# Team: PirateMiner

**Team Members:** Sanjay Bhargav Madamanchi

## Project Name: Traffic Sign Recognition

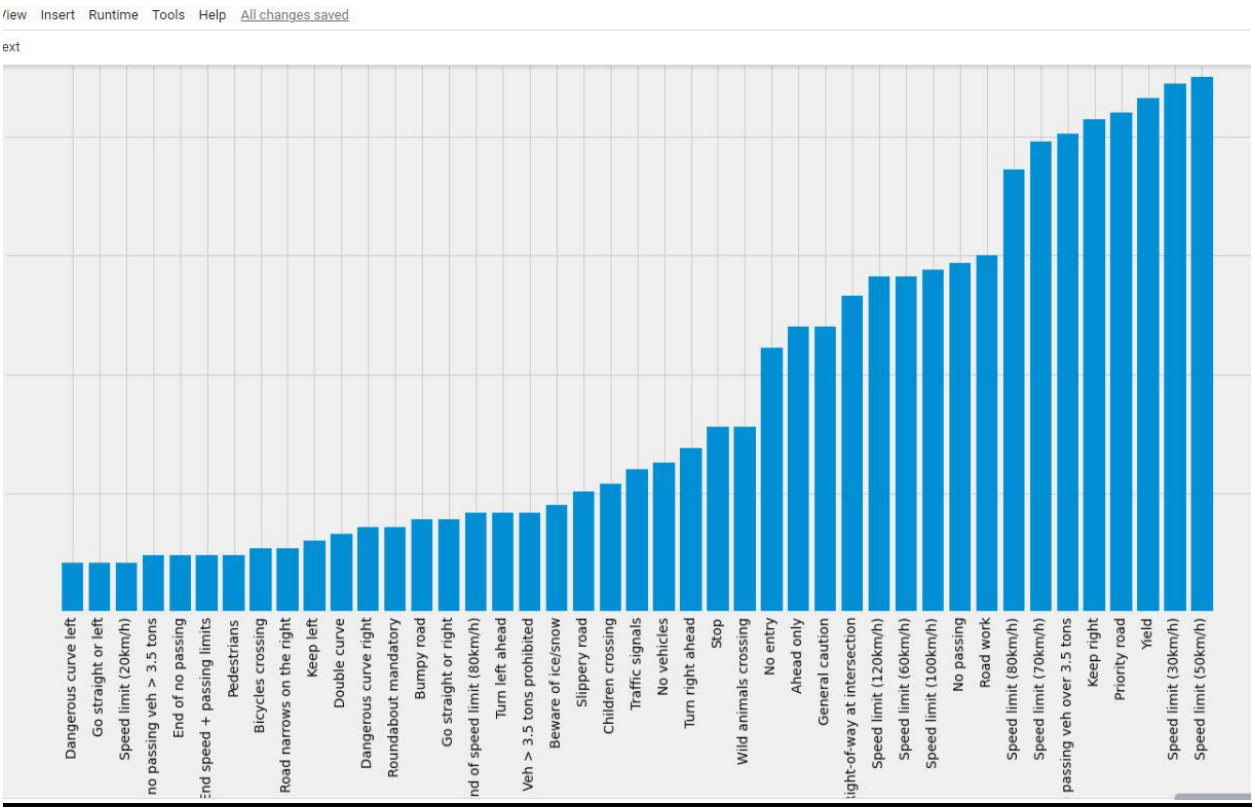
## Data Science Approaches and Algorithms:

1. **Sign Recognition:** For recognizing the traffic signs.
    - At first we use Keras to implement 2D convolution networks
    - Then we use max pooling to reduce and highlight parameters for the image
    - Using Batch Normalization to normalize the inputs
    - Flattening the layers to convert to a single dimension
    - Changing the dimensions of the given vector using Dense Layer
    - Adding a dropout laver to prevent over fitting

2. **CNN Model:** We have one Kaggle dataset of different traffic signs. We take the images and classify all the images present in the dataset and classify all the data into various classes based on the features used and visualizing the data based on the classes created

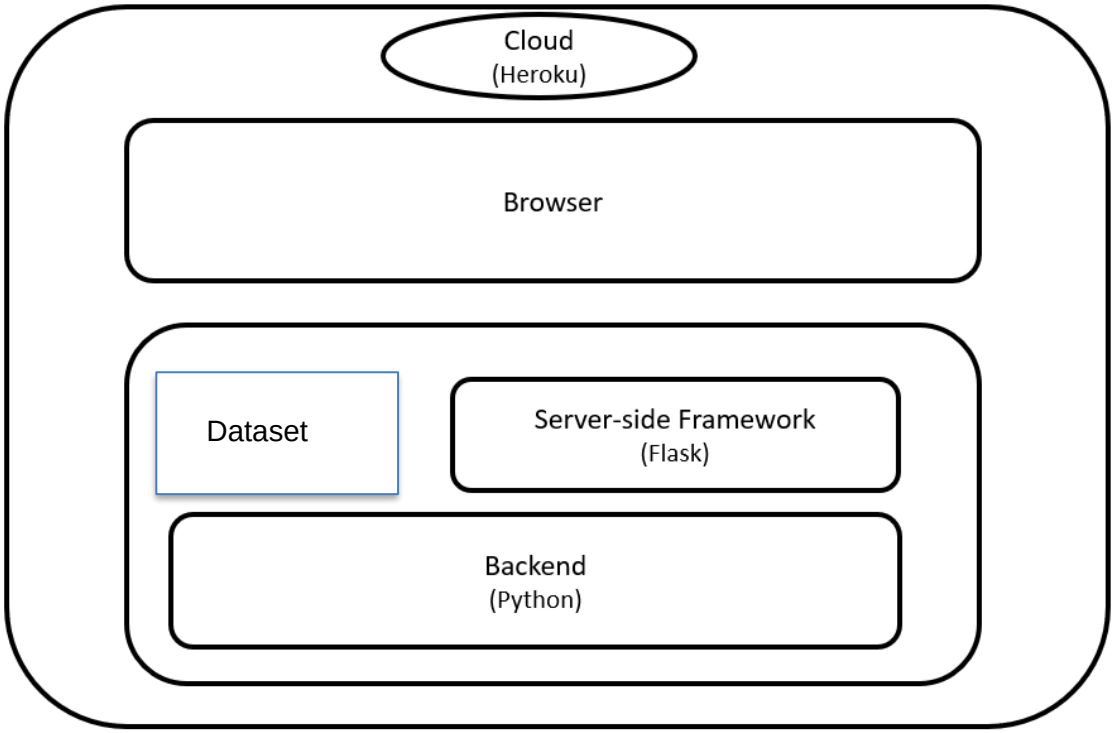
    **a.** We are training the model in the Exploratory analysis phase.
    **b.** The Dataset is, hereby, divided into training and testing datasets with number of records in the ratio 4:1.
    **c.** Then a sequential model is defined and the different types of layers like 2D convolution, dense layer, MaxPool layer etc. are added and in this way, the architecture of the model is defined
    **d.** Thereafter, the model is compiled where “Adam’s optimizers” is used and loss function defined here is categorical cross entropy

## Features Used




We are using the classes as the features to classify the various images into 43 different classes based the available images in the dataset. Speed limit(50kmph) has the highest number of images inside the given dataset whereas the images of dangerous curves has the least number.

## Client Side Design




### Cloud

Application deployed into the cloud is the shell of entire client centric development which makes it as a seamless interaction between user and application

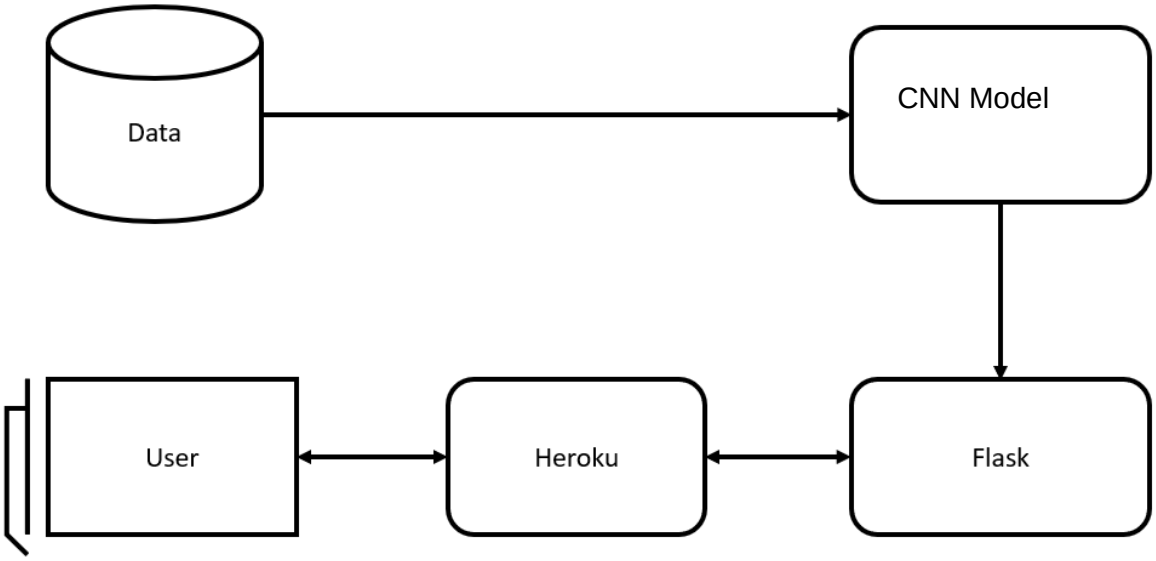
### Browser

Most common and feasible component for a web application and Edge, Opera, Firefox, Chrome, Safari, Safari mac, Firefox mac, Chrome mac are the major browsers while writing this document. As you can see, trying to build and test everything is difficult. Each browser has its own subtle nuances different in browser security, default font sizes, borders etc. All these issues can be overcome with the changes required in the programming.

### Programming

Programming with required components whether it can be frontend or backend captivates the users experience.
To make it a more user friendly and interactive interface it always moves in a frontend's direction. Frontend is a key that always captures customer's satisfaction to yield more income. Backend is always a hidden gem that performs it's actions to strengthen the application and makes it more efficient.

## Model Deployment



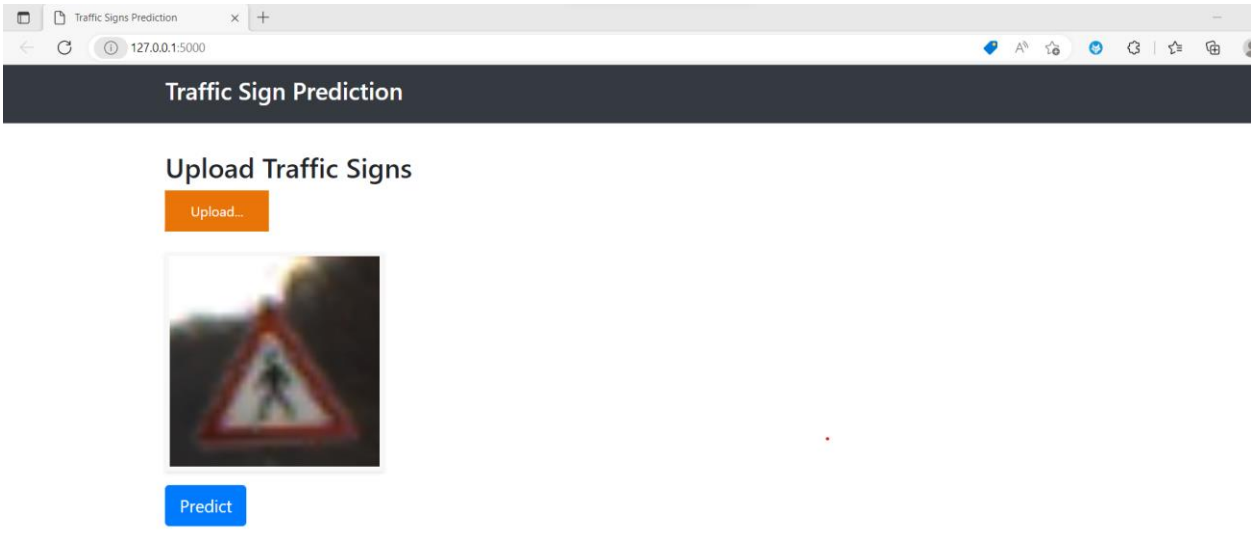

---

Fetching Training data ----------> training model ----------> Evaluating model ----------> Model Endpoint

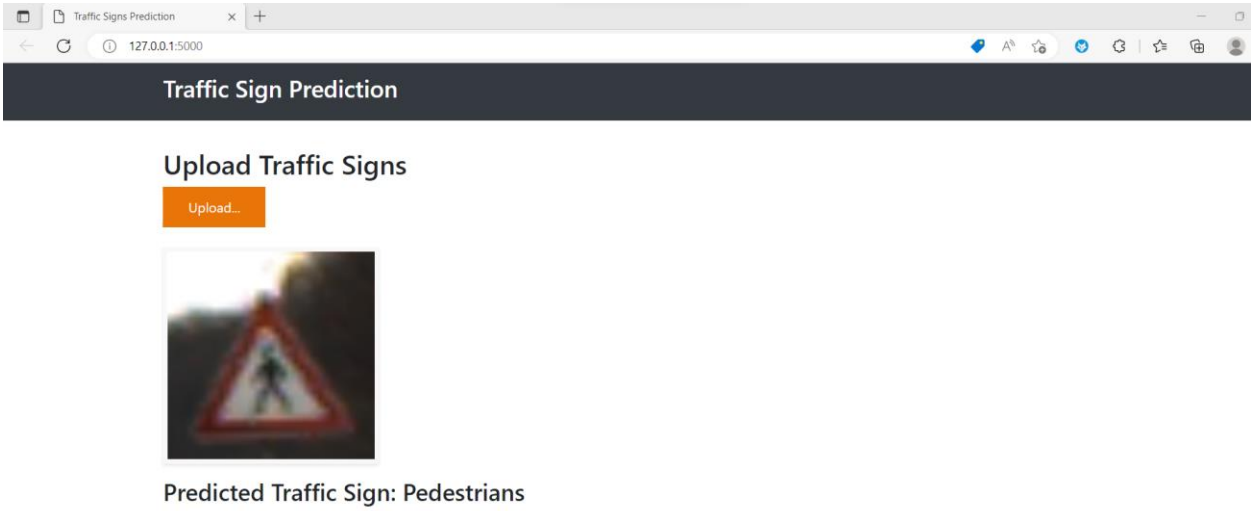
Fetching and training data from the trusted sources and then building a model to attain a classifier and using that h5 file and interface by making it a supervised approach takes it further by gathering inputs from users and to make a predictions with an accuracy in this spectrum looks an efficient build. Deploying the same into the cloud by compressing slug size without any compromises is an added advantage. We have used this to make a Traffic sign recognition model.

## Client Side Application: Web App

To make Interface more interactive, scalable and enjoyable we have picked Flask as our designers. As Flask is a micro-framework i.e with little to no dependencies to external libraries is the reason we picked it over Django and it is light, there are little dependency to update and watch for security bugs

We have crafted it to be simple yet elegant to use. The straight forward interface which has sentiment analyzer on the home screen and displays the result with a simple click.




## Project Colabs:

We have created a colab to do all the process regarding preprocessing and model evaluation and testing.

1**.Traffic_sign_recognition.ipynb**