BANKNOTE AUTHENTICATION REPORT

Prepared by

Manish Mohan Kamble

Manishkamble7547@gmail.com

July 2021

1.Overview

This data analysis report is to provide Binary Classification Model of ANN To Predict whether the note is genuine or not.

2.About Data

Data were extracted from images that were taken from genuine and forged banknote-like specimens. For digitization, an industrial camera usually used for print inspection was used. The final images have 400x 400 pixels. Due to the object lens and distance to the investigated object gray-scale pictures with a resolution of about 660 dpi were gained. Wavelet Transform tool were used to extract features from images.

Column:

- **variance**

  variance of Wavelet Transformed image (continuous)
- **skewness**

  skewness of Wavelet Transformed image (continuous)
- **curtosis**

  curtosis of Wavelet Transformed image (continuous)

- **entropy**
  entropy of image (continuous)
- **class**
  class (integer)

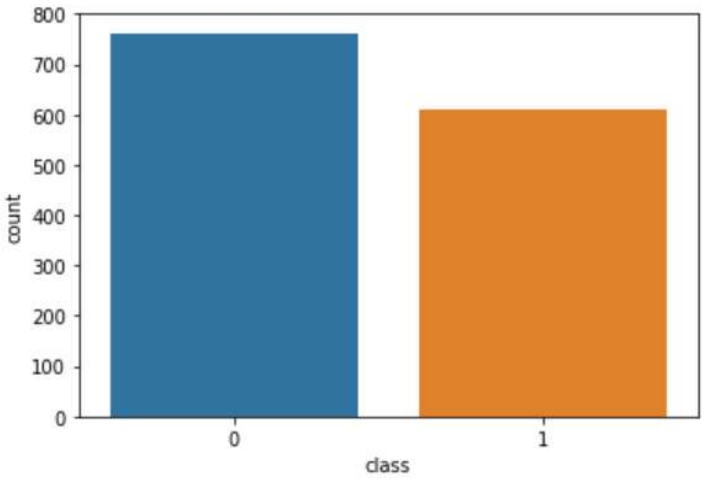
Class 0 is for Authentic notes and Class 1 is for Fake notes

This instances are divided into training, selection, and testing subsets. There are 824 Instances for training(60%), and 274 instances for testing (20%).

We calculate the data distributions and plot a count plot :

Let's have a look at the countplot of the classes available.

```
[9]  plt.figure()
     sns.countplot(data = df, x='class')
     plt.show()
```



Both the classes have nearly equal count. Hence, the dataset has good balance

3.EDA & Preprocessing

First , we check for null values so in the given dataset is very clean and has no null value present Then we get statistical summary suing describe function also next to that we plot a heatmap to find out correlation between features. There is some multicollinearity in features then we visualize continuous variable using distplot and most of the features nearly follow a normal distribution. For better understanding we can also use pairplot There are clear separations shown, especially for

pairs of features having 'variance'. The curtosis-entropy scatterplot exhibits the lowest separation.Then we use iloc for seprating labels and features. After that split it into traing and testing set.Then we apply PCA for decomposition and StandardScaler for scaling of data.

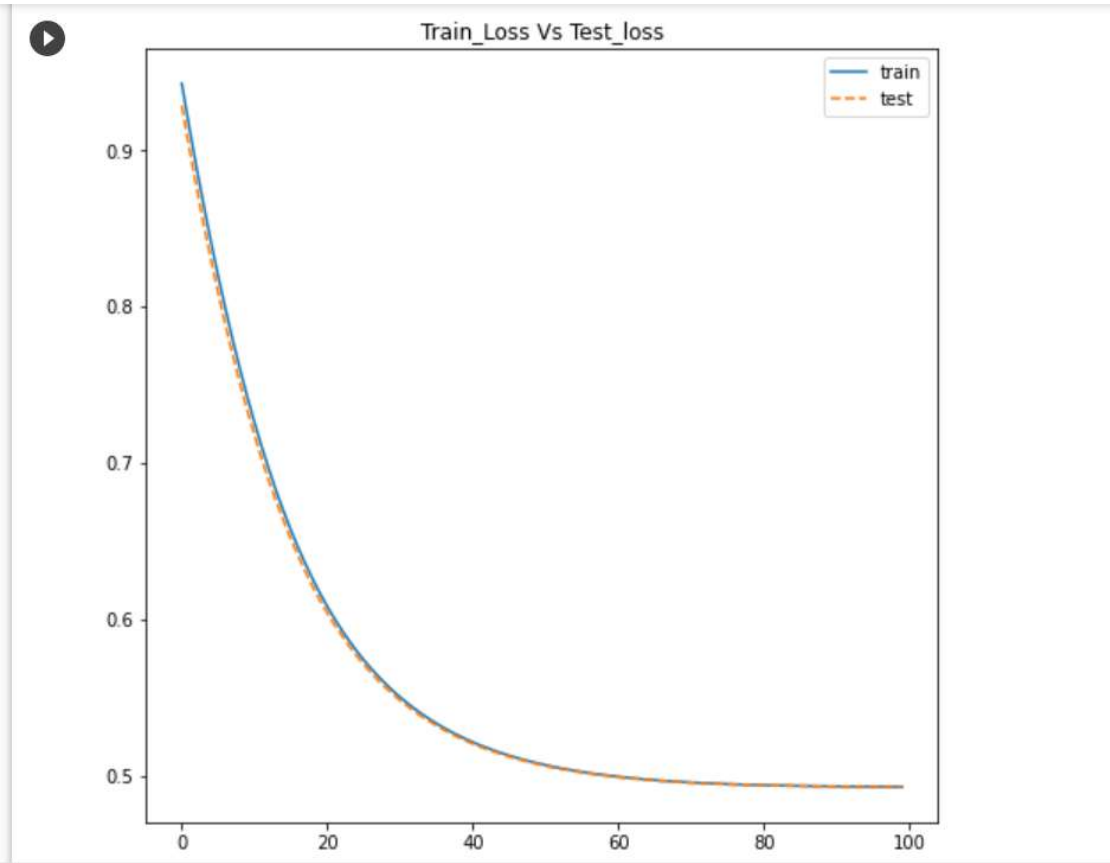
4 .Neural Network

Here we first initialized ANN. Then According to happy monk programming test we need to build a model with 1 hidden layer and it's a binary classification so we use 1 neuron and sigmoid as activation function .In summary output shape is (None,3) means total parameters is 3.Then we compile a model using adam optimizer and loss function is binary cross entropy with accuracy metrics. To fit our model we use 100 epochs of batch size 20 and we also pass data for validation.

After execution we plot Train loss Vs Test loss:

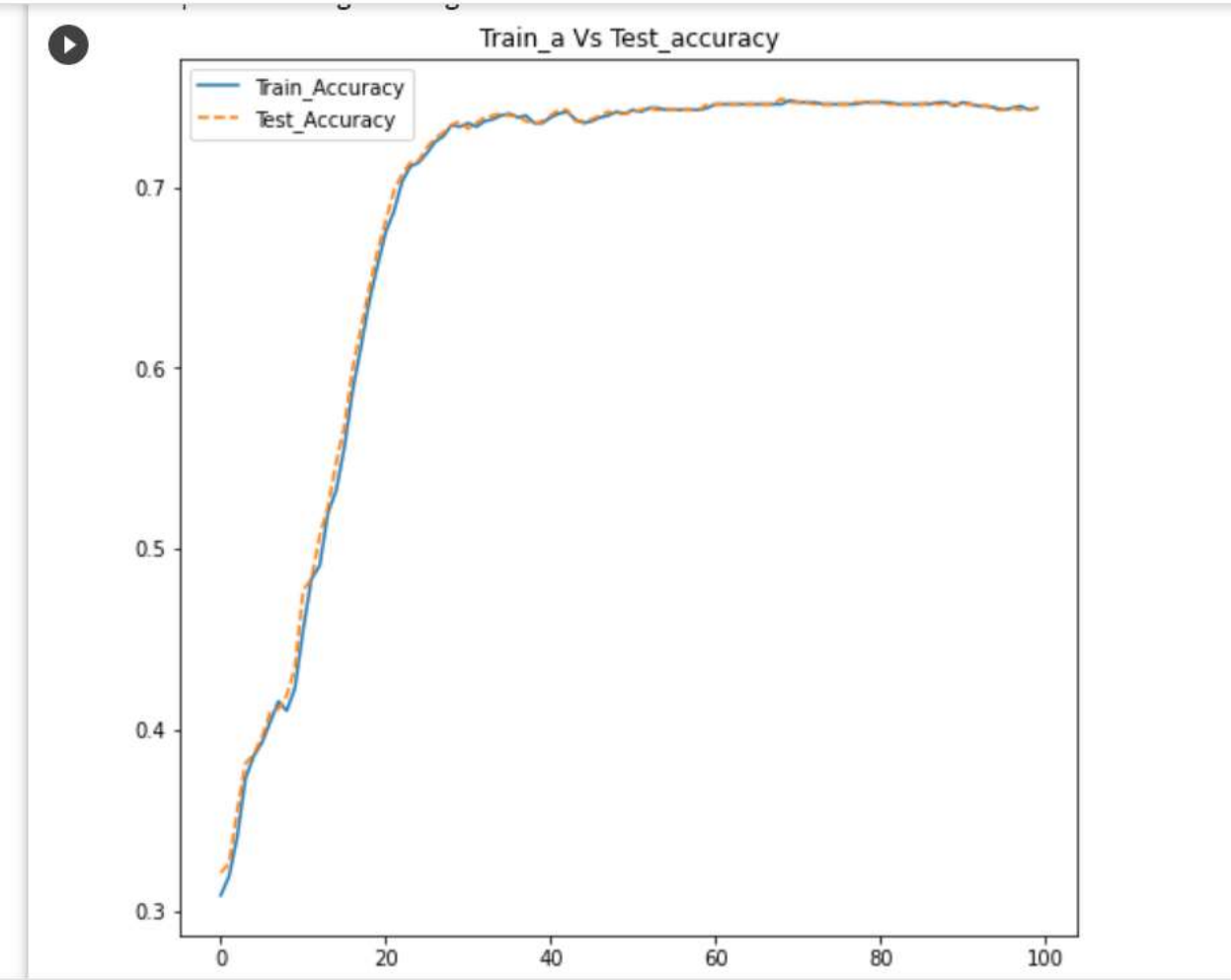
Training vs Test loss

```
fig , ax = plt.subplots(figsize=(8,8))
plt.title('Train_Loss Vs Test_loss')
plt.plot(model_history.history['loss'], label='train')
plt.plot(model_history.history['val_loss'],label='test',linestyle='--')
plt.legend()
plt.show()
```



Here we also plot Train Accuracy Vs Test Accuracy:

```
fig , ax = plt.subplots(figsize=(8,8))
plt.title('Train_a Vs Test_accuracy')
plt.plot(model_history.history['accuracy'], label='Train_Accuracy')
plt.plot(model_history.history['val_accuracy'],label='Test_Accuracy',linestyle='--')
plt.legend()
```



After plotting this graph we calculate y_pred from X_test and then we create a classification report

And **model achieved an estimated classification accuracy of about 75%.**

And we also plot classification report for better understanding.

5.Conclusion

we build a ANN model with 100 epochs and visualize train vs test loss, train and test accuracy,classification report and confusion metrics .So we can see that the model achieved an estimated classification accuracy of about 75%.