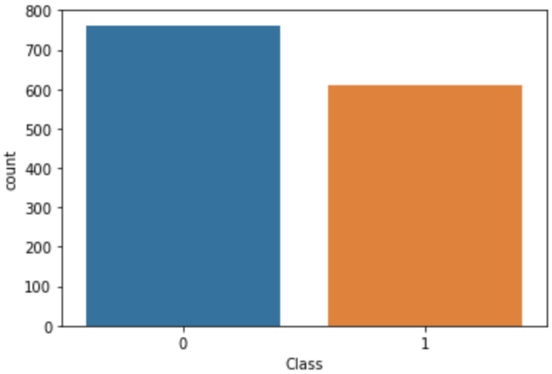
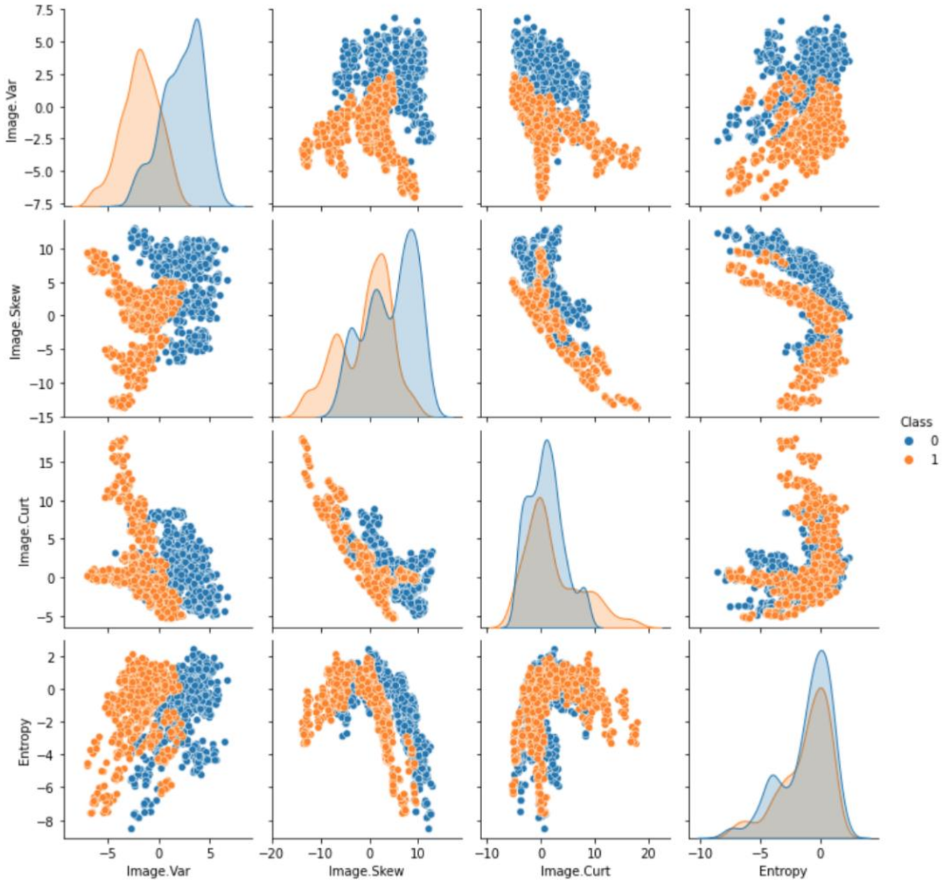
CSID: Hoque, Jupyter Notebook: http://localhost:8888/notebooks/Tasneem_Hoque_Assignment5.ipynb

Initially I imported the pandas framework to read the csv file provided with the bank note data. Then I imported seaborn library to make the counplot and pairplot, as well as set matplotlib inline for viewing. The following are the results:



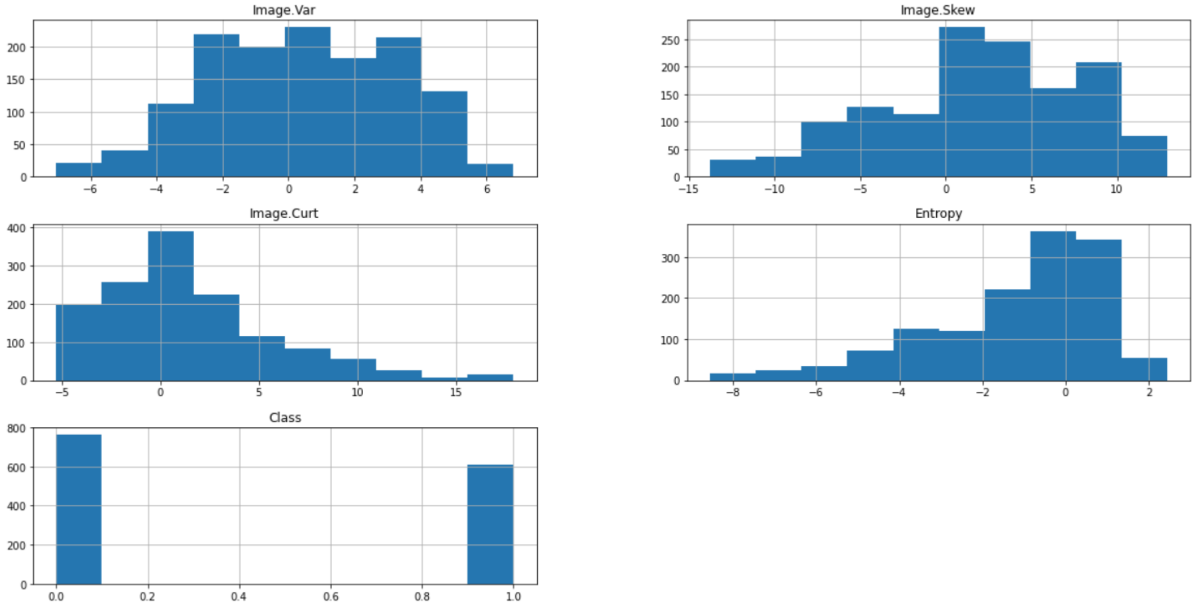
The countplot is showing that we have uneven number of authentic and unauthentic bank notes. There are more unauthentic bank notes in our dataset.



From the pairplot we can see that mostly there is a clear division between the features of authentic and non-authentic bank notes, this will be clearer in a 3D plain of view.

I then printed out the information of the data frame. All features are of type floating point value, the class is binary, so the data type is integer. There is no null count of data in the data frame. From this analysis I realized there is no need for pre-processing as all data are numerical and there are no missing data.

I then proceed to create a histogram of the data and following is the result:



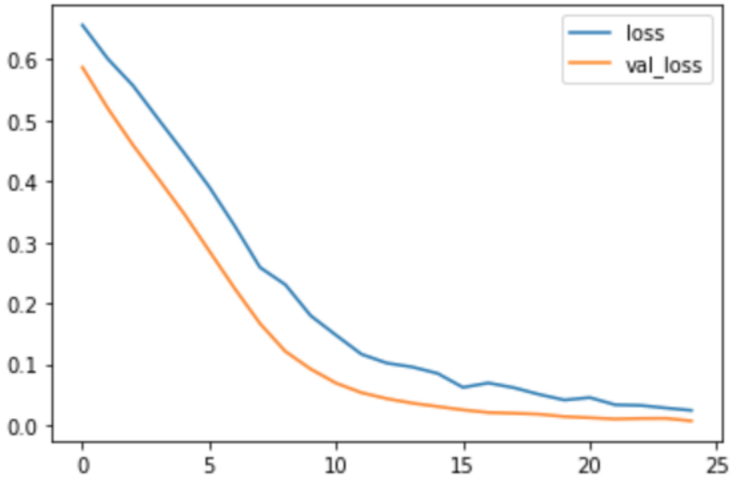
From this we can see the distribution of each individual columns. Image variance is almost equally distributed between the positive and negative numbers. Image skewness is similar but leans more towards positive values. As for the image curtosis, it goes higher up the scale of positive values with more weight on the negative side and image entropy is quite the opposite. The scale of entropy goes further along the negative side with heavier weights on the positive side. Finally, as we have seen before there is more non-authentic bank note data then there is positive.

I then proceed to separate my data into X and y. I imported the sklearn model selection train test split method. Using that, I have divided my data into train and test data, with 20% size of test data at a random state of 101. I also imported the standard scaler to standardize my train values.

Now it is time to build the model. For this problem I have decided to use the sequential model to predict the test data. I have implemented an input layer and three hidden layers with relu activation function and decreasing units. After each layer I have also added dropout layers to prevent overfitting. My final output later is of 1 unit with sigmoid activation function which gives me a value of 1 or 0 as the final result. I have compiled the model with the binary cross entropy loss function and adam optimizer as they work best for binary classification problems.

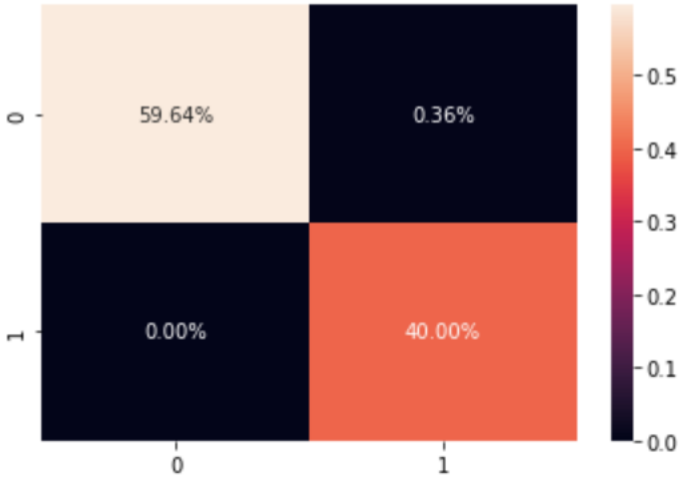
I then fit my model with the x and y train datasets and set the validation dataset as the test sets. I apply 25 epochs as the model shows significant improvements with each epoch. I also use a batch size of 25 for each epoch which is a good amount of data considering the whole dataset.

Once the model is fitted, I plot the loss and validation loss functions.



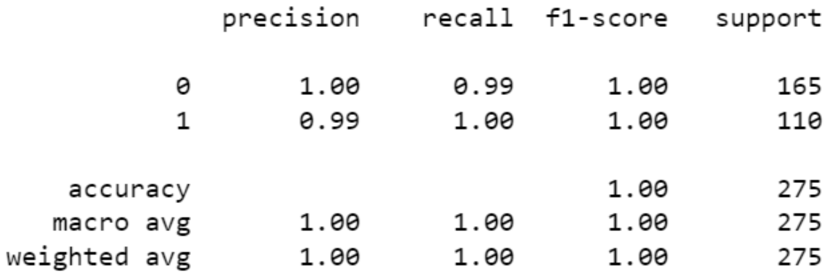
As seen from the curves, there is a steady decrease in loss and there is almost no loss by the end of the last epoch which means we have trained the model successfully. I then use the model to predict the data from the test set and analyze the results using a classification report and a confusion matrix.

Following is the result from the confusion matrix:



As we can see there is mostly true positives and true negatives and there is a very small percentage of 0.36 where the predicted values are false negative. Therefore, we can tell that the model is very accurate when it comes to classifying the test data.

The same can be said from the classification report:



```
              precision    recall  f1-score   support

           0       1.00      0.99      1.00       165
           1       0.99      1.00      1.00       110

    accuracy                           1.00       275
   macro avg       1.00      1.00      1.00       275
weighted avg       1.00      1.00      1.00       275
```

As we can see the precision, recall and f1-score are all giving very good numbers about the accuracy of this model.