Keras Basics

```
# Variance of wavelet Transformed image ( continuous )
# Skewness of wavelet Transformed image ( continuous )
# Curtosis of wavelet Transformed image ( continuous )
# Entropy of image ( continuous )
# Class ( integer )
```

```
from numpy import genfromtxt
df=genfromtxt('/content/bank_note_data.txt',delimiter=',')
df
```

```
array([[  3.6216 ,   8.6661 ,  -2.8073 ,  -0.44699,   0.     ],
       [  4.5459 ,   8.1674 ,  -2.4586 ,  -1.4621 ,   0.     ],
       [  3.866  ,  -2.6383 ,   1.9242 ,   0.10645,   0.     ],
       ...,
       [ -3.7503 , -13.4586 ,  17.5932 ,  -2.7771 ,   1.     ],
       [ -3.5637 ,  -8.3827 ,  12.393  ,  -1.2823 ,   1.     ],
       [ -2.5419 ,  -0.65804,   2.6842 ,   1.1952 ,   1.     ]])
```

```
y=df[:,4]
y
```

```
array([0., 0., 0., ..., 1., 1., 1.])
```

```
x=df[:,0:4]
x
```

```
array([[  3.6216 ,   8.6661 ,  -2.8073 ,  -0.44699],
       [  4.5459 ,   8.1674 ,  -2.4586 ,  -1.4621 ],
       [  3.866  ,  -2.6383 ,   1.9242 ,   0.10645],
       ...,
       [ -3.7503 , -13.4586 ,  17.5932 ,  -2.7771 ],
       [ -3.5637 ,  -8.3827 ,  12.393  ,  -1.2823 ],
       [ -2.5419 ,  -0.65804,   2.6842 ,   1.1952 ]])
```

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.33,random_state=42)
```

```
x_train
```

```
array([[-0.8734  , -0.033118, -0.20165 ,  0.55774 ],
       [ 2.0177  ,  1.7982  , -2.9581  ,  0.2099  ],
       [-0.36038 ,  4.1158  ,  3.1143  , -0.37199 ],
       ...,
       [-7.0364  ,  9.2931  ,  0.16594 , -4.5396  ],
       [-3.4605  ,  2.6901  ,  0.16165 , -1.0224  ],
       [-3.3582  , -7.2404  , 11.4419  , -0.57113 ]])
```

```
x_test
```

```
array([[ 1.5691  ,  6.3465  , -0.1828  , -2.4099  ],
       [-0.27802 ,  8.1881  , -3.1338  , -2.5276  ],
       [ 0.051979,  7.0521  , -2.0541  , -3.1508  ],
       ...,
       [ 3.5127  ,  2.9073  ,  1.0579  ,  0.40774 ],
       [ 5.504   , 10.3671  , -4.413   , -4.0211  ],
       [-0.2062  ,  9.2207  , -3.7044  , -6.8103  ]])
```

```
y_train
```

```
array([1., 1., 0., 1., 0., 0., 1., 0., 0., 1., 0., 0., 0., 1., 1., 1., 0.,
       1., 1., 1., 0., 1., 1., 1., 0., 1., 0., 0., 1., 0., 0., 0., 1., 0.,
       1., 0., 0., 0., 0., 1., 1., 0., 0., 1., 0., 0., 1., 1., 1., 0., 0.,
       0., 1., 1., 1., 1., 1., 1., 0., 0., 0., 0., 0., 1., 0., 1.,
       0., 0., 0., 0., 1., 1., 0., 1., 1., 0., 1., 0., 1., 1., 1., 1., 1.,
       1., 1., 1., 1., 1., 1., 0., 1., 0., 0., 0., 0., 1., 1., 1., 0., 0.,
       0., 1., 0., 1., 1., 1., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 1., 1., 0., 1., 0., 1., 0., 0., 0., 0., 0., 1., 1., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1., 1., 0., 1., 0., 0., 1.,
       1., 0., 1., 0., 0., 1., 0., 1., 1., 0., 0., 0., 0., 0., 1., 0.,
       0., 0., 1., 1., 0., 1., 1., 1., 1., 1., 0., 1., 1., 0., 0., 1., 1.,
       0., 1., 1., 0., 1., 1., 0., 0., 0., 0., 1., 0., 0., 1., 0., 1., 0.,
       0., 1., 0., 0., 1., 1., 0., 0., 1., 1., 0., 1., 1., 1., 0., 0., 0.,
       1., 0., 0., 1., 0., 0., 0., 1., 0., 0., 1., 0., 1., 0., 0., 0., 0.,
       0., 1., 1., 0., 0., 1., 0., 0., 1., 0., 0., 1., 0., 1., 1., 1., 0.,
       1., 0., 0., 0., 1., 0., 0., 1., 1., 1., 0., 1., 0., 1., 1., 0., 0.,
       1., 1., 1., 1., 0., 0., 1., 0., 1., 0., 1., 0., 0., 0., 1., 0., 1.,
```

```
       1., 0., 0., 1., 0., 1., 0., 1., 0., 1., 0., 0., 0., 0., 1., 0., 1.,
       1., 1., 1., 1., 1., 0., 0., 0., 1., 0., 0., 0., 1., 0., 1., 0., 0.,
       0., 0., 0., 1., 0., 1., 1., 0., 1., 0., 0., 1., 0., 0., 0., 0., 0.,
       0., 1., 1., 1., 0., 1., 0., 0., 0., 1., 1., 0., 0., 1., 0., 1., 0.,
       0., 0., 0., 0., 0., 1., 1., 1., 0., 0., 0., 0., 0., 0., 1., 1., 1.,
       0., 0., 0., 1., 1., 1., 1., 0., 0., 1., 0., 0., 0., 1., 0., 0., 1.,
       0., 0., 0., 1., 0., 0., 1., 1., 0., 1., 0., 0., 1., 0., 1., 1., 1.,
       0., 0., 1., 0., 0., 1., 0., 1., 1., 0., 1., 0., 0., 0., 0., 1., 0.,
       0., 0., 1., 0., 0., 0., 0., 0., 1., 1., 0., 1., 0., 0., 0., 0., 0.,
       0., 1., 0., 1., 1., 1., 0., 0., 1., 1., 1., 1., 0., 0., 1., 0., 0.,
       1., 0., 0., 0., 1., 0., 1., 0., 0., 0., 1., 0., 0., 1., 1., 1., 1.,
       0., 1., 1., 0., 1., 0., 1., 0., 0., 0., 1., 0., 0., 1., 0., 0., 1.,
       0., 1., 0., 1., 0., 0., 0., 0., 0., 0., 1., 1., 0., 1., 0., 0., 1.,
       0., 1., 0., 1., 1., 1., 0., 1., 0., 1., 1., 0., 0., 0., 0., 1., 0.,
       0., 1., 1., 1., 1., 0., 0., 0., 0., 0., 1., 1., 0., 0., 1., 0., 0.,
       0., 1., 1., 0., 0., 0., 1., 1., 1., 1., 0., 1., 1., 0., 0., 0., 1.,
       1., 0., 0., 1., 1., 0., 0., 1., 0., 0., 0., 0., 0., 0., 1., 1., 1.,
       0., 1., 1., 1., 0., 1., 1., 0., 0., 0., 0., 1., 1., 1., 0., 1., 0.,
       0., 0., 1., 0., 0., 1., 1., 1., 1., 1., 0., 1., 0., 0., 0., 1., 0.,
       0., 0., 1., 0., 1., 1., 0., 1., 0., 1., 1., 1., 1., 0., 1., 1., 1.,
       1., 1., 0., 1., 0., 1., 0., 0., 0., 1., 0., 0., 1., 0., 1., 0., 0.,
       1., 0., 0., 1., 1., 1., 0., 0., 1., 1., 1., 1., 1., 1., 0., 0., 0.,
       0., 0., 1., 1., 1., 1., 0., 1., 0., 1., 0., 0., 0., 1., 0., 0., 1.,
       1., 1., 0., 1., 0., 0., 0., 1., 1., 0., 0., 1., 1., 0., 1., 0., 0.,
       1., 0., 0., 0., 1., 0., 0., 1., 1., 1., 1., 0., 0., 1., 0., 1., 0.,
       0., 0., 0., 0., 0., 1., 1., 0., 1., 0., 0., 0., 1., 1., 0., 0., 0.,
       1., 0., 1., 1., 1., 0., 0., 0., 1., 1., 0., 1., 1., 0., 1., 0., 0.,
       0., 0., 0., 1., 0., 1., 1., 1., 0., 0., 1., 0., 1., 0., 1., 0., 1.,
       0., 0., 0., 0., 0., 1., 0., 1., 0., 0., 1., 1., 1., 0., 1., 0., 1.,
       0., 1., 0., 0., 1., 0., 1., 0., 0., 0., 1., 0., 1., 0., 1., 1., 0.,
       1., 0., 1., 1., 0., 1., 1., 0., 0., 1., 0., 0., 1., 1., 1., 1., 1.,
       1.])
```

```
y_test
```

```
array([0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1., 1., 0., 1., 0., 1., 1.,
       1., 1., 1., 0., 0., 1., 0., 1., 0., 0., 1., 0., 0., 1., 0., 0., 1.,
       1., 0., 1., 1., 1., 0., 0., 1., 1., 0., 1., 1., 1., 0., 0., 1., 0.,
       0., 0., 0., 0., 1., 0., 0., 0., 0., 1., 0., 1., 0., 0., 0., 0., 0.,
       0., 1., 1., 0., 1., 0., 1., 0., 0., 1., 1., 1., 1., 0., 1., 0., 0.,
       0., 0., 1., 1., 0., 0., 0., 1., 1., 0., 1., 1., 0., 0., 0., 1., 0.,
       0., 0., 1., 0., 0., 1., 1., 1., 1., 1., 0., 1., 1., 1., 0., 1., 1.,
       0., 1., 0., 1., 0., 1., 0., 1., 1., 0., 1., 1., 0., 0., 0., 0., 0.,
       1., 0., 0., 0., 0., 0., 1., 0., 1., 1., 1., 1., 1., 0., 1., 1., 1.,
       0., 1., 0., 1., 0., 0., 0., 1., 1., 1., 1., 1., 0., 1., 0., 0., 0.,
       0., 0., 0., 1., 0., 0., 1., 1., 0., 0., 0., 0., 1., 0., 1., 0., 1.,
       1., 0., 0., 1., 0., 0., 1., 1., 1., 1., 0., 0., 1., 1., 1., 0., 0.,
       1., 1., 1., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1., 1., 1., 1.,
       1., 1., 0., 1., 0., 0., 1., 1., 1., 1., 0., 1., 0., 1., 1., 1., 1.,
       0., 0., 0., 1., 0., 1., 1., 1., 0., 0., 0., 0., 0., 1., 0., 1., 1.,
       0., 0., 0., 1., 1., 0., 0., 0., 1., 1., 0., 1., 0., 1., 1., 1., 1.,
       0., 0., 0., 0., 0., 1., 0., 0., 1., 0., 1., 0., 0., 1., 0., 0., 0.,
       0., 0., 1., 1., 0., 1., 0., 1., 1., 0., 0., 0., 0., 0., 0., 0., 1.,
       1., 1., 0., 0., 1., 1., 0., 0., 0., 0., 0., 0., 1., 0., 1., 1., 1.,
       0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 1., 0., 1., 1., 1.,
       1., 0., 1., 1., 0., 1., 1., 0., 0., 1., 0., 0., 1., 0., 1., 0., 1.,
       1., 0., 0., 1., 0., 0., 0., 0., 0., 1., 0., 0., 1., 0., 0., 0., 0.,
       0., 1., 1., 0., 1., 0., 1., 1., 1., 0., 1., 1., 0., 1., 1., 0., 1.,
       0., 0., 0., 1., 1., 1., 0., 0., 0., 0., 1., 0., 0., 1., 1., 1., 1.,
       0., 1., 0., 0., 1., 1., 0., 1., 0., 1., 0., 0., 0., 1., 1., 0., 0.,
       0., 1., 0., 0., 1., 0., 0., 0., 0., 1., 0., 1., 0., 0., 1., 0., 0.,
       0., 0., 0., 0., 1., 1., 0., 0., 0., 0., 0.])
```

```
from sklearn.preprocessing import MinMaxScaler
scaler=MinMaxScaler()
scaled_train=scaler.fit_transform(x_train)
scaled_test=scaler.fit_transform(x_test)
```

```
scaled_test
```

```
array([[0.65576832, 0.76019778, 0.22784929, 0.52919178],
       [0.51021119, 0.83150383, 0.09609469, 0.51778828],
       [0.53621584, 0.78751834, 0.14430053, 0.45740888],
       ...,
       [0.80892829, 0.62703326, 0.28324337, 0.80218187],
       [0.96584712, 0.91587388, 0.03898168, 0.37308892],
       [0.51587076, 0.87148571, 0.07061886, 0.10285426]])
```

Building the network with Keras

```
from keras.models import Sequential
from keras.layers import Dense


model=Sequential()
model.add(Dense(4,input_dim=4,activation='relu'))
model.add(Dense(8,activation='relu'))
model.add(Dense(1,activation='sigmoid'))
# sigmoid function to output 0 or 1


model.compile(loss='binary_crossentropy',optimizer='adam',metrics=['accuracy'])


model.fit(scaled_train,y_train,epochs=100,verbose=1)
```

```
Epoch 1/100
29/29 [==============================] - 1s 3ms/step - loss: 0.7047 - accuracy: 0.4570
Epoch 2/100
29/29 [==============================] - 0s 2ms/step - loss: 0.6972 - accuracy: 0.4538
Epoch 3/100
29/29 [==============================] - 0s 3ms/step - loss: 0.6899 - accuracy: 0.4864
Epoch 4/100
29/29 [==============================] - 0s 2ms/step - loss: 0.6828 - accuracy: 0.5484
Epoch 5/100
29/29 [==============================] - 0s 2ms/step - loss: 0.6714 - accuracy: 0.5495
Epoch 6/100
29/29 [==============================] - 0s 2ms/step - loss: 0.6549 - accuracy: 0.5495
Epoch 7/100
29/29 [==============================] - 0s 2ms/step - loss: 0.6438 - accuracy: 0.5789
Epoch 8/100
29/29 [==============================] - 0s 2ms/step - loss: 0.6344 - accuracy: 0.6072
Epoch 9/100
29/29 [==============================] - 0s 2ms/step - loss: 0.6251 - accuracy: 0.6627
Epoch 10/100
29/29 [==============================] - 0s 3ms/step - loss: 0.6154 - accuracy: 0.6910
Epoch 11/100
29/29 [==============================] - 0s 2ms/step - loss: 0.6054 - accuracy: 0.7247
Epoch 12/100
29/29 [==============================] - 0s 2ms/step - loss: 0.5953 - accuracy: 0.7312
Epoch 13/100
29/29 [==============================] - 0s 2ms/step - loss: 0.5851 - accuracy: 0.7650
Epoch 14/100
29/29 [==============================] - 0s 3ms/step - loss: 0.5729 - accuracy: 0.7639
Epoch 15/100
29/29 [==============================] - 0s 2ms/step - loss: 0.5609 - accuracy: 0.7845
Epoch 16/100
29/29 [==============================] - 0s 2ms/step - loss: 0.5486 - accuracy: 0.7943
Epoch 17/100
29/29 [==============================] - 0s 2ms/step - loss: 0.5362 - accuracy: 0.7965
Epoch 18/100
29/29 [==============================] - 0s 3ms/step - loss: 0.5231 - accuracy: 0.8041
Epoch 19/100
29/29 [==============================] - 0s 3ms/step - loss: 0.5095 - accuracy: 0.8052
Epoch 20/100
29/29 [==============================] - 0s 3ms/step - loss: 0.4963 - accuracy: 0.8139
Epoch 21/100
29/29 [==============================] - 0s 3ms/step - loss: 0.4819 - accuracy: 0.8172
Epoch 22/100
29/29 [==============================] - 0s 2ms/step - loss: 0.4680 - accuracy: 0.8237
Epoch 23/100
29/29 [==============================] - 0s 4ms/step - loss: 0.4526 - accuracy: 0.8259
Epoch 24/100
29/29 [==============================] - 0s 3ms/step - loss: 0.4385 - accuracy: 0.8422
Epoch 25/100
29/29 [==============================] - 0s 3ms/step - loss: 0.4239 - accuracy: 0.8487
Epoch 26/100
29/29 [==============================] - 0s 3ms/step - loss: 0.4098 - accuracy: 0.8575
Epoch 27/100
29/29 [==============================] - 0s 3ms/step - loss: 0.3948 - accuracy: 0.8672
Epoch 28/100
29/29 [==============================] - 0s 4ms/step - loss: 0.3812 - accuracy: 0.8803
Epoch 29/100
29/29 [==============================] - 0s 4ms/step - loss: 0.3673 - accuracy: 0.8825
```

```
y_pred=model.predict(x_test)
y_pred
```

```
15/15 [==============================] - 0s 2ms/step
array([[9.20854514e-28],
       [6.93063454e-11],
       [8.19733668e-15],
       [0.00000000e+00],
```

```
       [3.75721536e-28],
       [2.85020317e-22],
       [3.00352596e-26],
       [2.52650863e-29],
       [2.81372959e-24],
       [4.02340680e-28],
       [1.00000000e+00],
       [1.00000000e+00],
       [1.30226319e-21],
       [9.99287248e-01],
       [1.40388018e-17],
       [1.00000000e+00],
       [9.98186886e-01],
       [9.98186886e-01],
       [9.99721706e-01],
       [9.98186886e-01],
       [4.44235067e-31],
       [1.75702188e-25],
       [9.98186886e-01],
       [2.64845137e-32],
       [9.98186886e-01],
       [1.69667070e-29],
       [7.18662487e-37],
       [9.99108672e-01],
       [0.00000000e+00],
       [0.00000000e+00],
       [1.00000000e+00],
       [9.66346010e-24],
       [2.75836817e-18],
       [9.98186886e-01],
       [9.98186886e-01],
       [5.08299943e-35],
       [1.00000000e+00],
       [1.00000000e+00],
       [1.00000000e+00],
       [5.92959009e-26],
       [8.35832565e-27],
       [1.00000000e+00],
       [1.00000000e+00],
       [4.44491278e-30],
       [9.98186886e-01],
       [9.99458730e-01],
       [9.98186886e-01],
       [8.12956167e-18],
       [3.53270683e-18],
       [1.00000000e+00],
       [5.57486315e-18],
       [4.58032502e-31],
       [6.48378941e-29],
       [2.75666532e-32],
       [2.60135519e-21],
       [1.00000000e+00],
```

```
import numpy as np
y_pred=np.round(y_pred).astype(int)
y_pred
```

```
array([[0],
       [0],
       [0],
       [0],
       [0],
       [0],
       [0],
       [0],
       [0],
       [1],
       [1],
       [0],
       [1],
       [0],
       [1],
       [1],
       [1],
       [1],
       [1],
       [0],
       [0],
       [1],
       [0],
       [1],
       [0],
       [0],
       [1],
       [0],
       [0],
       [0],
       [1],
       [0],
```

```
       [0],
       [1],
       [1],
       [0],
       [1],
       [1],
       [1],
       [0],
       [0],
       [1],
       [1],
       [0],
       [1],
       [1],
       [1],
       [0],
       [0],
       [1],
       [0],
       [0],
       [0],
       [0],
       [0],
       [1],
       [0],
       [0]
```

```
# model.fit(scaled_train,y_train,epochs=100,verbose=0)
```

```
# model.fit(scaled_train,y_train,epochs=100,verbose=2)
```

```
# model.fit(scaled_train,y_train,epochs=100,verbose=3)
```

```
model.evaluate(x=scaled_test,y=y_test)
```

```
15/15 [==============================] - 0s 2ms/step - loss: 0.0831 - accuracy: 0.9757
[0.08308165520429611, 0.9757174253463745]
```

```
from sklearn.metrics import confusion_matrix,classification_report
```

```
confusion_matrix(y_test,y_pred)
```

```
array([[254,   3],
       [  1, 195]])
```

```
print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

         0.0       1.00      0.99      0.99       257
         1.0       0.98      0.99      0.99       196

    accuracy                           0.99       453
   macro avg       0.99      0.99      0.99       453
weighted avg       0.99      0.99      0.99       453
```

```
model.save('mymodel.h5')
from keras.models import load_model
newmodel=load_model('mymodel.h5')
```

```
/usr/local/lib/python3.10/dist-packages/keras/src/engine/training.py:3000: UserWarning: You are saving your model as an HDF5 file v
  saving_api.save_model(
```

```
newmodel.predict([[0.65576832, 0.76019778, 0.22784929, 0.52919178],
  [0.51021119, 0.83150383, 0.09609469, 0.51778828]])
```

```
1/1 [==============================] - 0s 83ms/step
array([[0.00290658],
       [0.28803214]], dtype=float32)
```