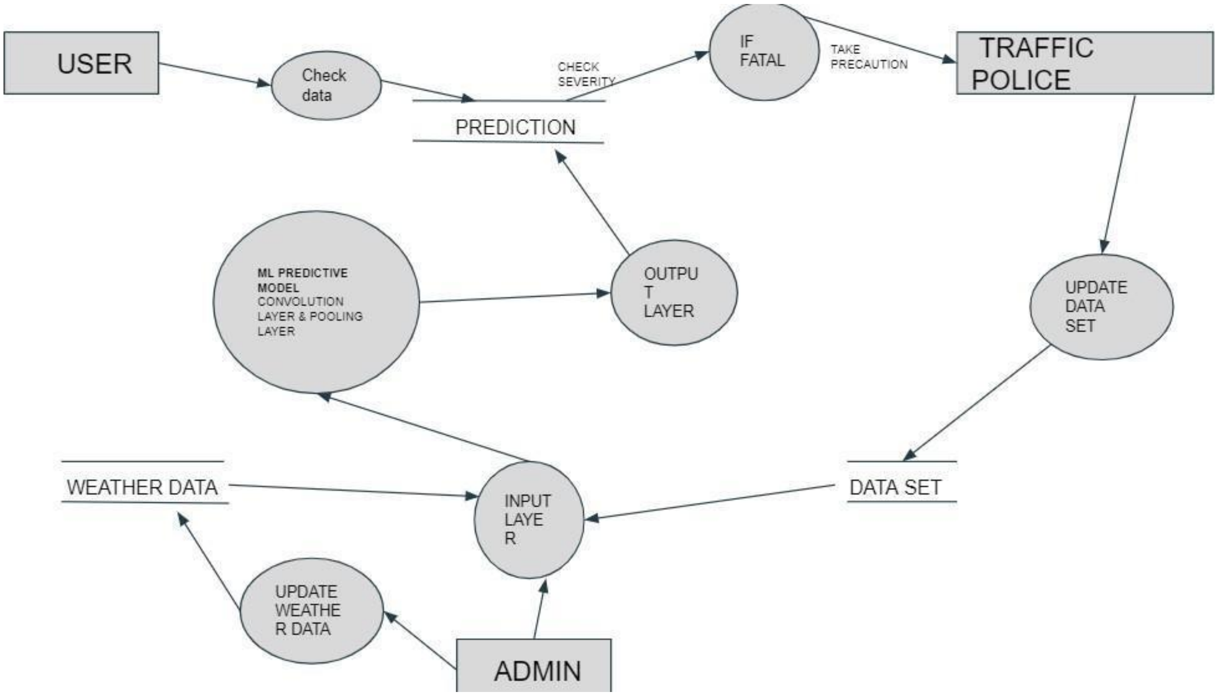


Figure 4.1.2 DFD level 1

- The ML model is further divided into 3 layers
  - Input layer
  - Convolution layer or Pooling layer
  - Output layer

- If the output predicted is severity FATAL, which means that there is high probability for an accident to occur, so an alert is sent to the traffic police to take respective action.

3.4.2 UML Diagrams

UML is the international standard notation for object-oriented analysis and design. The object management group defines it. The heart of object-oriented problem solving is the construction of a model. The model abstracts the essential details of the underlying problem from its usually complicated real world. The scope UML is a language for specifying artifacts, visualizing artifacts, constructing artifacts and documenting artifacts. UML provides the following diagrams to represent the software process:

- Class Diagram
- Use Case diagram
- Sequence diagram

3.4.2.1Class Diagram

A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

The class diagram is the main building block of object-oriented modeling. It is used for general conceptual modeling of the structure of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling. The classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed.

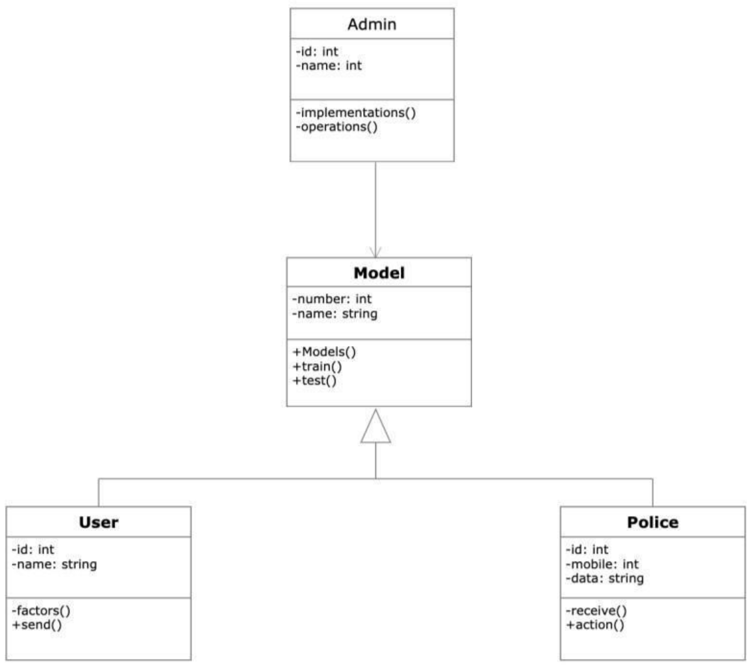


Figure 4.2.3 Class diagram

3.4.2.2Use Case Diagram

Use case Diagram at its least difficult is a portrayal of a client's cooperation with the framework that shows the connection between the client and the distinctive use cases in which the client is involved. A utilization case outline can distinguish the various kinds of clients of a framework and the distinctive use cases and will regularly be joined by different sorts of charts also. The utilization cases are addressed by either circles or ellipses.

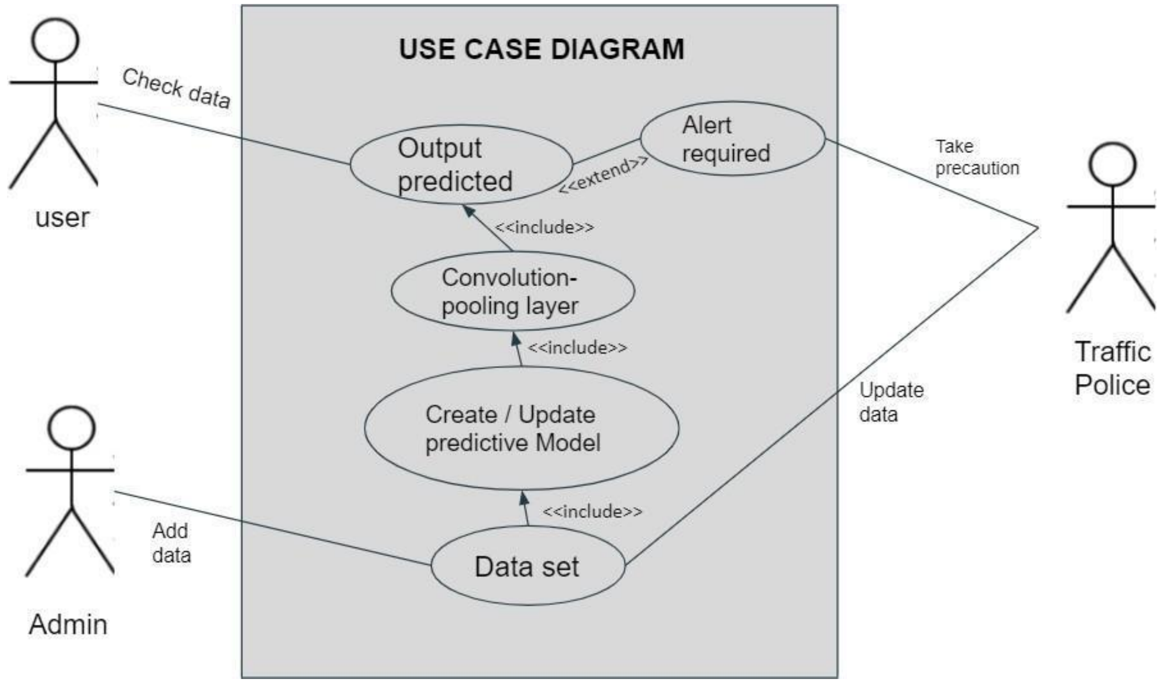
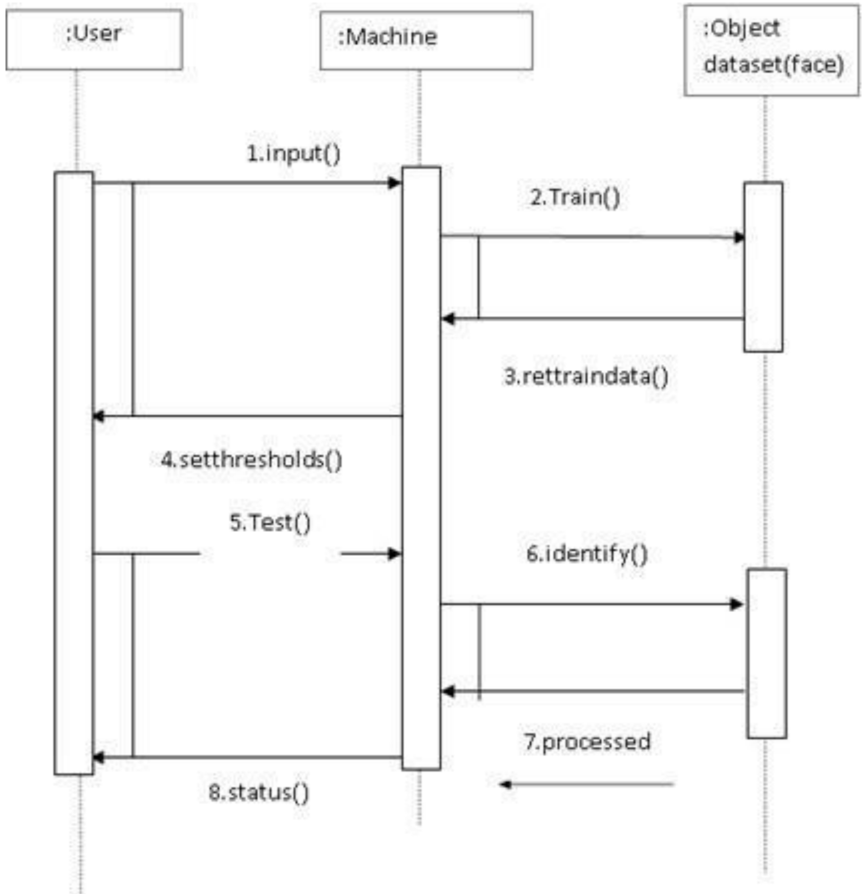


Figure 4.2.2 Usecase diagram

- In the system, there are two actors: user and play store.
- The user performs the tasks of searching for an application and viewing the result if an application is malicious.
- The play store downloads the required application and its comments.
- The other actions performed in the system are testing and sentiment analysis on the download application and comments.

3.4.2.3Sequence Diagram

A grouping graph appears to question intelligence organized in time sequence. It portrays



the objects and classes included within the situation and the grouping of messages traded between the objects required to carry out the usefulness of the scenario. Arrangement charts are regularly related to utilize case realizations within the Consistent See of the framework beneath improvement. Graphs are in some cases called occasion charts. An arrangement chart appears, as parallel vertical lines, diverse forms or objects that live at the same time, and, as level bolts, the messages traded between them, within the arrangement in which they occur. This permits the determination of basic runtime scenarios in a graphical manner.

Figure 4.2.3 Sequence diagram

● The user represented by the object of 'User' class performs the first operation 'searchApp' in the system by sending a message to the object of 'Store' class that represents the Google Play

Store. This operation searches for an application in the Play Store.

● The object of ‘Store’ class then sends a message ‘download’ to an object of ‘Server’ class. This denotes that the required application has to be downloaded.

● The ‘Server’ class object sends the downloaded application to ‘Analysis’ class object and indicates it to perform sentiment analysis and testing on it. This is done through a message ‘test’.

● Finally, the analysis is completed by the ‘Analysis’ object and returns the result to the user through ‘send Result’.

3.5 Implementation of Proposed solution

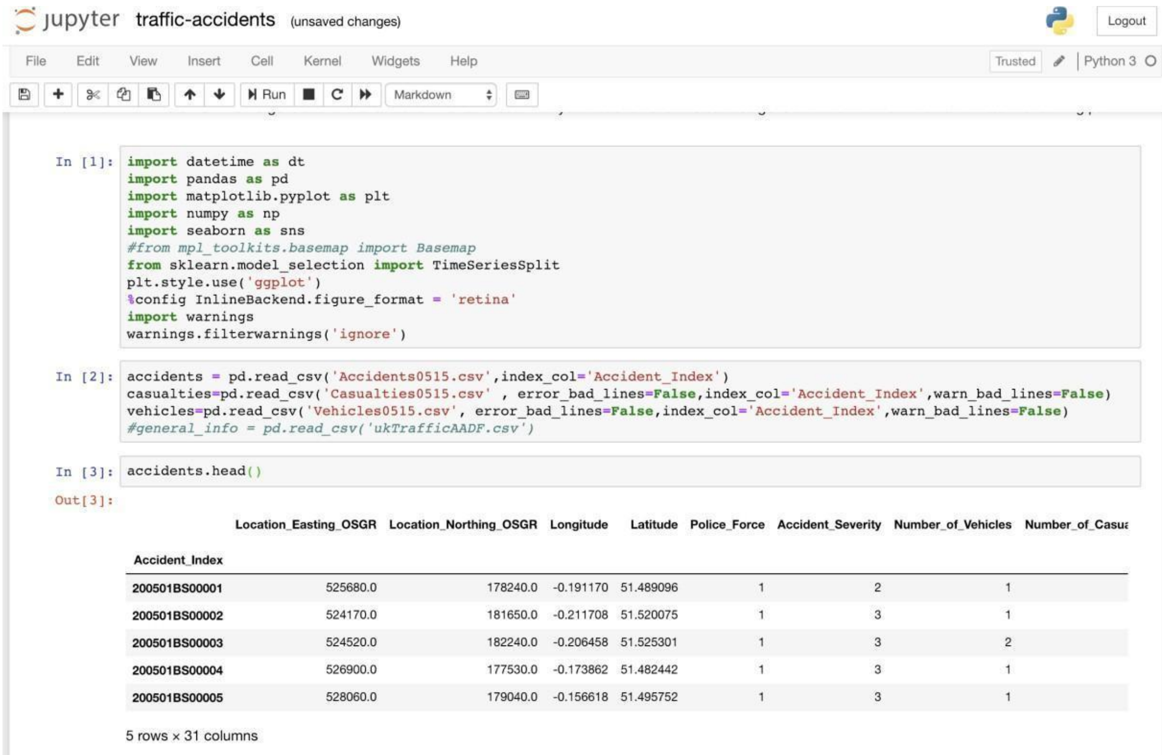
There are four important steps:

1. Preprocessing
2. Training
3. Testing
4. Web App Integration

3.5.1 Data Importing

We import three files to perform analysis on this data. This data consists of three files that are accidents, casualties and vehicles. However, we have one more file which is general information about the traffic count for the year 2000 to 2015. We can use general traffic information data for the machine learning part.

- Importing of packages needed is done.
- 3 CSV files Accidents.csv Casualties.csv Vehicles.csv
- Using pandas to import data into dataframe
- accident.head() views top 5 rows of dataframe



jupyter traffic-accidents (unsaved changes)

```
In [1]: import datetime as dt
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
#from mpl_toolkits.basemap import Basemap
from sklearn.model_selection import TimeSeriesSplit
plt.style.use('ggplot')
%config InlineBackend.figure_format = 'retina'
import warnings
warnings.filterwarnings('ignore')

In [2]: accidents = pd.read_csv('Accidents0515.csv',index_col='Accident_Index')
casualties=pd.read_csv('Casualties0515.csv' , error_bad_lines=False,index_col='Accident_Index',warn_bad_lines=False)
vehicles=pd.read_csv('Vehicles0515.csv', error_bad_lines=False,index_col='Accident_Index',warn_bad_lines=False)
#general_info = pd.read_csv('ukTrafficAADF.csv')

In [3]: accidents.head()
Out[3]:
```

Accident_Index	Location_Easting_OSGR	Location_Northing_OSGR	Longitude	Latitude	Police_Force	Accident_Severity	Number_of_Vehicles	Number_of_Casua
200501BS00001	525680.0	178240.0	-0.191170	51.489096	1	2	1	
200501BS00002	524170.0	181650.0	-0.211708	51.520075	1	3	1	
200501BS00003	524520.0	182240.0	-0.206458	51.525301	1	3	2	
200501BS00004	526900.0	177530.0	-0.173862	51.482442	1	3	1	
200501BS00005	528060.0	179040.0	-0.156618	51.495752	1	3	1	

5 rows × 31 columns

Fig 3.1 Importing

3.5.2 Preprocessing of Data

Data Cleaning

Here we identify noisy, irrelevant data. We also understand through visualization which factors are more important.

Identifying Missing Values

In this particular dataset, there are two types of missing values '-1' and 'Nan'. We will investigate each column with total missing values. We will not be imputing any mean or median value since the dataset is big enough to perform analysis.

```
In [5]: accidents.drop(['Location_Easting_OSGR', 'Location_Northing_OSGR','LSOA_of_Accident_Location',
                        'Junction_Control' ,'2nd_Road_Class'], axis=1, inplace=True)
        accidents['Date_time'] =  accidents['Date'] +' '+ accidents['Time']

        for col in accidents.columns:
            accidents = (accidents[accidents[col]!=-1])
            #print(col ,' ' , x)
        for col in casualties.columns:
            casualties = (casualties[casualties[col]!=-1])

            accidents['Date_time'] = pd.to_datetime(accidents.Date_time)
        accidents.drop(['Date','Time'],axis =1 , inplace=True)
        accidents.dropna(inplace=True)
```

Using join method to combine accidents and vehicles files as they have the same primary key Accident Index.

```
In [4]: accidents = accidents.join(vehicles, how='outer')
```

Fig 3.2 Join

Data Visualization

The first thing we can do is to find out about accidents, time to get intuition and some driver's age who are involved in the accident.

- We can find out the number of accidents on the days of a week.
- We can find out about the accident number using hours of the day.
- Finding out about the age of the driver can tell us more about the accidents.

Accidents on Day of Week

We can find out the number of accidents on the days of a week. As we can see, Thursday has the highest number of accidents in this dataset from 2005 to 2015. We have to keep in mind that accident numbers could be depending on traffic on a particular day.

```
In [6]: plt.figure(figsize=(12,6))
        accidents.Date_time.dt.dayofweek.hist(bins=7,rwidth=0.55,alpha=0.5, color= 'orange')
        plt.title('Accidents on the day of a week' , fontsize= 30)
        plt.grid(False)
        plt.ylabel('Accident count' , fontsize = 20)
        plt.xlabel('0 - Sunday ,  1 - Monday  ,2 - Tuesday , 3 - Wednesday , 4 - Thursday , 5 - Friday , 6 - Saturday' , fontsi

Out[6]: Text(0.5,0,'0 - Sunday ,  1 - Monday  ,2 - Tuesday , 3 - Wednesday , 4 - Thursday , 5 - Friday , 6 - Saturday')
```

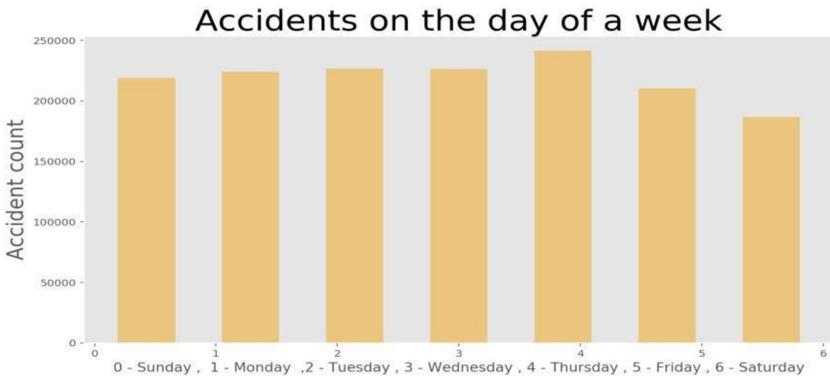


Fig 3.3 Accidents

Time of Accident

He found out that most of the accidents happened around afternoon. We can assume that this time of the day has the most traffic moving such as people leaving from work.

```
In [7]:
        plt.figure(figsize=(12,6))
        accidents.Date_time.dt.hour.hist(rwidth=0.75,alpha =0.50, color= 'orange')
        plt.title('Time of the day/night',fontsize= 30)
        plt.grid(False)
        plt.xlabel('Time 0-23 hours' , fontsize = 20)
        plt.ylabel('Accident count' , fontsize = 15)

Out[7]:
        Text(0,0.5,'Accident count')
```

Fig 3.4 Time

Age Band of Casualties

In this dataset, age band is grouped in 11 different codes. We will create the labels and pass it to the plot as xticks so we can have idea about the bins representation

```
In [8]: objects = ['0','0-5','6-10','11-15','16-20','21-25','26-35',
                   '36-45', '46-55','56-65','66-75','75+']

        plt.figure(figsize=(12,6))
        casualties.Age_Band_of_Casualty.hist(bins = 11,alpha=0.5,rwidth=0.90, color= 'red',)
        plt.title('Age of people involved in the accidents', fontsize = 25)
        plt.grid(False)
        y_pos = np.arange(len(objects))
        plt.xticks(y_pos , objects)
        plt.ylabel('Accident count' , fontsize = 15)
        plt.xlabel('Age of Drivers', fontsize = 15,)

Out[8]: Text(0.5,0,'Age of Drivers')
```

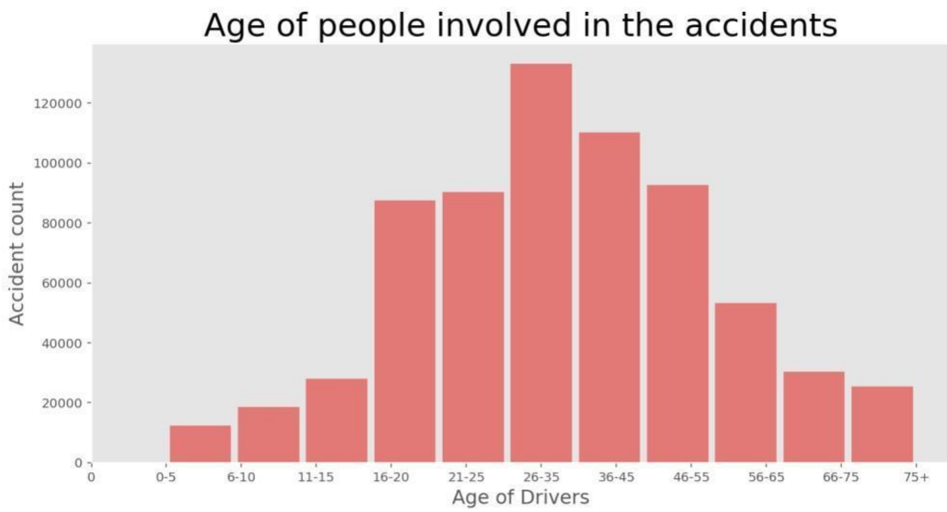


Fig 3.5 Age

This is a very interesting fact about this dataset. Most of the drivers are around 225 to 35 who are involved in the accident. However, we do not know the number of drivers aged 25 to 35 on the road compared to other ages. Intuitively, I would assume that the number of drivers aged 25 to 35 are more than the number of drivers with different ages.

Correlation between variables

Since our dataset is in numeric values. We can find out the correlation between columns.
As we see that there are not so many strong correlations between any variables. There is only one positive strong correlation between speed limit and Urban or Rural Area.

```
In [52]: corr =  accidents.corr()
         plt.subplots(figsize=(20,9))
         sns.heatmap(corr)

Out[52]: <matplotlib.axes._subplots.AxesSubplot at 0x1a3fde1c50>
```

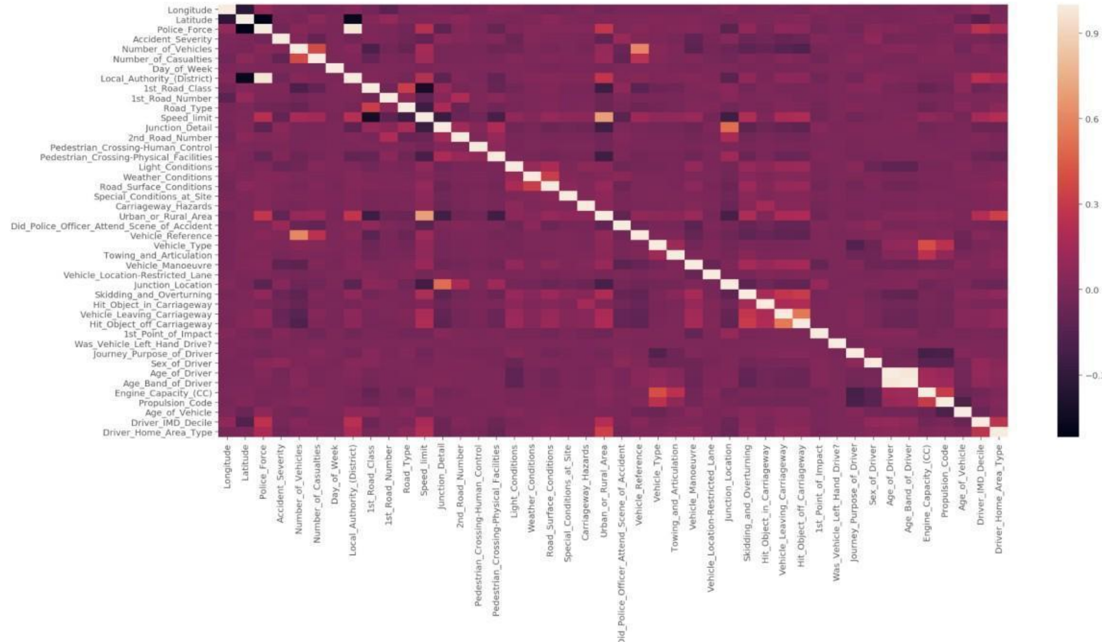


Fig 3.6 Correlation

Speed of Cars

```
In [41]: speed_zone_accidents = accidents.loc[accidents['Speed_limit'].isin(['20' ,'30' ,'40' ,'50' ,'60' ,'70'])]
         speed  = speed_zone_accidents.Speed_limit.value_counts()

         explode = (0.0, 0.0, 0.0 , 0.0 ,0.0,0.0)
         plt.figure(figsize=(10,8))
         plt.pie(speed.values,  labels=None,
                 autopct='%.1f',pctdistance=0.8, labeldistance=1.9 ,explode = explode, shadow=False, startangle=160,textprops={'

         plt.axis('equal')
         plt.legend(speed.index, bbox_to_anchor=(1,0.7), loc="center right", fontsize=15,
                    bbox_transform=plt.gcf().transFigure)
         plt.figtext(.5,.9,'Accidents percentage in Speed Zone', fontsize=25, ha='center')
         plt.show()
```

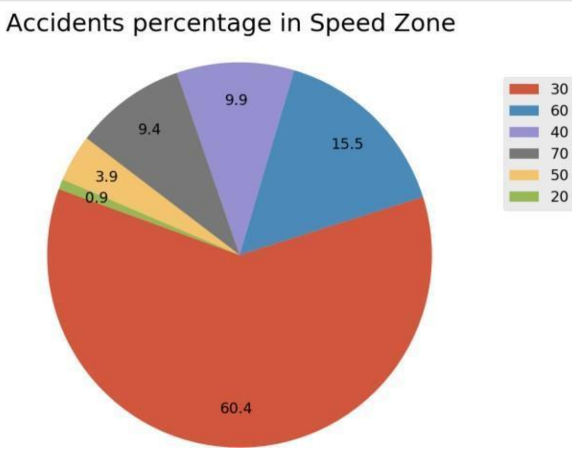


Fig 3.7 Speed

Most of the accidents occurred on the road where the speed limit is 30. We were expecting more accidents on highways or major roadways. Some of the accidents could be because of stop signs, changing lanes or turning into parking lots etc.

Plotting accidents Location on Google Maps

Classifying locations based on severity

In [11]:
```
accidents_2014 = accidents[accidents.Date_time.dt.year ==2014]
accidents_2014_01 = accidents_2014[accidents_2014.Accident_Severity == 1]
accidents_2014_02 = accidents_2014[accidents_2014.Accident_Severity == 2]
accidents_2014_03 = accidents_2014[accidents_2014.Accident_Severity == 3]
```

In [50]:
```
!pip install gmaps
 !jupyter nbextension enable --py gmaps
import gmaps
gmaps.configure(api_key='AIzaSyD3t4mfJNy9NxxVKT4J_T47soKBgCRUTO4')

fig = gmaps.figure(center=(53.0, 1.0), zoom_level=6)
heatmap_layer = gmaps.heatmap_layer(accidents_2014_01[["Latitude", "Longitude"]],
                                    max_intensity=30,point_radius=8)
heatmap_layer = gmaps.heatmap_layer(accidents_2014_02[["Latitude", "Longitude"]],
                                    max_intensity=5,point_radius=3)
heatmap_layer = gmaps.heatmap_layer(accidents_2014_03[["Latitude", "Longitude"]],
                                    max_intensity=1,point_radius=1)

fig = gmaps.figure()
fig.add_layer(heatmap_layer)
fig
```

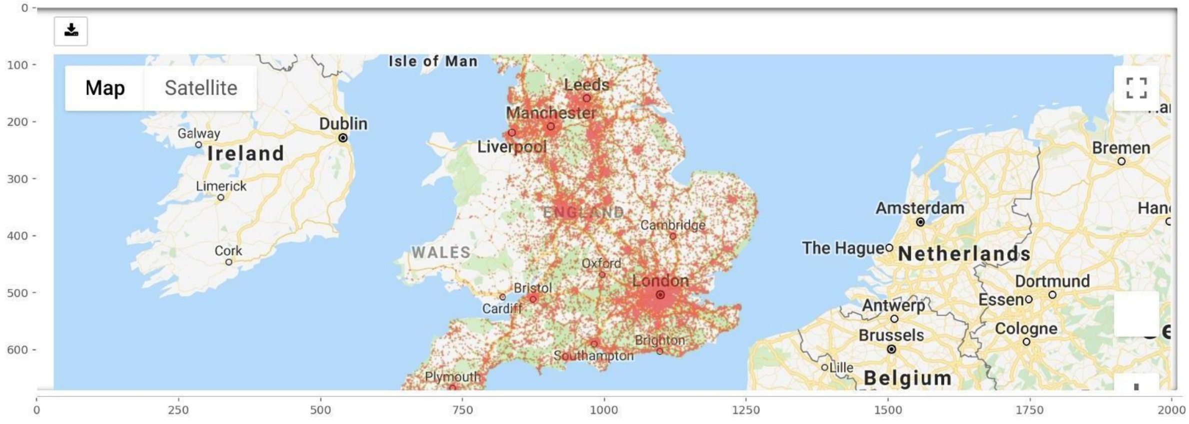
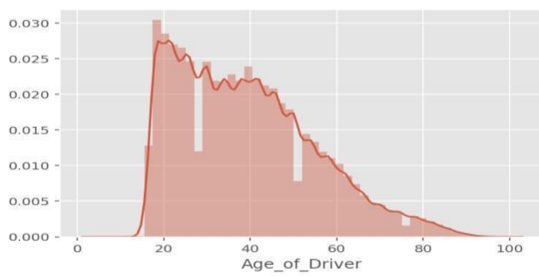


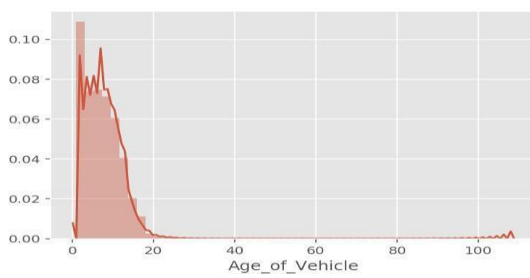
Fig 3.8 Heatmap

Normalize the Data

There are few columns that we will standardize, so it would not affect negatively on our machine learning algorithms. Age of the driver is from 18 to 88 in the dataset and we can normalize it. Also, the age of the vehicle is also from 0 to 100 and it can skew the performance of your machine learning algorithm and we will normalize this predictor too.

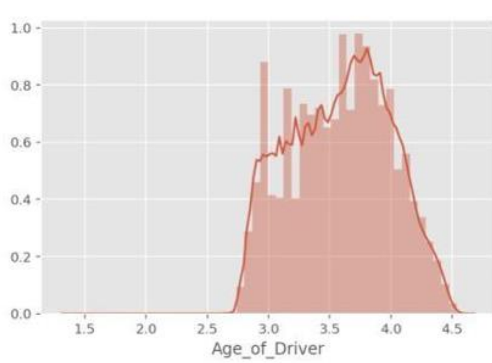
Before Normalization






```
In [ ]: accidents['Age_of_Driver'] = np.log(accidents['Age_of_Driver'])
        accidents['Age_of_Vehicle'] = np.log(accidents['Age_of_Vehicle'])
        sns.distplot(accidents['Age_of_Driver']);
        fig = plt.figure()
        sns.distplot(accidents['Age_of_Vehicle']);
        fig = plt.figure()
```

After Normalization



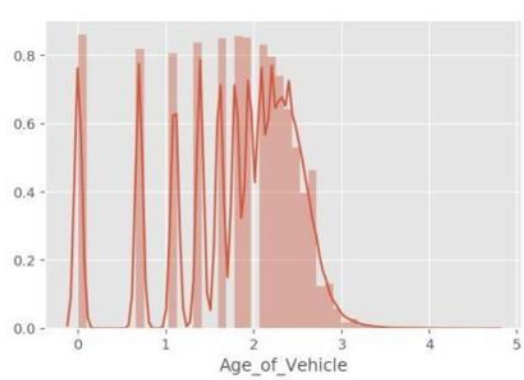


Fig 3.9 Normalization

3.5.3 Machine Learning

We will be looking at different columns to figure out predicting the accident's severity. After we can predict the accident severity, we can make some recommendations to law enforcement for looking into this and be prepared for the future.

Following packages are being imported.

```
In [136]: from sklearn.model_selection import train_test_split
          from sklearn.linear_model import LogisticRegression
          from sklearn.metrics import accuracy_score
          from sklearn.metrics import classification_report
          from sklearn.tree import DecisionTreeClassifier
          from sklearn.ensemble import RandomForestClassifier
          from sklearn.svm import SVC, LinearSVC
          from sklearn.metrics import log_loss
```

Fig 4.0 Packages

Splitting the data into training and test data

X is the input data and Y is the class label.

20% of the data is for testing and 80% for training.

```
In [ ]: accident_ml = accidents.drop('Accident_Severity' ,axis=1)
        accident_ml = accident_ml[['Did_Police_Officer_Attend_Scene_of_Accident' , 'Age_of_Driver' ,'Vehicle_Type', 'Age_of_Veh
                                 , 'Light_Conditions', 'Sex_of_Driver' ,'Speed_limit']]

        # Split the data into a training and test set.
        X_train, X_test, y_train, y_test = train_test_split(accident_ml.values,
                                                            accidents['Accident_Severity'].values,test_size=0.20, random_state=99)
```

Algorithms and Techniques

Logistic Regression

```
('Accuracy', 86.23)
              precision    recall  f1-score   support

           1   0.000000  0.000000  0.000000      4111
           2   0.000000  0.000000  0.000000     38151
           3   0.862320  0.999996  0.926069    264697

   micro avg   0.862317  0.862317  0.862317    306959
   macro avg   0.287440  0.333332  0.308690    306959
weighted avg   0.743596  0.862317  0.798568    306959
```

Predicted	1	3	All
Actual			
1	0	4111	4111
2	0	38151	38151
3	1	264696	264697
All	1	306958	306959

Fig 4.1 Accuracy: Logistic Regression

Decision Tree

```
('Accuracy', 75.26)
              precision    recall  f1-score   support

           1   0.036793  0.046217  0.040970      4111
           2   0.158974  0.187780  0.172180     38151
           3   0.871137  0.844921  0.857829    264697

   micro avg   0.752550  0.752550  0.752550    306959
   macro avg   0.355635  0.359639  0.356993    306959
weighted avg   0.771451  0.752550  0.761672    306959
```

Predicted	1	2	3	All
Actual				
1	190	894	3027	4111
2	931	7164	30056	38151
3	4043	37006	223648	264697
All	5164	45064	256731	306959

Fig 4.2 Accuracy: Decision Tree

Random Forest

```
('Accuracy', 86.86)
              precision    recall  f1-score   support

           1   0.031496  0.002928  0.005358      1366
           2   0.195915  0.040622  0.067291     20777
           3   0.884926  0.979143  0.929653    166321

   micro avg   0.868601  0.868601  0.868601    188464
   macro avg   0.370779  0.340898  0.334101    188464
weighted avg   0.802781  0.868601  0.827884    188464

done
```

Fig 4.31 Accuracy: Random Forest

Algorithm	**Accuracy**
Decision Tree	75.32
Random Forest	86.86
Logistic Regression	86.23

3.6 System Requirements

1. Windows XP, 7, 8, 10, Server 2003.
2. MacOS, iOS
3. Android
4. Windows Phone 8 and Windows 10 Mobile
5. Language Used: HTML5, CSS3, Python, JavaScript
6. IDE and Framework: Jupyter Notebook, Sublime, Flask
7. Cloud: Google Colab

3.6.1 Browsers

1. Chrome
2. Internet Explorer
3. Firefox
4. Safari
5. Edge

3.6.2 Hardware Requirements

1. System	: Microsoft Windows 7 or up.
2. Hard Disk	: 20 GB
3. Ram	: 16 GB
4. Internet Connection	: Network using 10 Mbps or higher.

CHAPTER 4

RESULTS AND DISCUSSIONS

The Front-End which is the home page takes input for the prediction factors in two ways.

1. Area and Latitude: This will be consequently taken from the program utilizing the GeoLocation API and it is shipped off the OpenWeatherMap API. This programming interface gives us the climate, street conditions, light conditions and day of the week which are verifiably refreshed in the backend.
2. Client age, sex, vehicle type, vehicle age and motor limit: This information is to be unequivocally entered by the client.

The model is conveyed in the back-end. The info information from the front-end is taken care of into the Machine Learning model. We have utilized the Random Forest calculation which showed the most noteworthy exactness of 86.86% as our model. The model runs and predicts the seriousness. The measures are 1= Fatal, 2= Serious, 3= Slight.

The result is sent back to the front-end and shown to the client.

A SMS containing the area facilitates and the seriousness of the mishap is shipped off to the police with the goal that it can go to preventive lengths at the area.

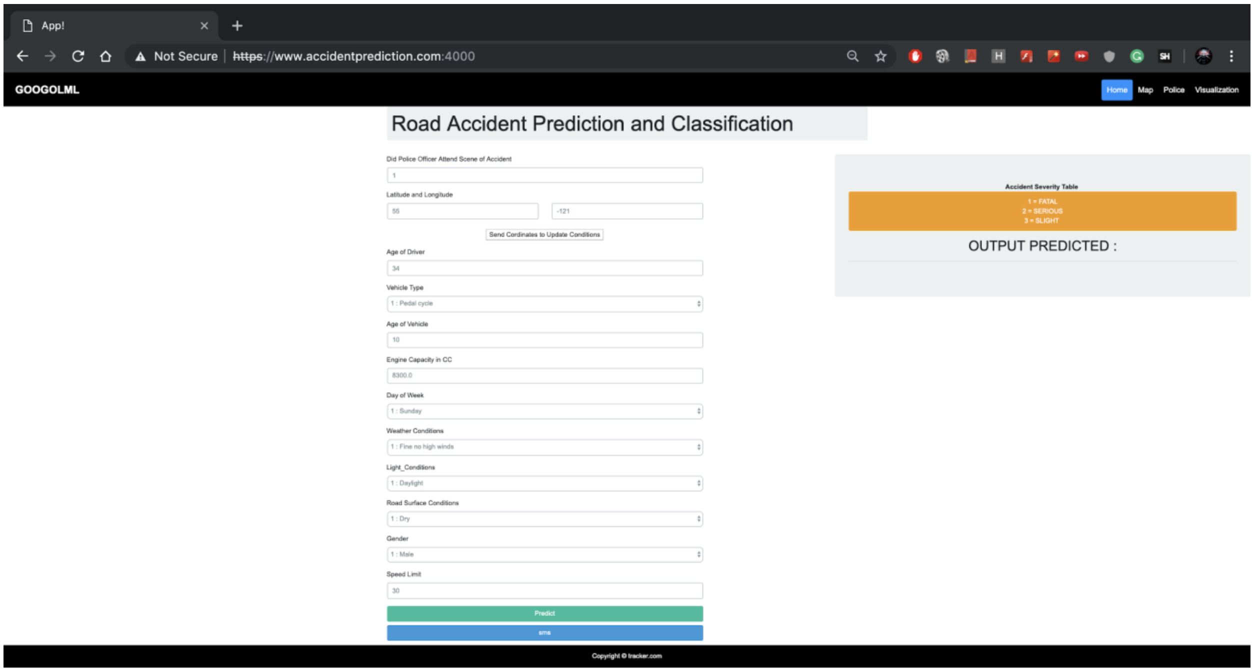


Figure 4.1 User page

The above figure 4.1 shows the landing page of the web application. The web area is obtained with HTTPS which has been acquired from the testament expert for secure information move and to have the option to utilize the Geolocation API. Shows the information proprietor login page, which permits information proprietors to login and furthermore to enroll, in the event that the client doesn't have a current record.

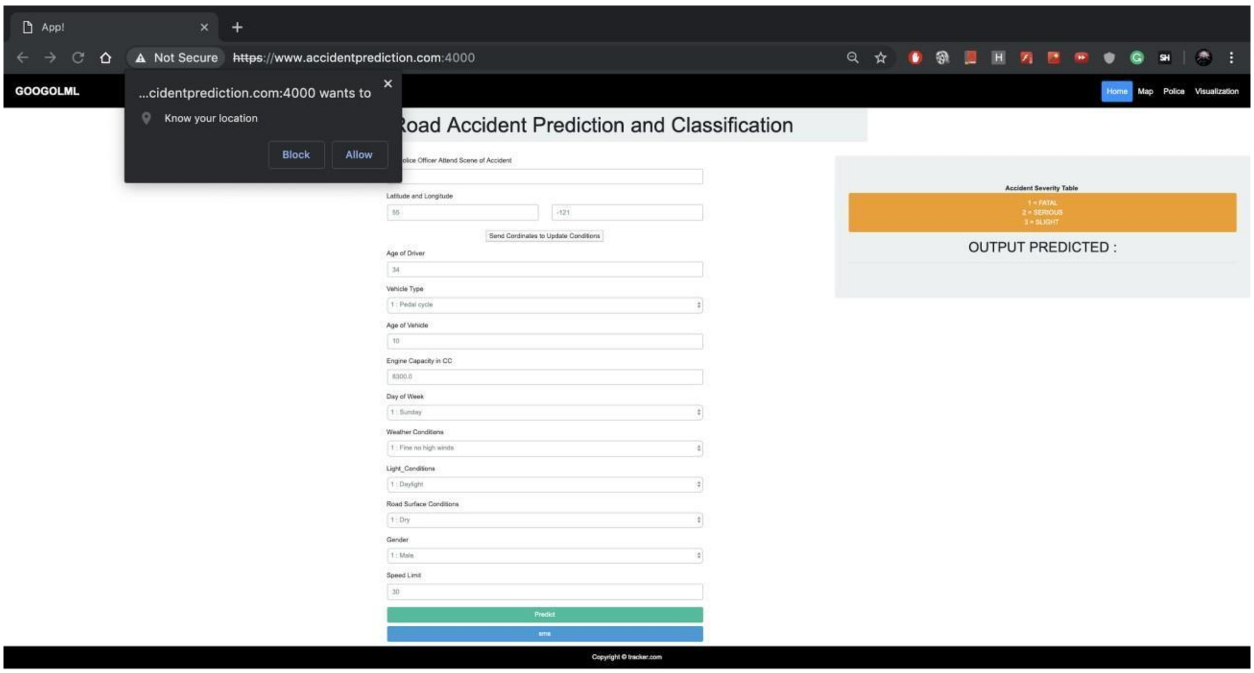


Figure 4.2 User Location by GPS

Figure 6.2.2 shows that when a client taps on the update facilitates button, the page demands the program to take client orders. In the backend carafe module, GeoLocation API is utilized to get the area of the client. Ajax is utilized to refresh the scope and longitude of the client in the site page.

The directions are shipped off the OpenWeatherMap API in the backend for the climate subtleties. From the reaction we remove the subtleties we require like climate, street and light conditions.

Day of the week is taken from the getDate function of javascript.

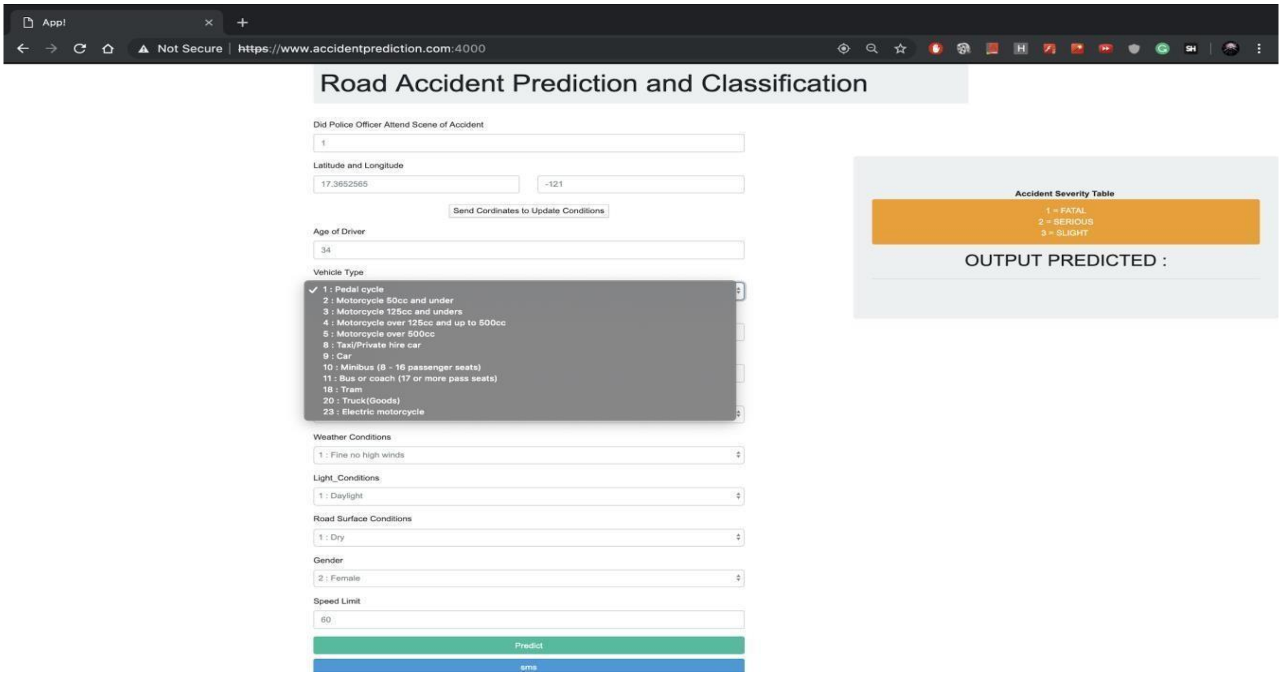


Figure 4.9 User input for other parameters

Figure 4.9 shows the contribution for boundaries taken from clients. These incorporate the vehicle type, age, sexual orientation and speed limit.

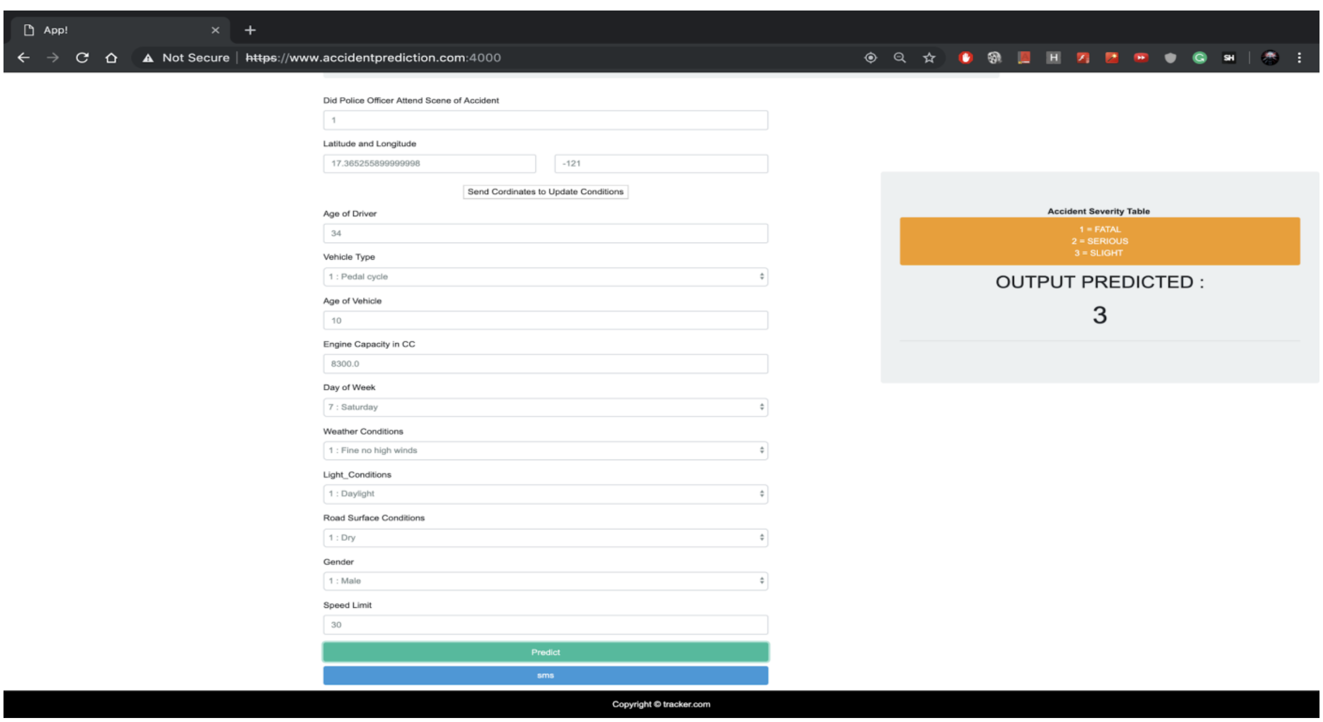


Figure 4.10 Output Predicted

Figure 4.10 shows every one of the information of the client. At the point when the client taps on Predict, that information is shipped off the backend from where it is given to our picked AI calculation which is Random Forest. The measures are 1= Fatal, 2= Serious, 3= Slight.

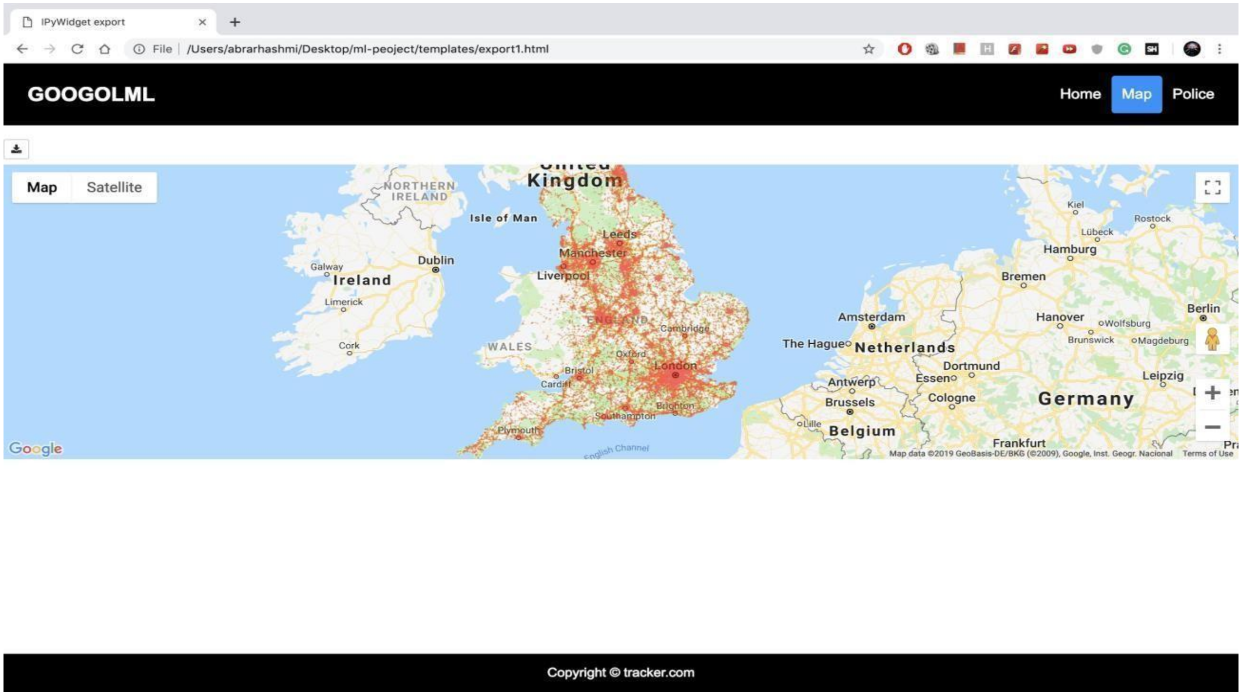


Figure 4.12 Map

In Figure 4.12 This website page shows an intuitive hotness map for clients. Hazier focuses mean more noteworthy seriousness. The gmaps programming interface is utilized to plot on google maps.

Figure 4.13 Visualization

Figure 4.13. It shows what various variables of the dataset mean for the result.

For instance: in the picture above with regards to the time of drivers, we can gather that individuals in the ages 26-35 are more inclined to have a mishap. From such factual information we had the option to pick the variables from the dataset.

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 Conclusions

This project aims at using Machine Learning classification techniques to predict severity of an accident at any particular location.

Machine Learning has enabled us to analyze meaningful data to provide solutions with greater accuracy than with humans. We have built a model with an accuracy greater than 17% of the conventional system [1]. A web-based app using the most accurate algorithm has been developed which can be accessed through the domain name https://www.accidentprediction.com:4000.

This project can be used by governments to prevent accidents.

5.2 Future Work

With more resources, continuous prediction and alerts can be sent to the police for every location at regular intervals of time to take preventive measures. The web app can be incorporated with Google Maps which can be live tracked by the police. A fully-fledged web app for user and police interaction can be published for use in real-time. It can be used for Indian states or cities, if proper data of accidents is provided by the Indian Government.

REFERENCES

[1] Lu Wenqi, Luo Dongyu & Yan Menghua, “A Model of Traffic Accident Prediction” INSPEC Accession Number: 17239218 DOI: 10.1109/ICITE.2017.8056908

[2] Thineswaran Gunasegaran Yu-N Cheah, “Evolutionary Cross validation” INSPEC Accession Number: 17285520 DOI: 10.1109/ICITECH.2017.8079960

[3] Simon Bernard, Laurent Heutte and Sebastien Adam, “On the Selection of Decision Trees in Random Forests” INSPEC Accession Number: 10802866 DOI: 10.1109/IJCNN.2009.5178693

[4] Fu Huilin, Zhou Yucai, “The Traffic Accident Prediction Based on Neural Network”, 2011

[5] Lin, L., Wang, Q., Sadek, A.W., 2014. Data mining and complex networks algorithms for traffic accident analysis. In: Transportation Research Board 93rd Annual Meeting (No. 14-4172).

[6] Gunasegaran, T., & Cheah, Y.-N. (2017). Evolutionary cross validation. 2017 8th International Conference on Information Technology (ICIT). doi:10.1109/icitech.2017.8079960

[7] Bernard, S., Heutte, L., & Adam, S. (2009). On the selection of decision trees in Random Forests. 2009 International Joint Conference on Neural Networks. doi:10.1109/ijcnn.2009.5178693

[8] Matthew Bihis ; Sohini Roychowdhury, “A generalized flow for multi-class and binary classification tasks: An Azure ML approach”