Team Name: Deep Thinkers
Team Member: Gaurav Kumar Daharia
Team Member: Dhananjay Kumar
Theme of Hackathon: Financial Technology (AI/ML)
Email Id: gauravdaharia80@gmail.com

Problem statement

It is a good that now day’s people are supporting online payments and at the same time many third party are ensuring there transaction is safe and secure. But problem arises when these parties don’t make any autonomous changes which can detect the anomaly in the transaction because in today’s time no one want to wait for a second and public want everything to be quick and fast however people also have no idea about what can be the consequences that they can face, actually they don’t know there are many unauthorised users who are in the sight of these rush kind of scenario though which they can take the advantage in a form of fraud.

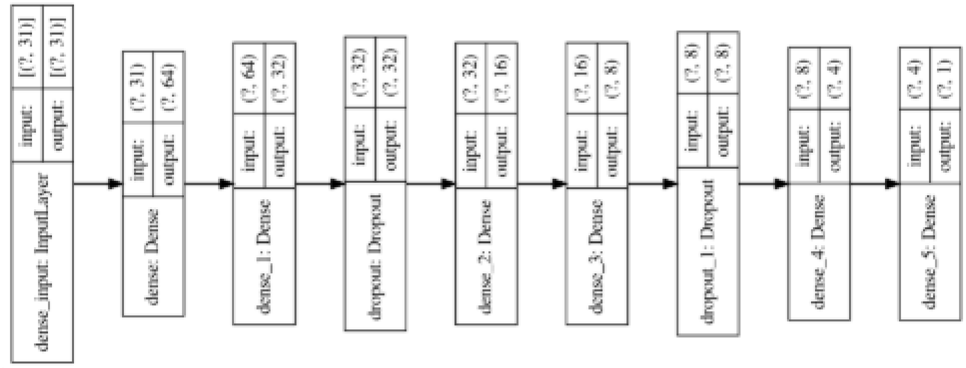
Objective

Our objective is to build a smart system with the help of deep learning and computer vision so that we can abort the frauds before it is going to occur. We have used a simple scenario based problem where a user is authenticated to proceed for further transaction only when the face of the user matches with the database saved image, because the things like OTP, finger prints etc. These are some user details that can be stolen from online transaction details but face is something which no one can stole until and unless someone try to do so. Now we have also used Deep learning for predicting whether the transaction is fraudulent or not, we have developed a **deep neural network** which will act like a watchdog for all transaction that a user is performing and it will classify whether the transaction is the authorized once or not. So in this way we are trying to give a double protection to the user so that no can face any frauds in future.

Implementation

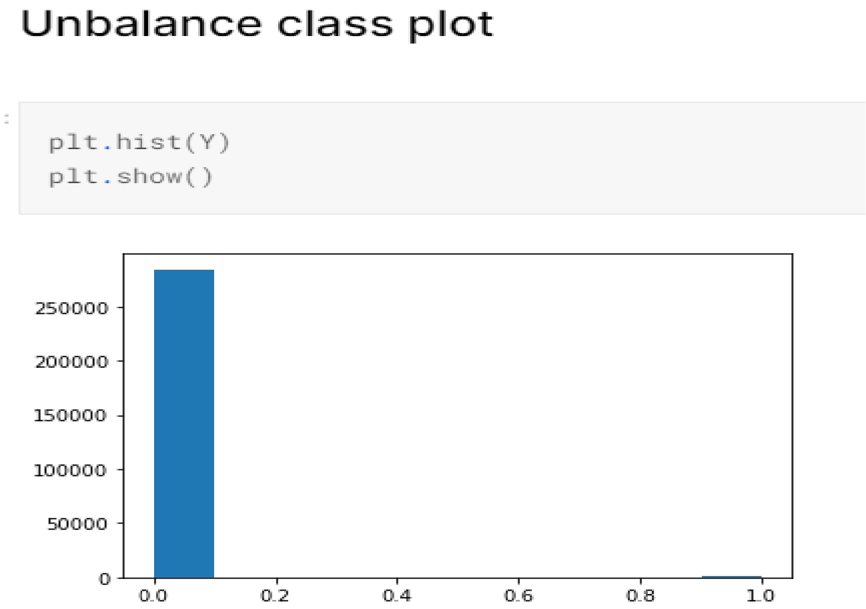
In this section we are going to share the detailed implementation of deep neural network and its performance.

1. **Neural Network Architecture**



We have used five layer dense neural network for developing my classification model. We have also used Dropout layer in between these network which controls and regulates model to train more effectively. This dropout layer help my model to learn different ways to extract values from given data points and classify it to its belonging class.

2. **Unbalanced Class Plot**



Unbalance class plot

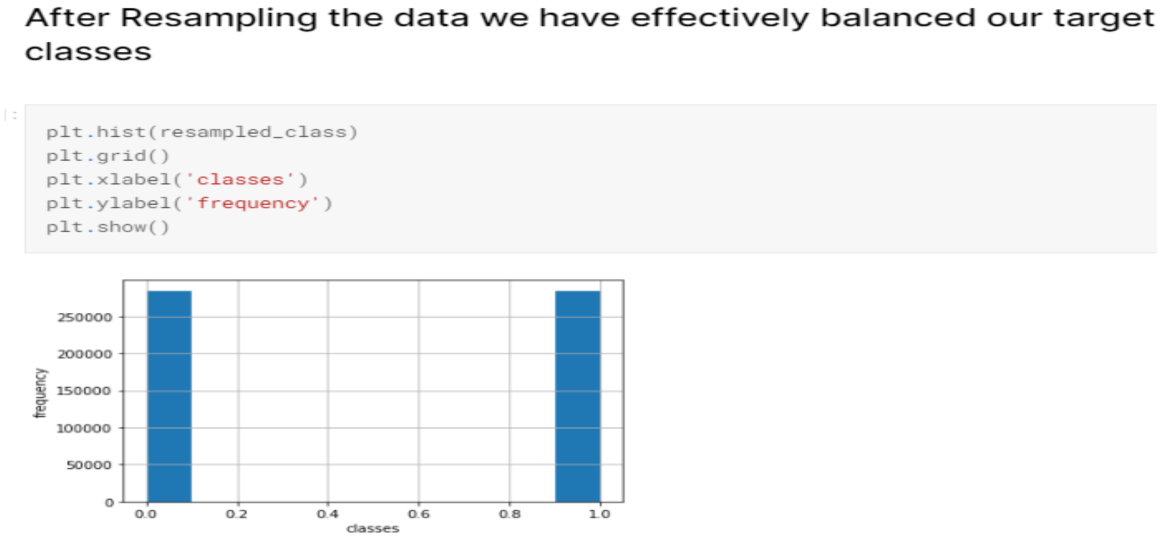
```
plt.hist(Y)
plt.show()
```

This is the plot of unbalance class distribution here:

- Zero: - genuine transaction class
- One: - Fraud transaction class

This was the major problem and in order to overcome with this problem we have resampled the dataset.

3. **Resampled Class Plot**

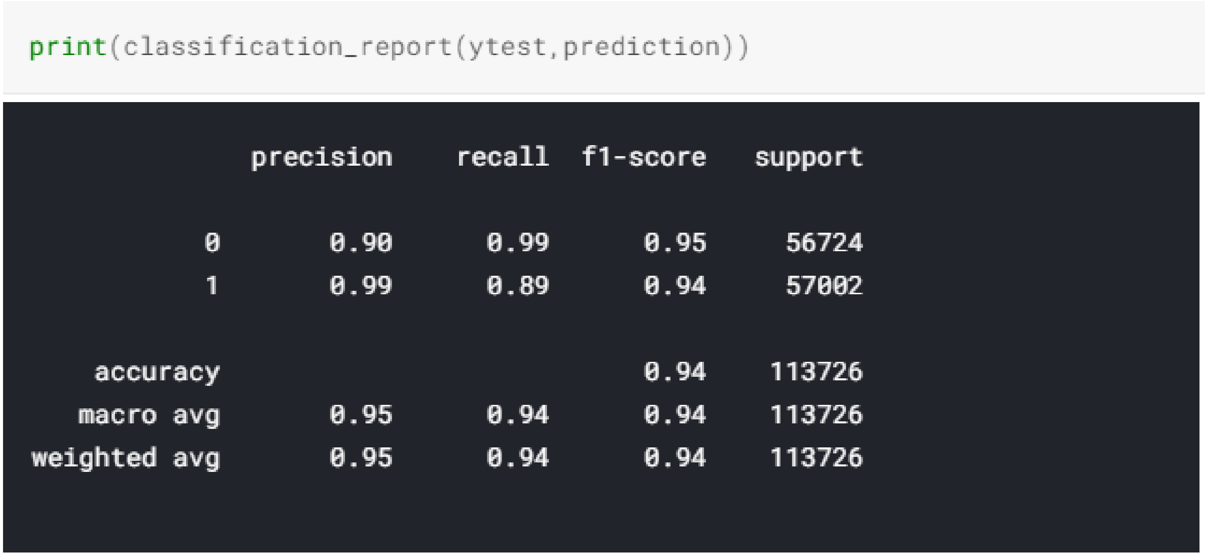


After Resampling the data we have effectively balanced our target classes

```
plt.hist(resampled_class)
plt.grid()
plt.xlabel('classes')
plt.ylabel('frequency')
plt.show()
```

4. **Classification Report**
   This report shows everything about performance of model and it had used some tem like precision and recall, below I have mentioned there definition.

   - **Precision:** precision (also called positive predictive value) is the fraction of relevant instances among the retrieved instances.
   - **Recall:** while recall (also known as sensitivity) is the fraction of the total amount of relevant instances that were actually retrieved.



```
print(classification_report(ytest,prediction))
```

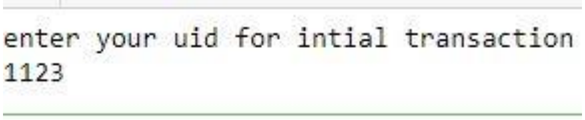
```
              precision    recall  f1-score   support

           0       0.90      0.99      0.95     56724
           1       0.99      0.89      0.94     57002

    accuracy                           0.94    113726
   macro avg       0.95      0.94      0.94    113726
weighted avg       0.95      0.94      0.94    113726
```

This particular report shows that model is 94% accurate in overall classification and it is 90 % sure that a particular data point belongs of class zero and 99% sure that a particular data point belongs to class one.

Facial Check Using Opencv

In this section we are sharing about how we have utilised face recognition technique to check the user is authorised user or not.

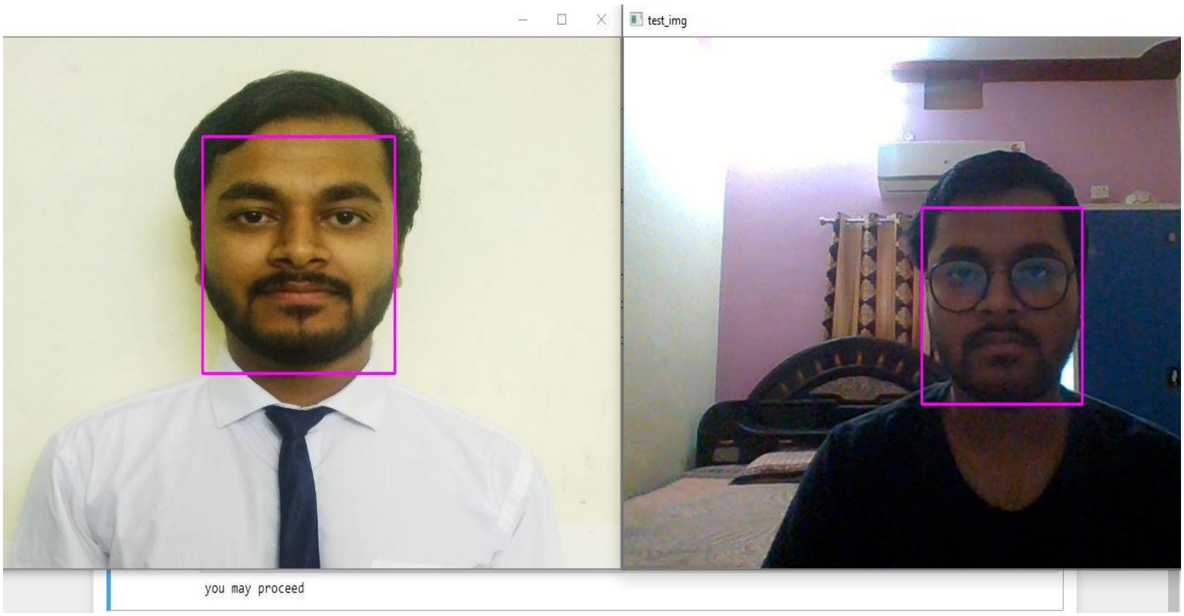
1. First we have asked user to enter the uid so that we can have our first check whether the user is genuine or not.



```
enter your uid for intial transaction
1123
```

   If this id of length four matched with the database saved id then we will proceed otherwise we will some warning.

2. After this we will proceed to next step of face check, so that we can check the user is authorized once or not. Here we will click photo of user from main stream and then we will see if face matched or not.



you may proceed

If face got matched then it will prompt the message that you may proceed and if the face is not matched then it will prompt try again later.

Applications

- This project can be implemented in real time payment systems.
- We can use deep neural network for any fraud transaction classification since **artificial neural networks** are more robust.

So these are some real time use of this project and we will try to make it more efficient since more work is required in this project preferably more in facial recognition part.