PROBLEM STATEMENTS

ENSURE SAFE DRIVING DISTANCE IN ADVERSE CONDITIONS

ORGANIZATION
ARAI

TEAM NAME
DREAM SLAYERS

TEAM LEADER
HARSHIT MAHAJAN

COLLEGE CODE
1-3512549572

1. Alerting the user about the accident prone area

1.1 A dataset (consisting of the no. of accidents that had taken place at particular coordinates, time of occurrence) would be used to warn the driver.

1.2 The user will be notified about danger, when they reach such an area, through an Android App.

1.3 Most accident prone areas would be displayed in RED on the Google Maps.

TECHNOLOGY STACK

- **Python**
- **Tensorflow**
- **Keras**
- **Flask**
- **Heroku server**
- **PyGame**
- **Firebase**
- **Google Maps API**
- **minSDK version19**

2. Avoiding collisions on highways

2.1 A RADAR would be used to fetch real time data,(hypothetical data in this case)

2.2 We have created an ML model ,which accepts the position coordinates and velocity of the vehicle in our vicinity and our own velocity and predicts the best lane and optimal speed for the vehicle to proceed.

2.3 We deploy our website on Heroku Server using Flask(backend).

2.4 We have used Keras with Tensorflow backend to design a regressor.

2.5 We used post request at the Heroku Server through our Android app to fetch our predictions.

2.6 We have successfully implemented an algorithm to avoid any possible collision considering all the scenarios in a highway.

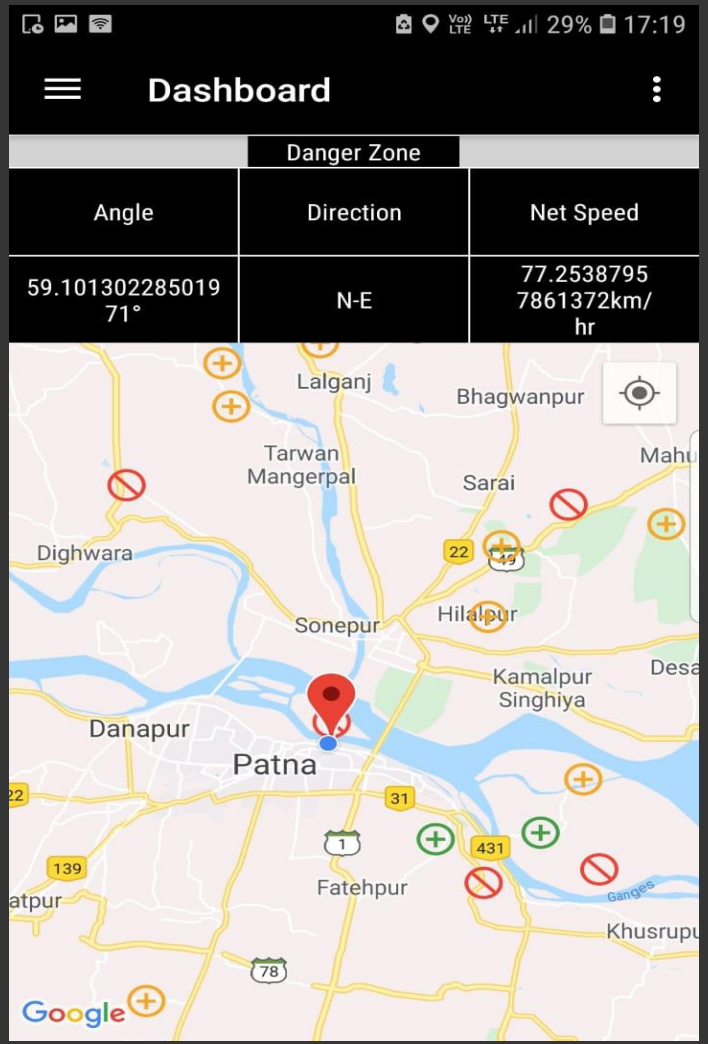
Machine Learning Model

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

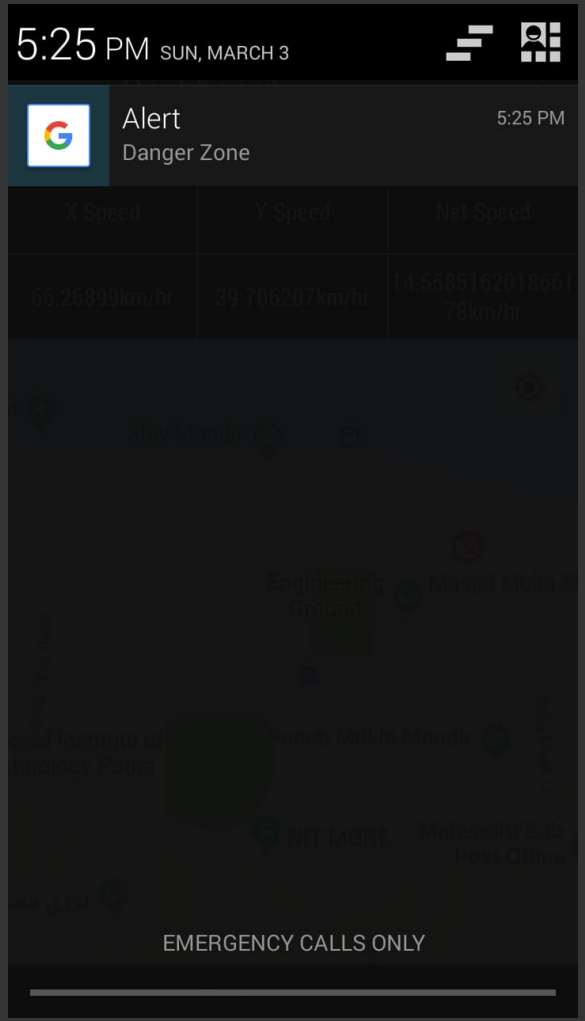
```
In [7]: xtest.shape
Out[7]: (250, 9)

In [8]: len(ytest) , len(ytrain)
Out[8]: (250, 749)

In [9]: xtrain.shape
Out[9]: (749, 9)

In [10]: model.add(Dense(128, kernel_initializer='normal',input_dim = xtrain.shape[1], activation='relu'))
         model.add(Dense(256, kernel_initializer='normal',activation='relu'))
         model.add(Dense(256, kernel_initializer='normal',activation='relu'))
         model.add(Dense(256, kernel_initializer='normal',activation='relu'))
         model.add(Dense(2, kernel_initializer='normal',activation='linear'))

WARNING:tensorflow:From /home/nilesh/anaconda3/lib/python3.7/site-packages/tensorflow/python/framework/op_def_lib
rary.py:263: colocate_with (from tensorflow.python.framework.ops) is deprecated and will be removed in a future v
ersion.
Instructions for updating:
Colocations handled automatically by placer.

In [11]: model.summary()

_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 128)               1280
_________________________________________________________________
dense_2 (Dense)              (None, 256)               33024
_________________________________________________________________
dense_3 (Dense)              (None, 256)               65792
_________________________________________________________________
dense_4 (Dense)              (None, 256)               65792
_________________________________________________________________
dense_5 (Dense)              (None, 2)                 514
=================================================================
Total params: 166,402
Trainable params: 166,402
Non-trainable params: 0
_________________________________________________________________

In [12]: model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mse'])

In [13]: model.fit(xtrain , ytrain , epochs = 200 , validation_data = (xtest , ytest))
```

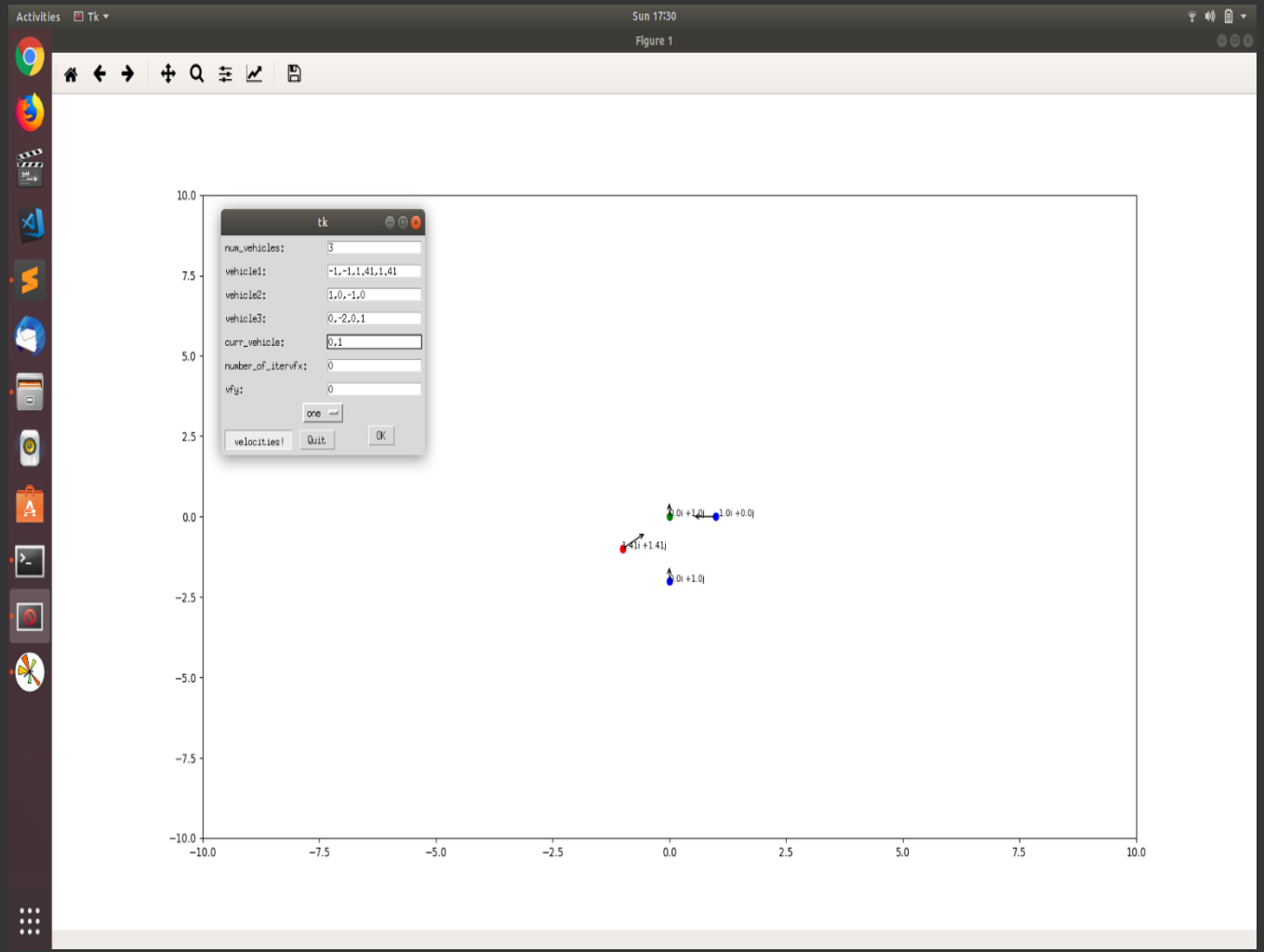
Functioning of app



- We will use HTTP post request which will fetch data from the server
- Server will process the current situation and send the optimum velocity that would be predicted through a Machine Learning algorithm
- Data points of different colours are to represent accident prone regions with different probabilities
- RED : highest probability
- YELLOW : moderate probability
- GREEN : least probability

We will use real time notification which will be updated every 50ms. Whenever a user enters a danger zone he will be warned with a notification and an alarm



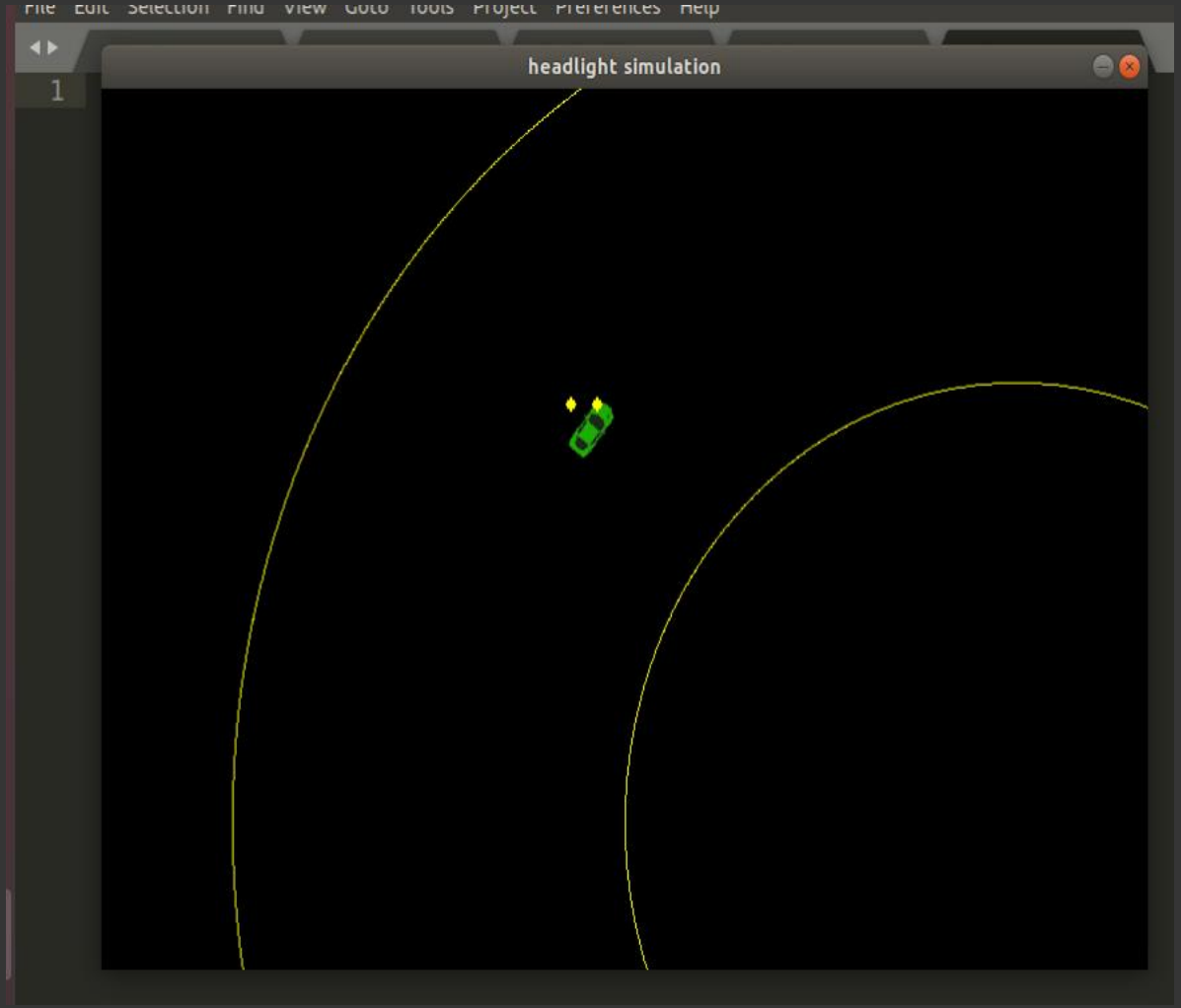


Alternative Algorithm

We compute the final velocity which our car should obtain so as to avoid collisions with nearby vehicles taking into consideration distance vectors , velocity vectors and their dot products.

Future scope of our project

We also plan on incorporating a system of adaptive headlights to tackle adverse weather conditions. We have started work on it !



THE END >_<