PES UNIVERSITY

Project Review-3 : 90% Milestone

Project Title : Data In Action - Telematics Data

Project Guide : Dr. Mamatha H R

Project Team :

PES1201800368	G Sree Pranavi
PES1201800797	Harshitha Batta
PES1201801087	Anjana V Murthy

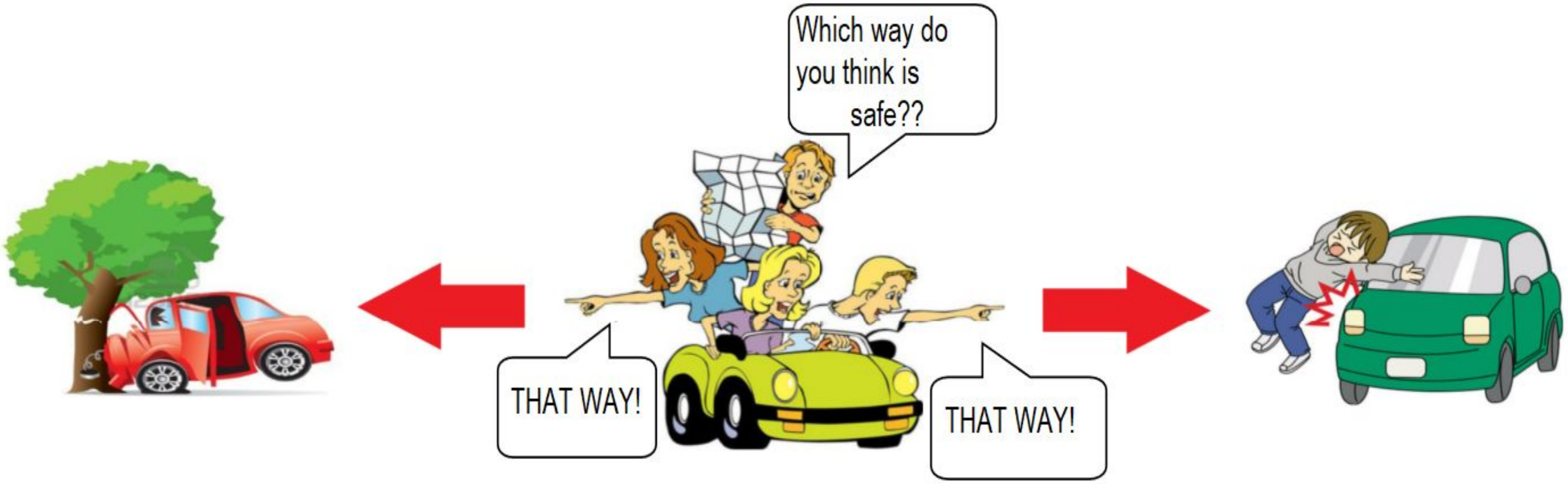
CONTENTS

Aim

The prime objective of the project is to ensure the user's safety and provide emergency assistance while trying to avoid varied types of road accidents based on the research and insights drawn from the provided fleet data.

Our model detects unexpected situations and alerts the driver before hand using the alarm details with respect to location and time using the prediction model constructed to avoid road mishaps.

Insights regarding the driver's behavior can be concluded from the analysis formed, which can be helpful for organizations like Ola to have a clear cut idea of its drivers' behavioural stats and driving patterns.



IS YOUR WAY THE SAFE WAY?

The Idea - in detail

Considering the significance of each of the alert types, the following can be expected from the model.

- PCW which is activated when the car senses a pedestrian, can be used to find out the location and time of the highest density of pedestrians in order to take some safety considerations.
- The driver is alerted with HMW if the threshold distance between the vehicle and its preceding vehicle is crossed.
- FCW detects imminent collisions and alerts upto 2.7 secs before collision by calculating time to collision.

If the above 2 alert types are observed repetitively at a particular location as well, we can consider that location endures dense traffic.

- If HB occurs recursively at a particular location irrespective of time and previous alarm, that location can be assumed to be causing some physical geographical troubles.

- If a HB follows FCW/ HMW or occurs irrespective of any of the alarms recursively for a particular vehicle (that is no emergency), it can be assumed to be rash driving. If stoppage follows HB, the intensity can be identified to be quite high as the speed drops suddenly.

- If alarm types FCW and HMW occur recursively for a particular vehicle, and reach beyond threshold, the driver can be assumed of rash driving. If followed by HB, assumption is strengthened.

NEED HELP GETTING OUT OF TRAFFIC?

Innovation Quotient

After extensive literature survey we found that substantial exploration has not been carried out on the telematics data and its potential to envisage traffic analysis.

- Most of the research and analysis at present focus on in car telematics analysis such as driver’s response to the alarm types or driver identification and reporting rash driving.

- The general usage of this data is observed to be route prediction, helping in automation of vehicles, air pollution analysis and vehicle fleet fuel consumption investigations. Our application of telematics data is not yet fully established in the research and analysis carried out till date.

- None of the studies and analysis focused on the pressing problem, traffic congestion prediction using fleet alerts, along with geographical disturbances' prediction for a better and safe journey.

- At present researchers are investigating with video and camera based surveillance systems' data for traffic flow prediction.

In conclusion, we have presented a new angle of using telematics data and putting it into action. Our model alerts the user with pedestrian density and traffic density at a particular location and time, providing a novel approach to traffic congestion prediction.

The user can either manually enter the location coordinates or can get a prediction at his/her live location (live tracking).

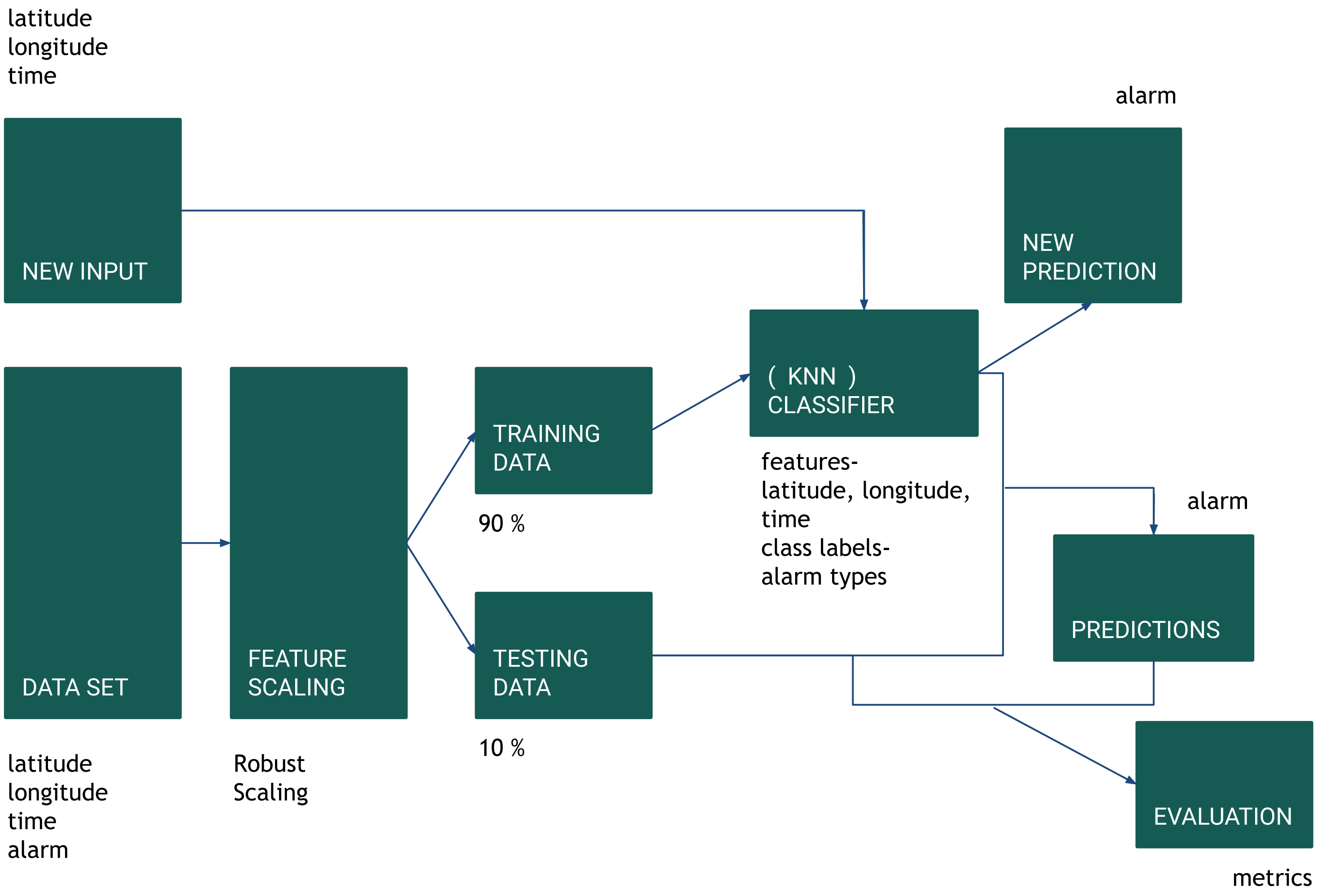
Plan For The Next Milestone

- Final submission

  Since our model is ready to accept the user input from the user interface and provide the predicted output, we will be focussing on improving the model and adding extra features to our user interface for better performance (discussed under key deliverables for next milestone).

  The final, 100 percent user friendly model to support our novel idea will be ready.

Model Architecture



latitude
longitude
time

NEW INPUT

alarm

NEW PREDICTION

(KNN)
CLASSIFIER

features-
latitude, longitude, time
class labels-
alarm types

DATA SET

latitude
longitude
time
alarm

FEATURE SCALING

Robust
Scaling

TRAINING DATA

90 %

TESTING DATA

10 %

alarm

PREDICTIONS

EVALUATION

metrics

Progress as on Date

The prediction model with a simple UI is ready to accept user inputs and provide the specific output in quite an appealing representation, discussed in detail in the further slides.

Link to the readme file:

DataInAction_90%_readme

Link to the code:

DataInAction_90%_code

Link to the clean, modified dataset:

DataInAction_90%_Data

** Please run the code with the provided dataset only.

Progress as on Date (GUI)

➢ Flask is used to deploy the prediction model.

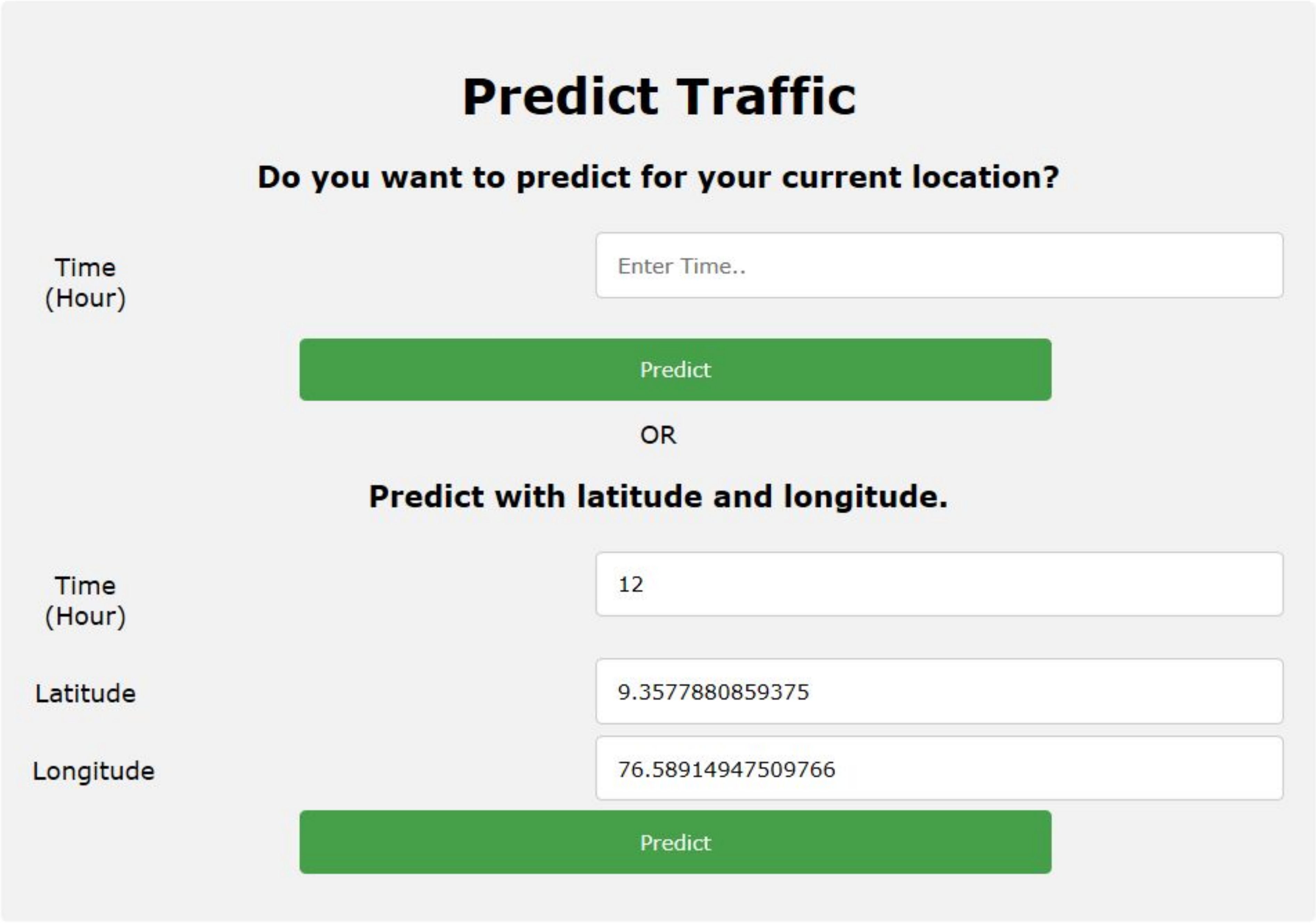
➢ HTML and CSS are used to implement the frontend.

➢ Our interface consists of 3 web pages.

The first page of our model as shown in the next 2 slides, takes in the user input in 2 different ways as below,

- One method allows the user to manually give the input i.e the latitude, longitude and hour, and on clicking the “Predict” button the traffic will be predicted.
- The second method makes it much easier for the user, as it directly takes in the user’s current location. All the user has to do is enter the time and on clicking “Predict”, it predicts the traffic at that hour.

home.htm



Predict Traffic

Do you want to predict for your current location?

Time (Hour): Enter Time..

OR

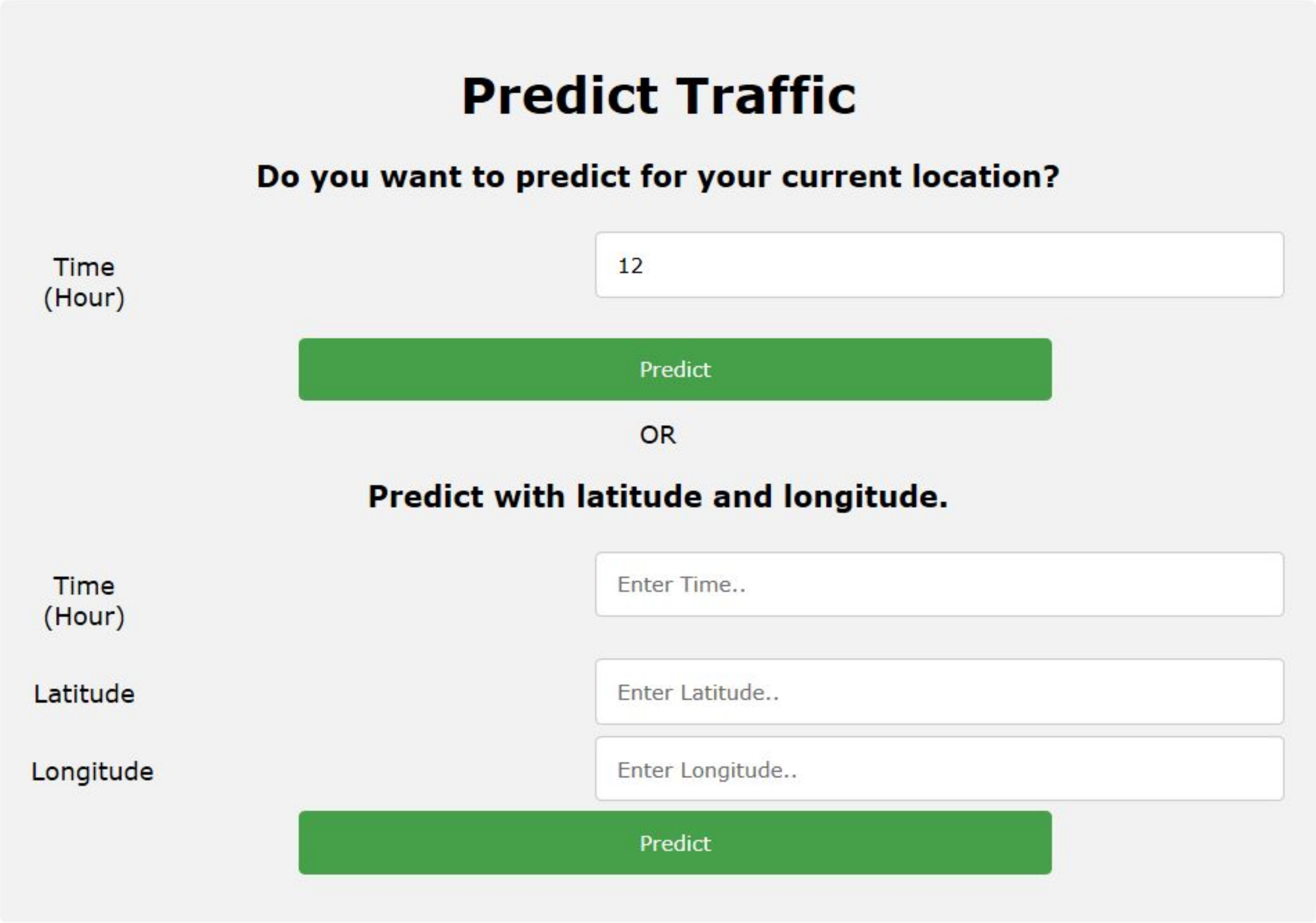
Predict with latitude and longitude.

Time (Hour): 12

Latitude: 9.3577880859375

Longitude: 76.58914947509766

home.htm (current location)



Predict Traffic

Do you want to predict for your current location?

Time (Hour): 12

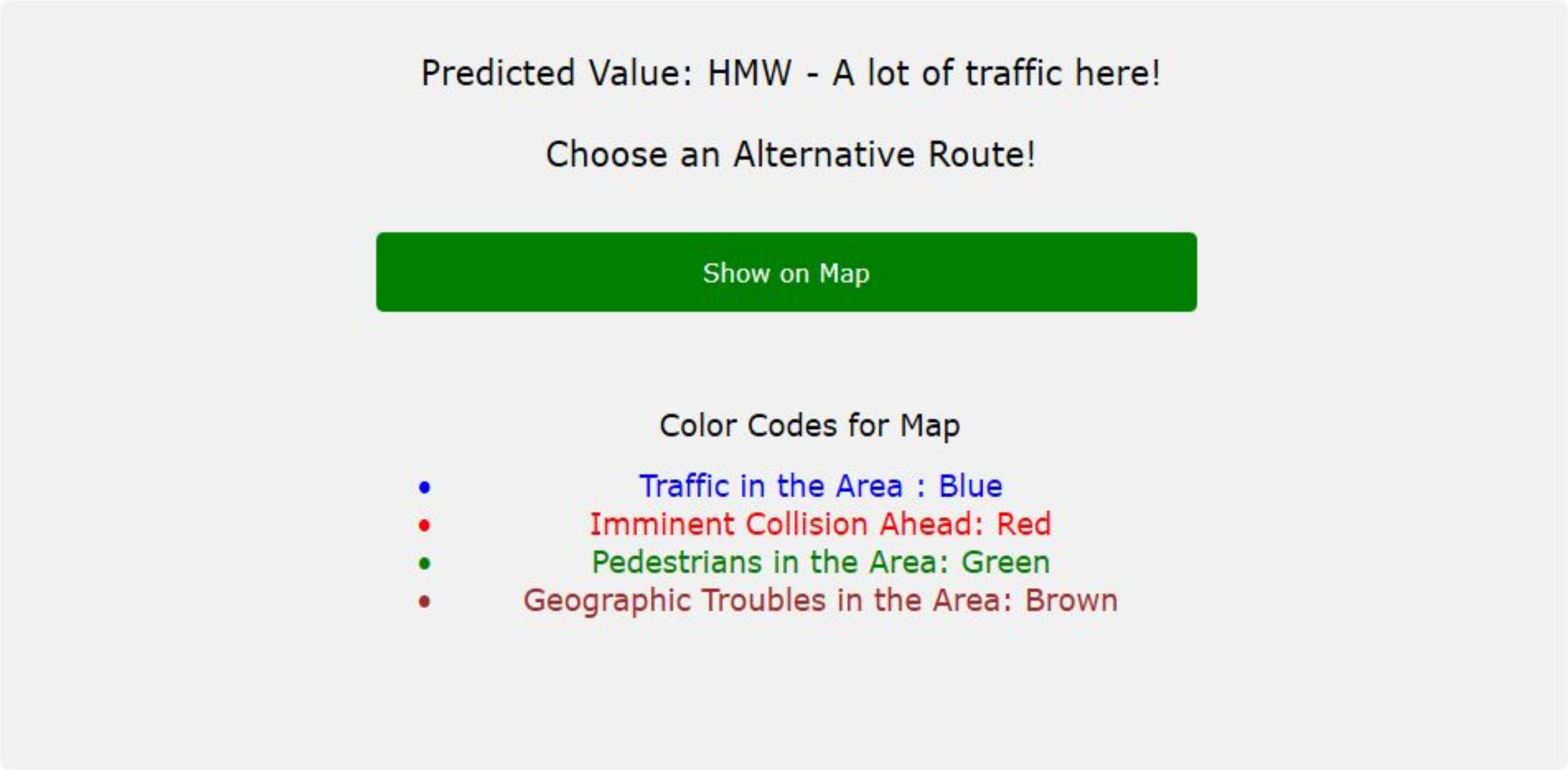
Predict with latitude and longitude.

The second web page as shown in the following slide, is where the results based on the user’s inputs will be shown.

- The predicted Alarm Type which would be used in that location, and whether there is a lot of traffic , pedestrians or if there are geographical troubles will be displayed, so that the user can watch out.

- We also provide a feature, where on clicking the button “Show on Map” , the latitude and longitude the user gave as an input, will be represented as a point on the map and the color indicates what type of traffic is in that particular location. The color code has also been mentioned.

result.htm



Predicted Value: HMW - A lot of traffic here!

Choose an Alternative Route!

Color Codes for Map

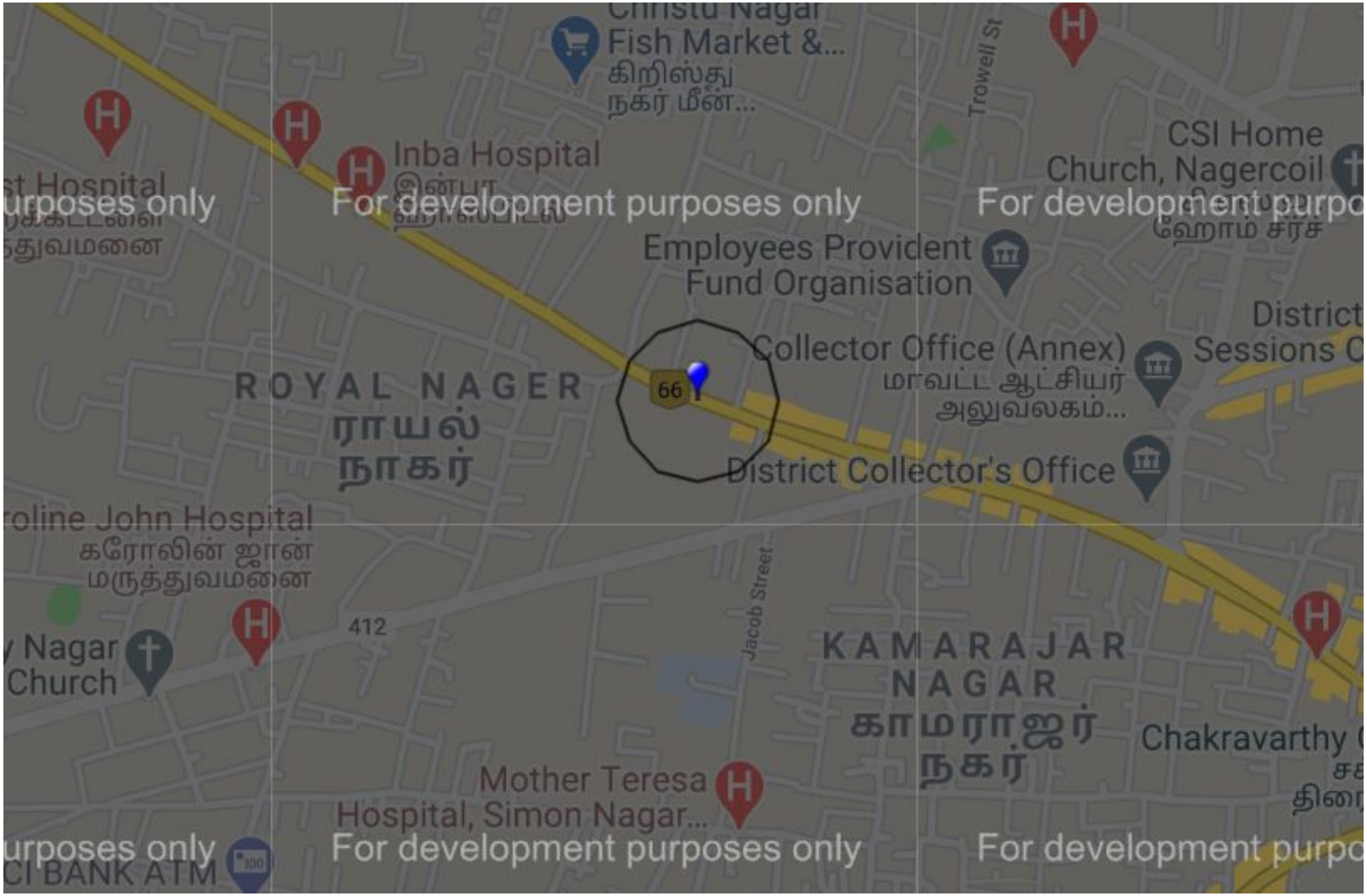
- Traffic in the Area : Blue
- Imminent Collision Ahead: Red
- Pedestrians in the Area: Green
- Geographic Troubles in the Area: Brown

Since, we have been given spatial data, we decided to play with that and try representing it on the map.

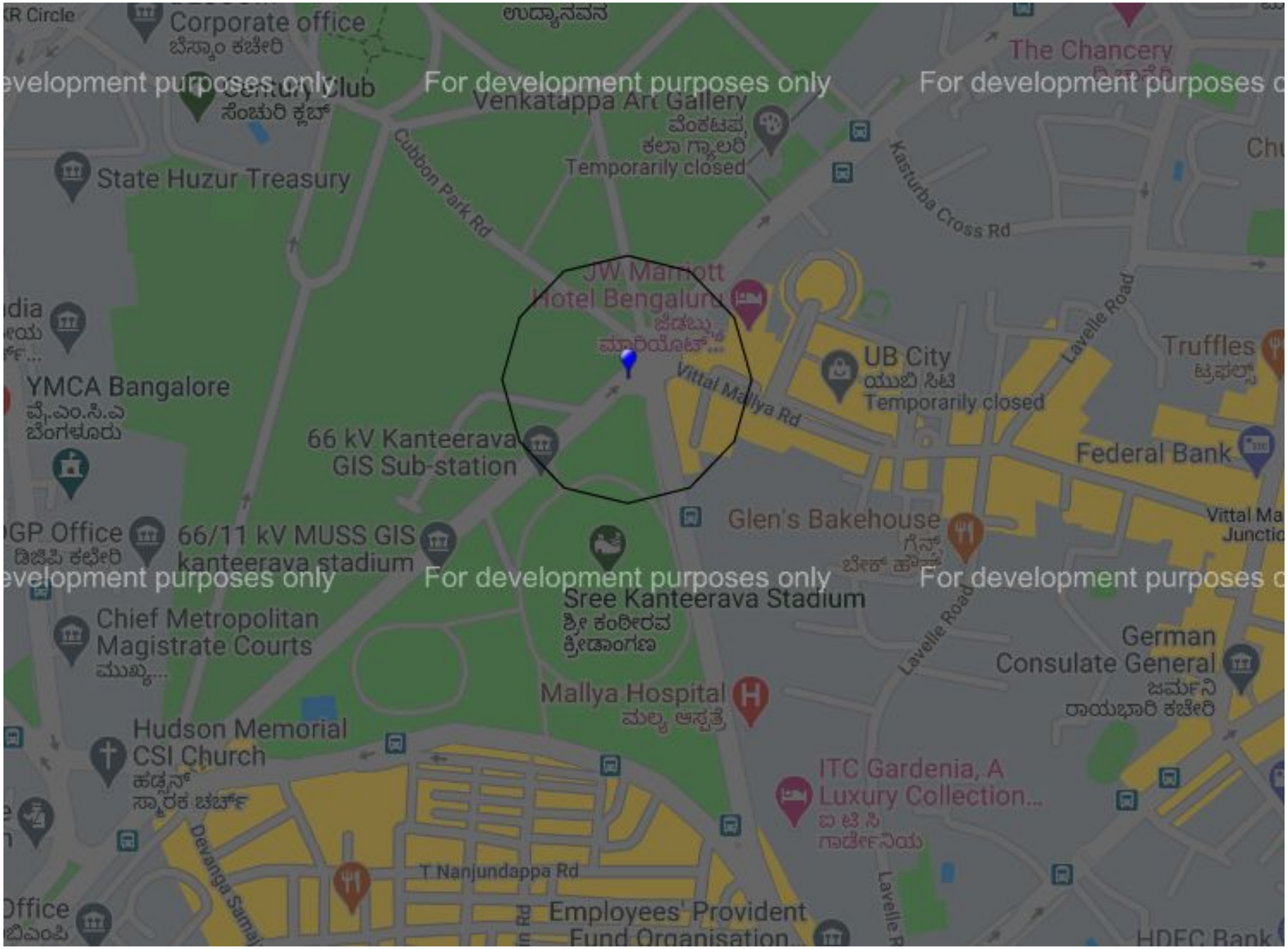
Based on the user’s input, we have represented the location as a point on the map, with the color indicating the type of traffic, as shown in the following slide, where blue indicates a lot of vehicles around.

As a part of our 100% model, we plan on adding another feature, where we will be representing the locations that are close to the location given as input and predicting traffic of those, as well.

map.htm
(output based on user’s input)



map.htm
(output based on user’s live location)



Progress as on Date (Prediction Model)

- The model accepts the latitude, longitude and time as input as classifies the alarm type.

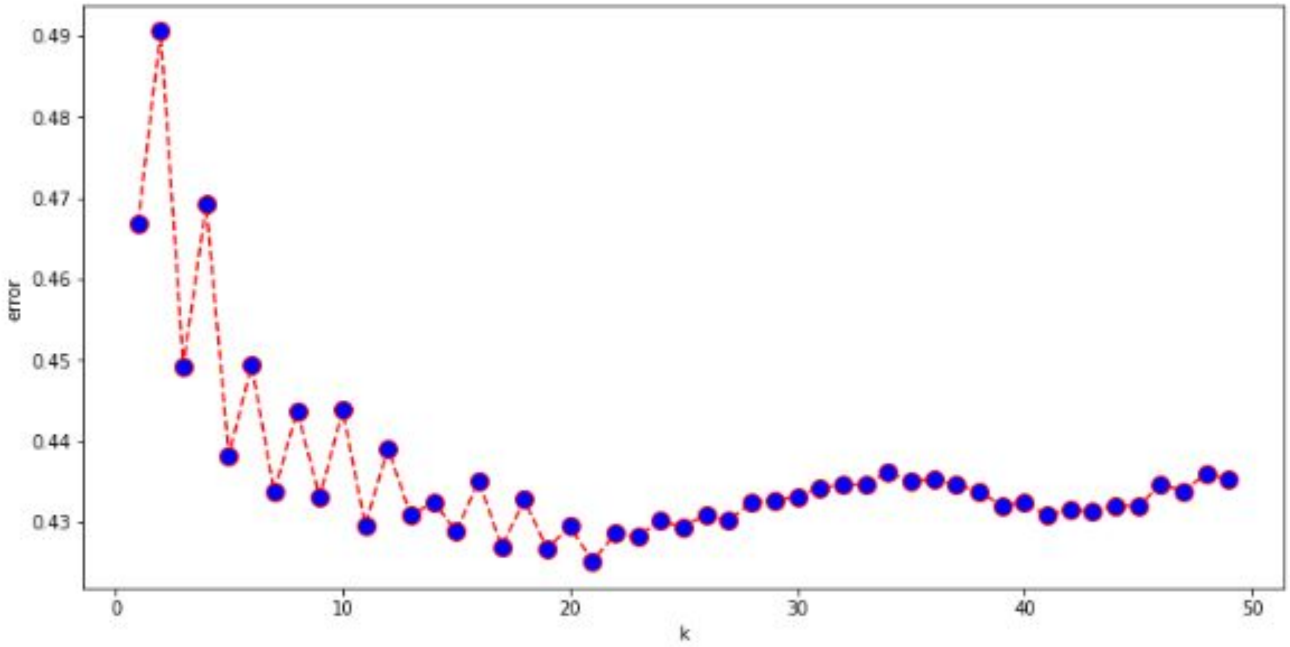
- It is trained with 90% of the cleaned and modified data with latitude, longitude and time as the features inorder the predict each of the 5 alarm types (class labels - encoded 0 to 4).

- Feature scaling is performed on the generated test and train data with the RobustScaler.

- The input is passed to the classifier developed using the KNN algorithm and the output (prediction) is returned.
  KNN is chosen post observing the training results with other algorithms - SVM, Naive Bayes and Decision tree. We have also trained the model with ANN.
  The most promising results were provided by KNN.

- The KNN model is constructed using the sought-after scikitlearn library’s KNeighborsClassifier module.

- A suitable K value (k=21) is selected on a trial and error basis. However, to improve the results the following graph is plotted between mean error and corresponding k.

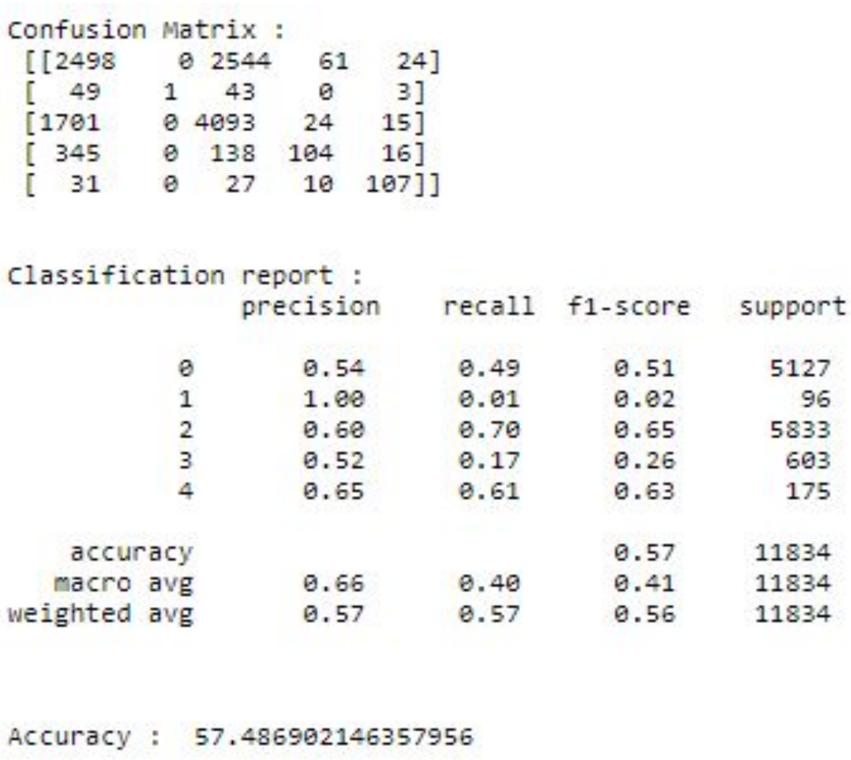


- The predicted results are generated from the testing data.

- Finally, the evaluation of the model is performed by comparing the predicted values and the actual test data (10%) in terms of confusion matrix for false positives and false negatives, classification report and accuracy (57%).

- Increasing the data shall generate better results. However, we shall try to improve the accuracy by next milestone.

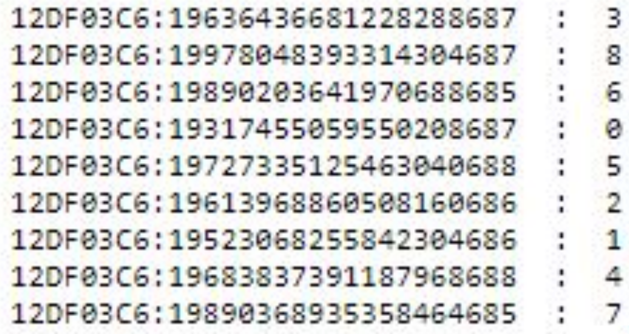
- The following are the evaluation metrics.



```
Confusion Matrix :
 [[2498    0 2544   61   24]
 [  49    1   43    0    3]
 [1701    0 4093   24   15]
 [ 345    0  138  104   16]
 [  31    0   27   10  107]]

Classification report :
               precision    recall  f1-score   support

           0       0.54      0.49      0.51      5127
           1       1.00      0.01      0.02        96
           2       0.60      0.70      0.65      5833
           3       0.52      0.17      0.26       603
           4       0.65      0.61      0.63       175

    accuracy                           0.57     11834
   macro avg       0.66      0.40      0.41     11834
weighted avg       0.57      0.57      0.56     11834


Accuracy :  57.486902146357956
```

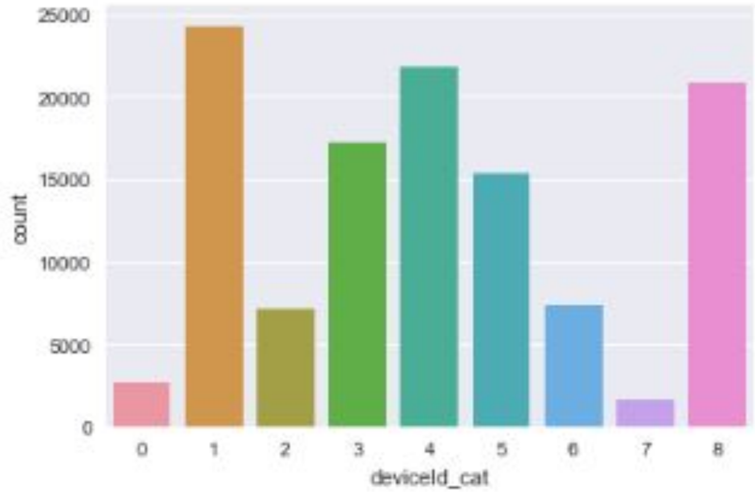
Progress as on Date (Analysis of vehicles + GUI)

- Basic analysis with vehicle ids.

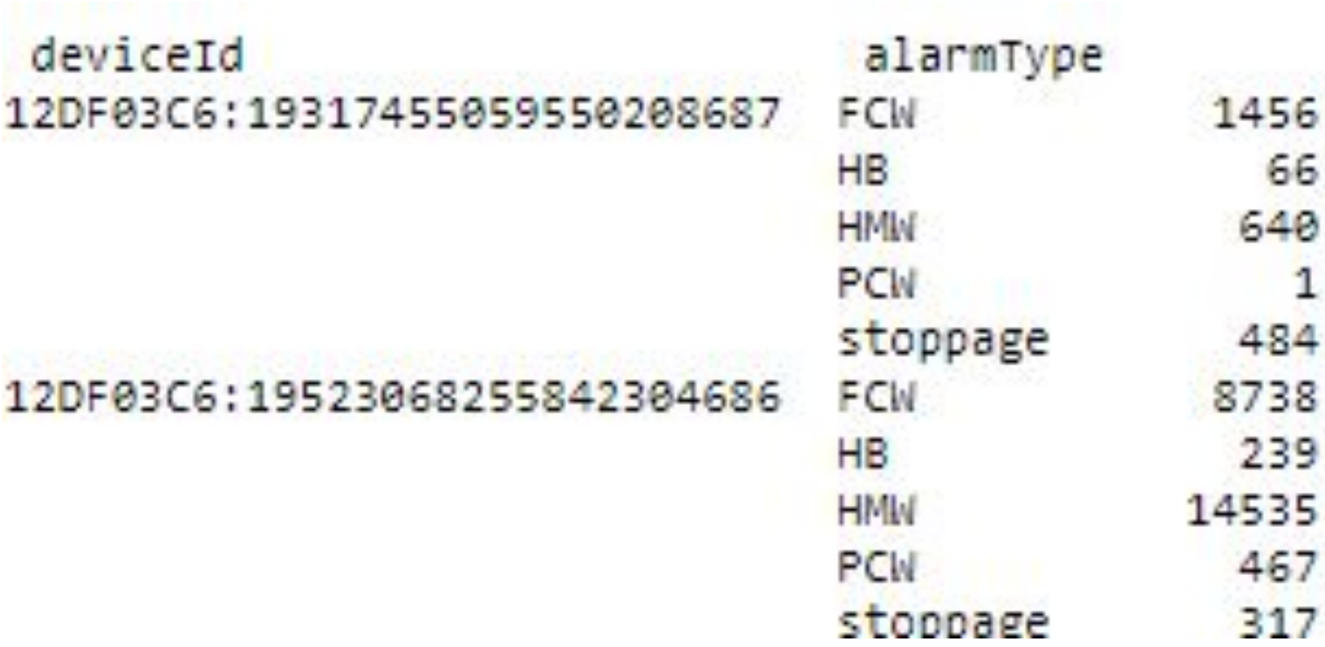


```
12DF03C6:19636436681228288687  :  3
12DF03C6:19978048393314304687  :  8
12DF03C6:19890203641970688685  :  6
12DF03C6:19317455059550208687  :  0
12DF03C6:19727335125463040688  :  5
12DF03C6:19613968860508160686  :  2
12DF03C6:19523068255842304686  :  1
12DF03C6:19683837391187968688  :  4
12DF03C6:19890368935358464685  :  7
```

vehicle density

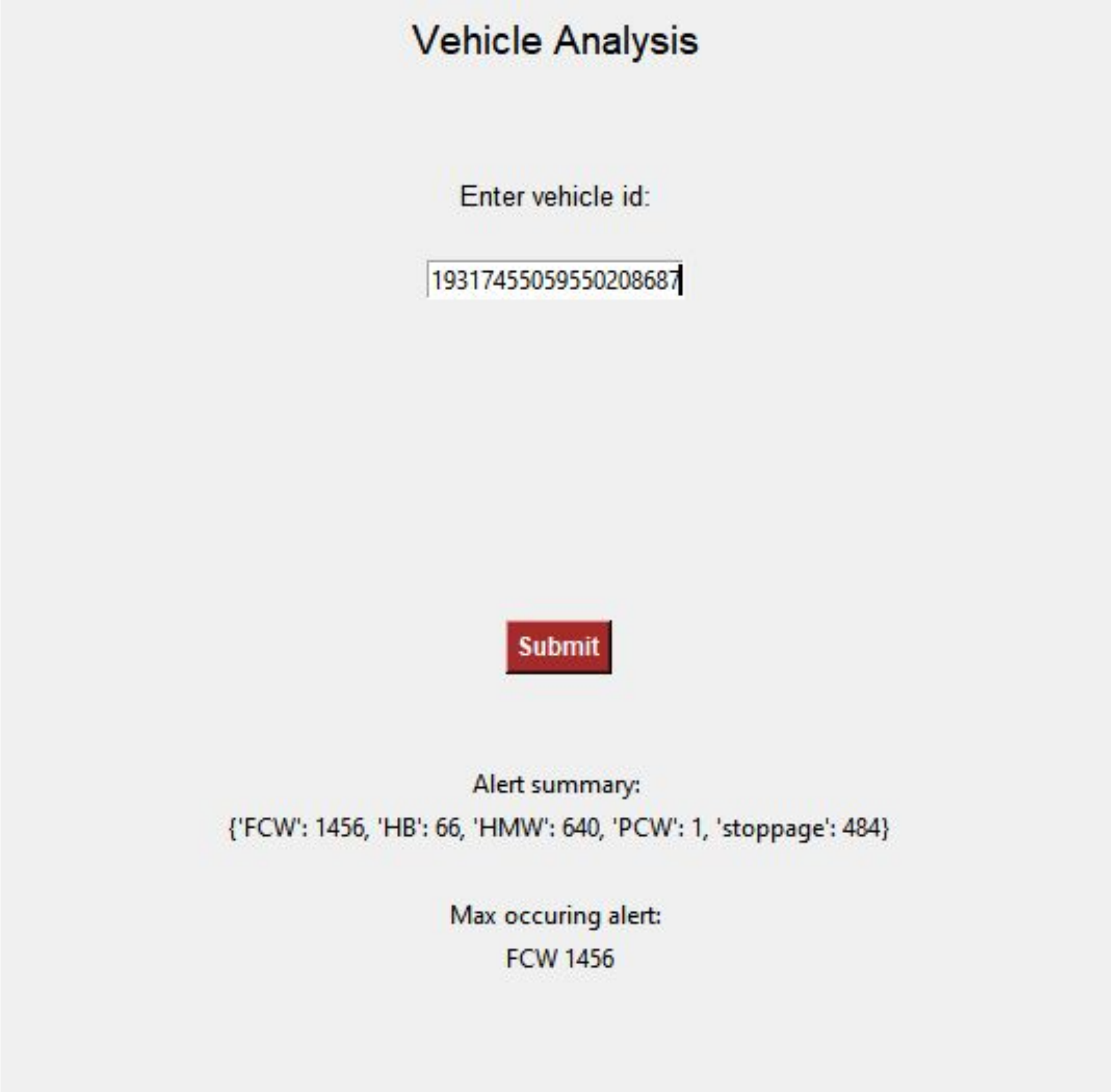


- The alarm type frequencies for each vehicle is calculated.



```
 deviceId                       alarmType
12DF03C6:19317455059550208687  FCW              1456
                               HB                 66
                               HMW               640
                               PCW                 1
                               stoppage          484
12DF03C6:19523068255842304686  FCW              8738
                               HB                239
                               HMW             14535
                               PCW               467
                               stoppage          317
```

- With a simple function, the alert with maximum frequency is displayed for the given device Id in quite a minimalistic way using ‘tkinter’.



Vehicle Analysis

Enter vehicle id:

19317455059550208687

Alert summary:

{'FCW': 1456, 'HB': 66, 'HMW': 640, 'PCW': 1, 'stoppage': 484}

Max occuring alert:

FCW 1456

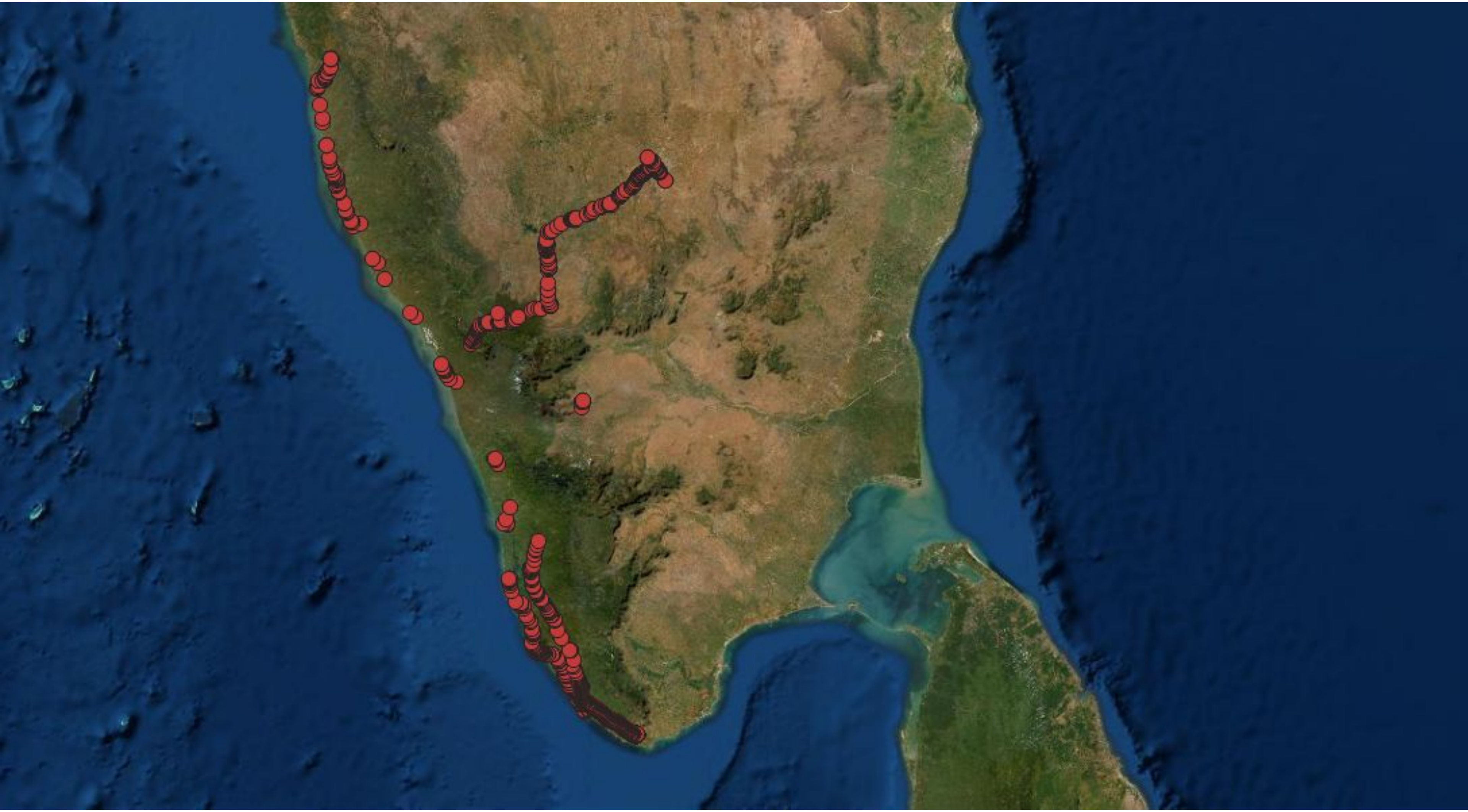
Key Visualizations

We plotted the graphs and maps as following, which helped us get a better understanding of the data we have been given,

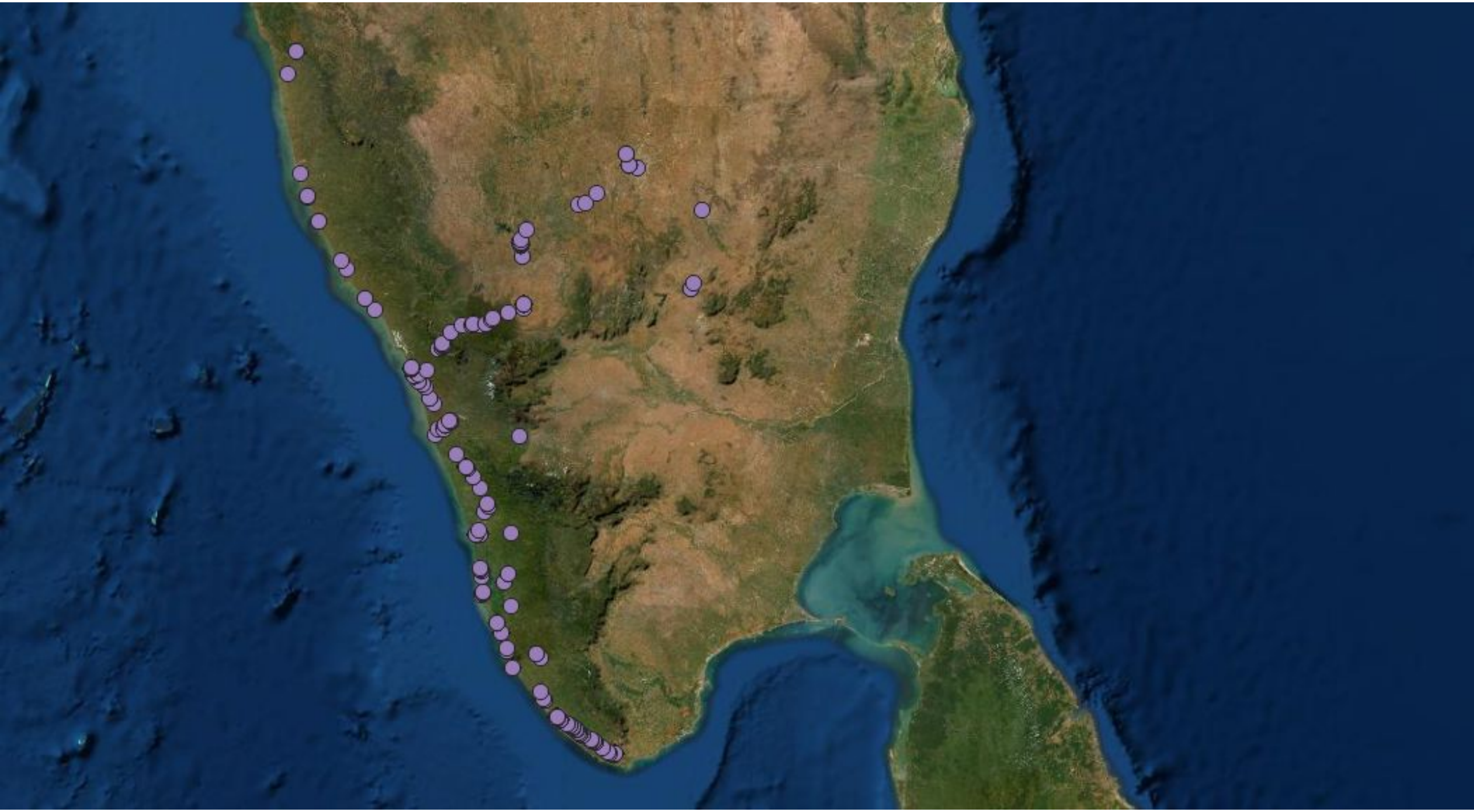
1. Alarm Type distribution on the Indian Map
2. Bar Graph on the Alarm Type distribution for the given dataset.
3. Plots between each pair of attributes.

The maps in the following slides, show the distribution of the alarm types based on latitude and longitude, which is represented on the indian map. Using this, we came up with the idea to incorporate maps into our model.

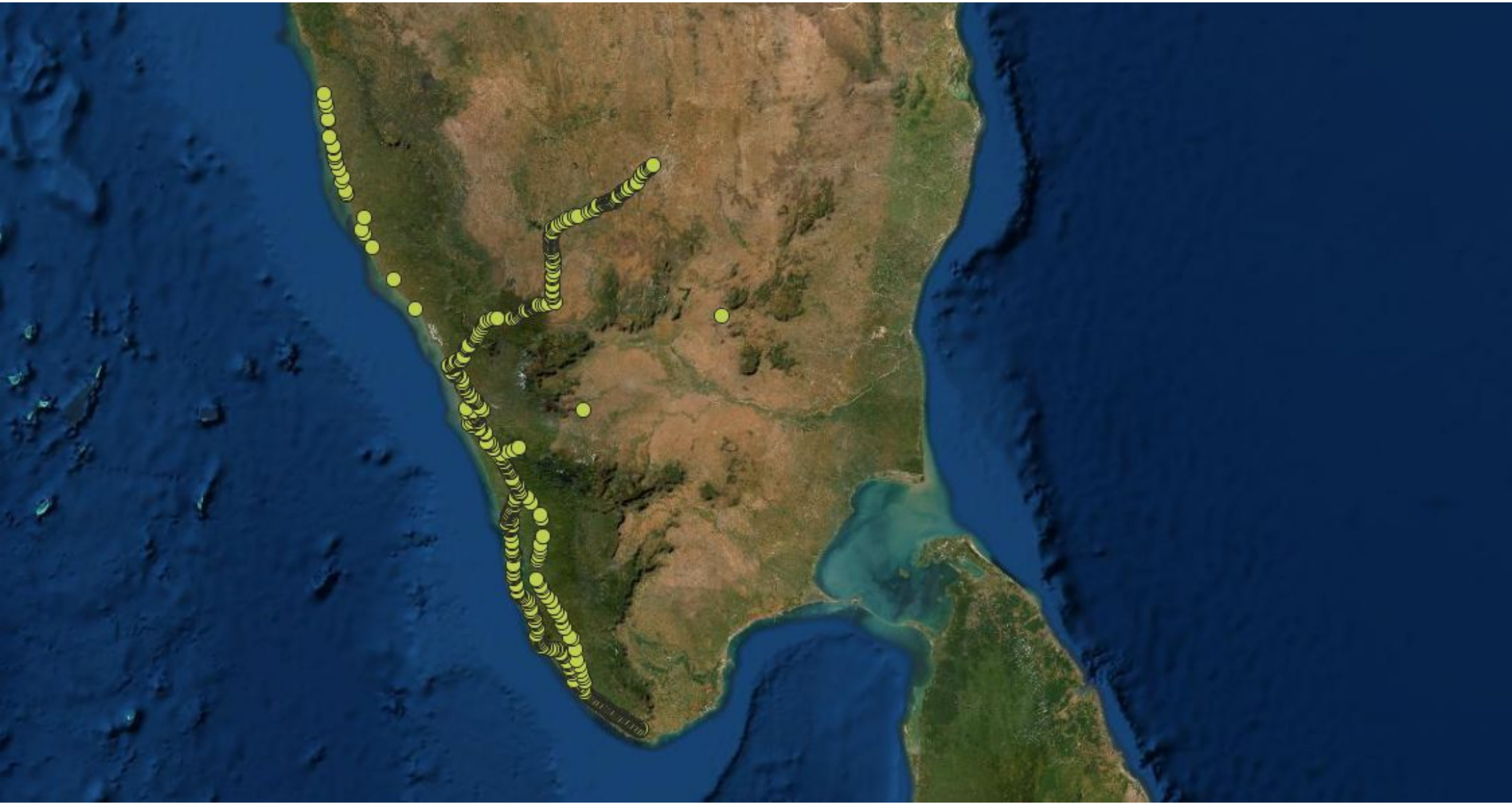
Alarm Type: PCW



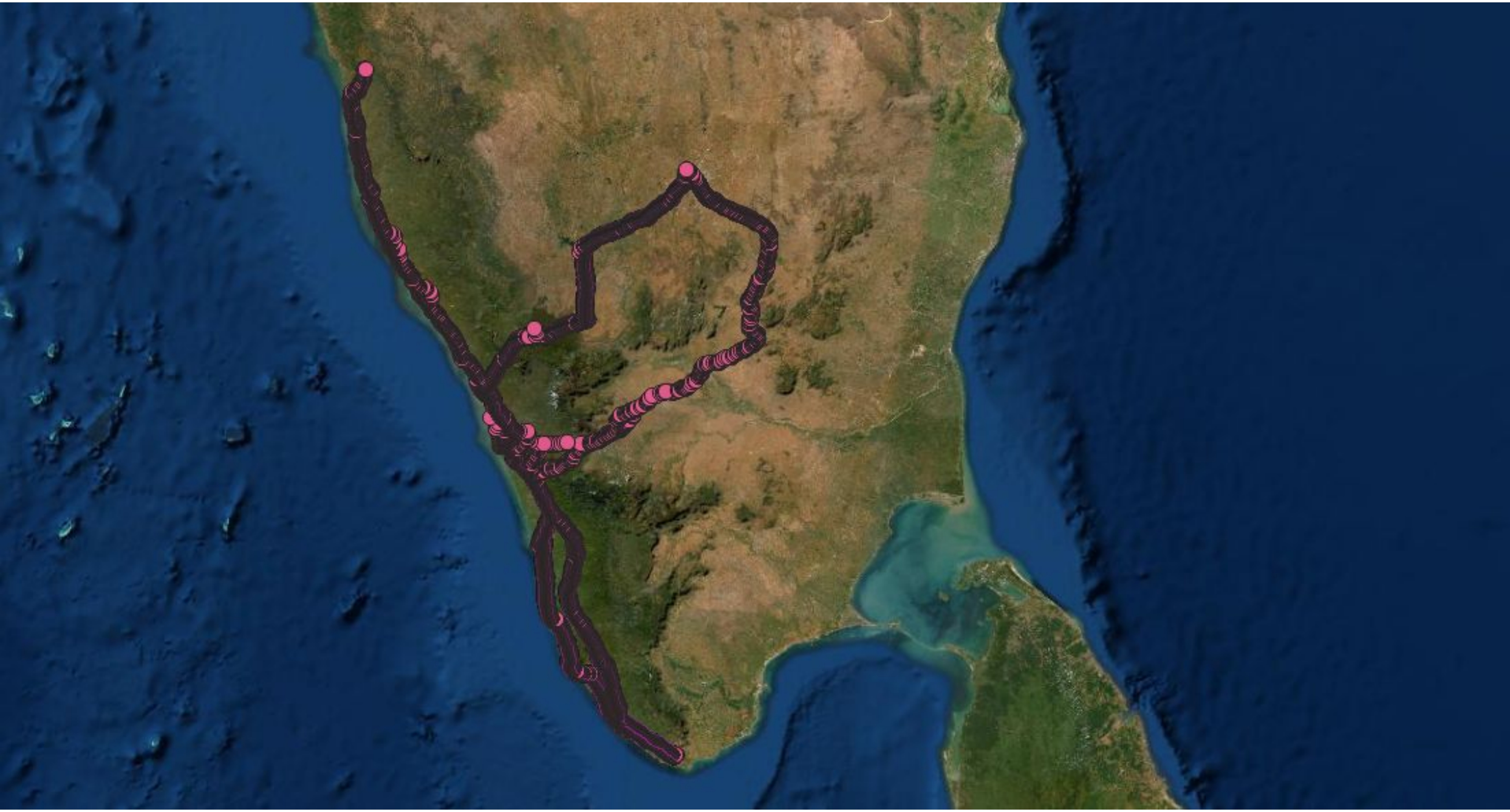
Alarm Type: Stoppage



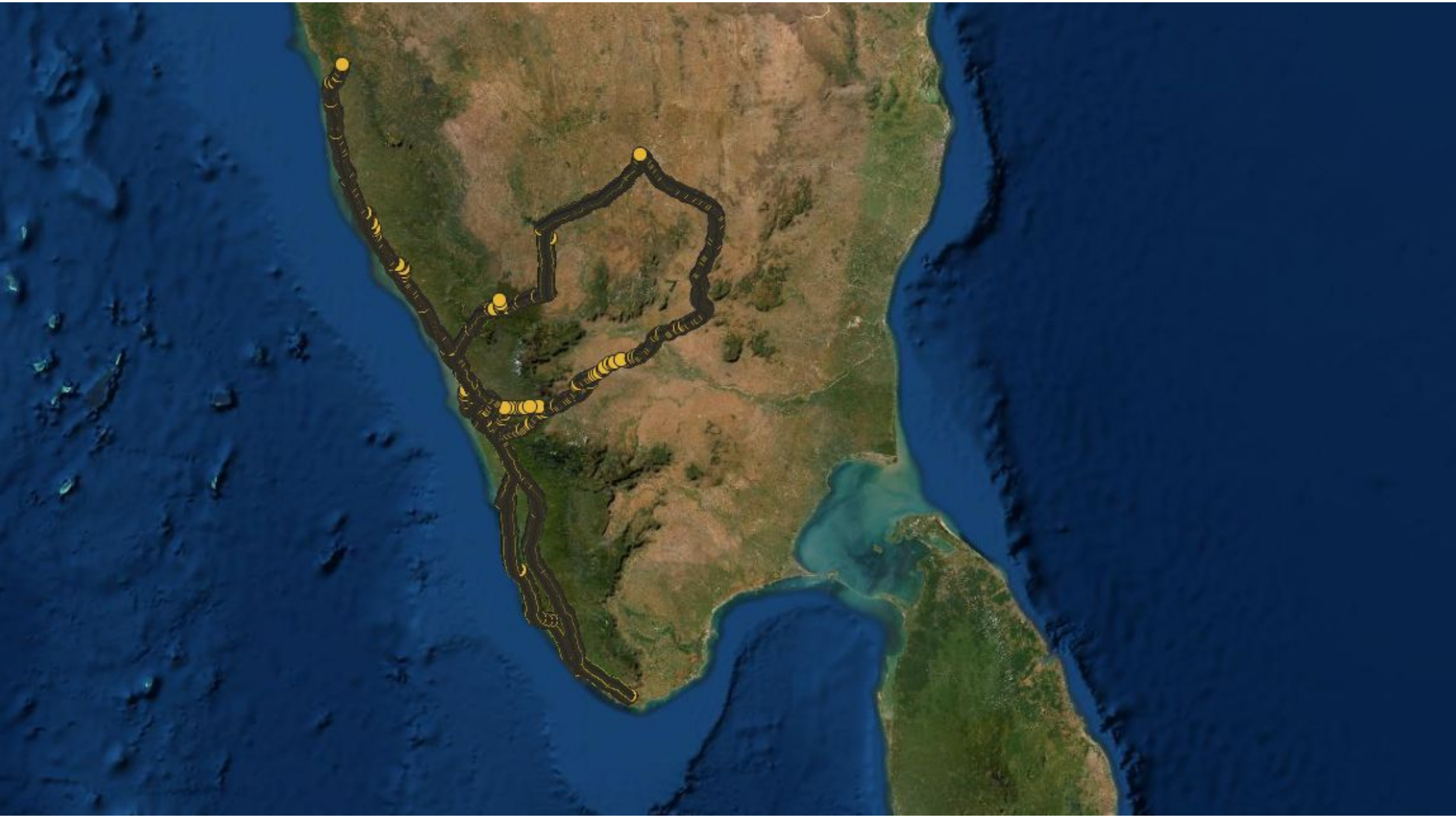
Alarm Type: HB



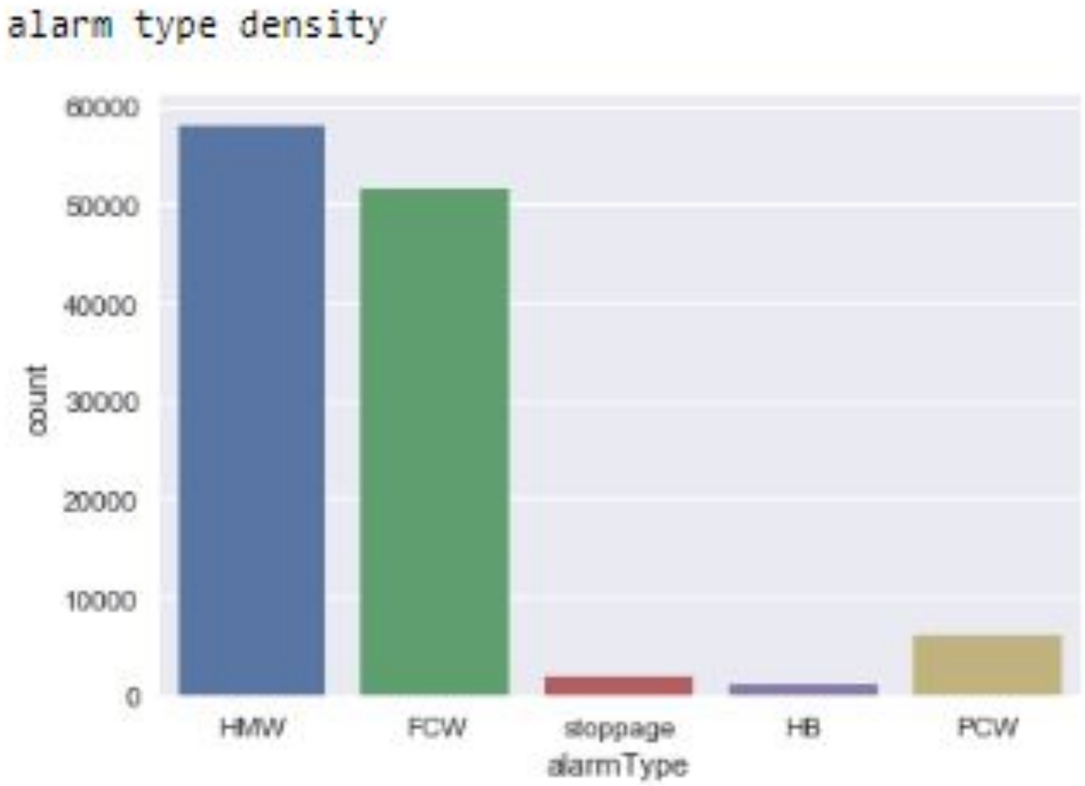
Alarm Type: FCW



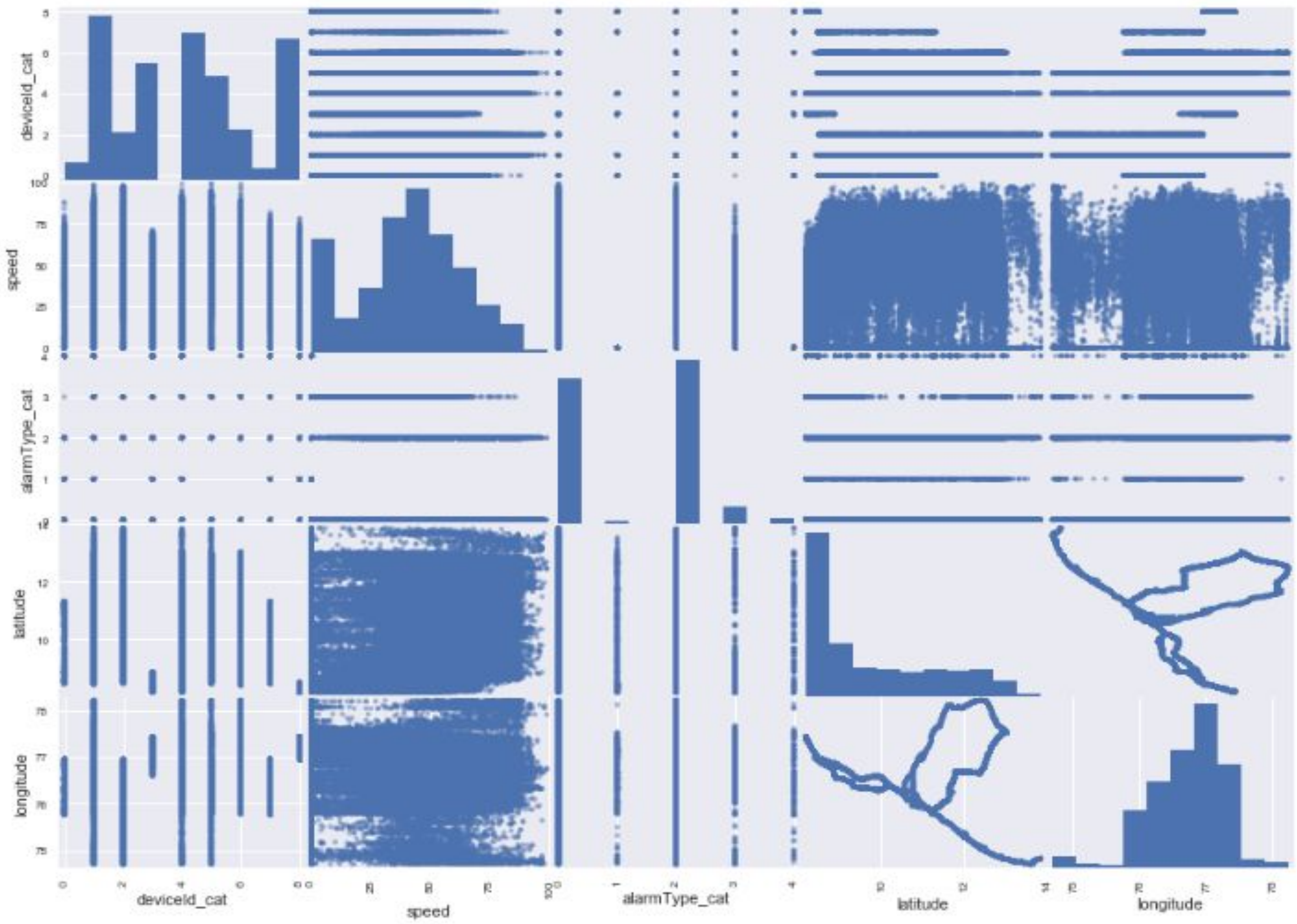
Alarm Type: HMW



- The following bar plot gives the distribution of each alarm type over the given data.



- Plots between each pair of attributes.



Tech Stack

lib/ package/ framework	version	lib/ package/ framework	version
Python	3+	Flask	1.1.1
HTML + CSS	5	JavaScript	V8 8.3.110.13
Geopandas	0.6.1	Pygmaps	0.1.1
Scikit-learn	0.22.1	Pickle	4.0
Seaborn	0.10.0	Matplotlib	3.1.3
Geocoder	1.38.1	tkinter	8.6
Numpy	1.18.1	Pandas	1.0.3

Future Work

As a part of future work, we would like to implement real time diagnostics on the vehicles to calculate traffic intensity provided we get live data of the alerts.

This can be immensely helpful when stuck in high density traffic situations (taking too long to be cleared with many vehicles) as these alert types are activated when the distance between the vehicles is less than the threshold.

The idea is to estimate the number of vehicles with active HMW / FCW as predict the distance.

The prerequisite is that, at least 60-70% vehicles should have the alert system in the vehicle for reasonable results.

Key Deliverables for the next milestone

The key points that will be focussed on,

1. When the user enters the values for a given location, traffic around that point will also be displayed on the map. This will help the user get a better idea on which path to take to get a smooth travel experience.

2. The accuracy of our current model shall be improved.

Summary

- Our model takes in latitude, longitude and time as input and predicts both pedestrian and traffic density and probable geographical disturbances in turn trying to reduce road mishaps.

- After applying many algorithms it was concluded that K nearest neighbour algorithm shows the best accuracy.

- A user friendly GUI is implemented which can accept the inputs and give the prediction along with its representation on google map.

- Easy to understand visualisations are included to get a better visual clarity of the data.

Thank you!