```
!pip install tensorflow
```

```
Requirement already satisfied: tensorflow in
/usr/local/lib/python3.10/dist-packages (2.15.0)
Requirement already satisfied: absl-py>=1.0.0 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (1.4.0)
Requirement already satisfied: astunparse>=1.6.0 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (1.6.3)
Requirement already satisfied: flatbuffers>=23.5.26 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (23.5.26)
Requirement already satisfied: gast!=0.5.0,!=0.5.1,!=0.5.2,>=0.2.1
in /usr/local/lib/python3.10/dist-packages (from tensorflow) (0.5.4)
Requirement already satisfied: google-pasta>=0.1.1 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (0.2.0)
Requirement already satisfied: h5py>=2.9.0 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (3.9.0)
Requirement already satisfied: libclang>=13.0.0 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (16.0.6)
Requirement already satisfied: ml-dtypes~=0.2.0 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (0.2.0)
Requirement already satisfied: numpy<2.0.0,>=1.23.5 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (1.23.5)
Requirement already satisfied: opt-einsum>=2.3.2 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (3.3.0)
Requirement already satisfied: packaging in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (23.2)
Requirement already satisfied: protobuf!=4.21.0,!=4.21.1,!=4.21.2,!
=4.21.3,!=4.21.4,!=4.21.5,<5.0.0dev,>=3.20.3 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (3.20.3)
Requirement already satisfied: setuptools in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (69.0.3)
Requirement already satisfied: six>=1.12.0 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (1.16.0)
Requirement already satisfied: termcolor>=1.1.0 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (2.4.0)
Requirement already satisfied: typing-extensions>=3.6.6 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (4.5.0)
Requirement already satisfied: wrapt<1.15,>=1.11.0 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (1.14.1)
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (0.35.0)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (1.60.0)
Requirement already satisfied: tensorboard<2.16,>=2.15 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (2.15.1)
Requirement already satisfied: tensorflow-estimator<2.16,>=2.15.0
in /usr/local/lib/python3.10/dist-packages (from tensorflow) (2.15.0)
Requirement already satisfied: keras<2.16,>=2.15.0 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (2.15.0)
Requirement already satisfied: wheel<1.0,>=0.23.0 in
```

```
/usr/local/lib/python3.10/dist-packages (from astunparse>=1.6.0-
>tensorflow) (0.42.0)
Requirement already satisfied: google-auth<3,>=1.6.3 in
/usr/local/lib/python3.10/dist-packages (from tensorboard<2.16,>=2.15-
>tensorflow) (2.17.3)
Requirement already satisfied: google-auth-oauthlib<2,>=0.5 in
/usr/local/lib/python3.10/dist-packages (from tensorboard<2.16,>=2.15-
>tensorflow) (1.2.0)
Requirement already satisfied: markdown>=2.6.8 in
/usr/local/lib/python3.10/dist-packages (from tensorboard<2.16,>=2.15-
>tensorflow) (3.5.2)
Requirement already satisfied: requests<3,>=2.21.0 in
/usr/local/lib/python3.10/dist-packages (from tensorboard<2.16,>=2.15-
>tensorflow) (2.31.0)
Requirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.0
in /usr/local/lib/python3.10/dist-packages (from
tensorboard<2.16,>=2.15->tensorflow) (0.7.2)
Requirement already satisfied: werkzeug>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from tensorboard<2.16,>=2.15-
>tensorflow) (3.0.1)
Requirement already satisfied: cachetools<6.0,>=2.0.0 in
/usr/local/lib/python3.10/dist-packages (from google-auth<3,>=1.6.3-
>tensorboard<2.16,>=2.15->tensorflow) (5.3.2)
Requirement already satisfied: pyasn1-modules>=0.2.1 in
/usr/local/lib/python3.10/dist-packages (from google-auth<3,>=1.6.3-
>tensorboard<2.16,>=2.15->tensorflow) (0.3.0)
Requirement already satisfied: rsa<5,>=3.1.4 in
/usr/local/lib/python3.10/dist-packages (from google-auth<3,>=1.6.3-
>tensorboard<2.16,>=2.15->tensorflow) (4.9)
Requirement already satisfied: requests-oauthlib>=0.7.0 in
/usr/local/lib/python3.10/dist-packages (from google-auth-
oauthlib<2,>=0.5->tensorboard<2.16,>=2.15->tensorflow) (1.3.1)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.10/dist-packages (from requests<3,>=2.21.0-
>tensorboard<2.16,>=2.15->tensorflow) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in
/usr/local/lib/python3.10/dist-packages (from requests<3,>=2.21.0-
>tensorboard<2.16,>=2.15->tensorflow) (3.6)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.10/dist-packages (from requests<3,>=2.21.0-
>tensorboard<2.16,>=2.15->tensorflow) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.10/dist-packages (from requests<3,>=2.21.0-
>tensorboard<2.16,>=2.15->tensorflow) (2023.11.17)
Requirement already satisfied: MarkupSafe>=2.1.1 in
/usr/local/lib/python3.10/dist-packages (from werkzeug>=1.0.1-
>tensorboard<2.16,>=2.15->tensorflow) (2.1.4)
Requirement already satisfied: pyasn1<0.6.0,>=0.4.6 in
/usr/local/lib/python3.10/dist-packages (from pyasn1-modules>=0.2.1-
>google-auth<3,>=1.6.3->tensorboard<2.16,>=2.15->tensorflow) (0.5.1)
Requirement already satisfied: oauthlib>=3.0.0 in
/usr/local/lib/python3.10/dist-packages (from requests-
oauthlib>=0.7.0->google-auth-oauthlib<2,>=0.5-
>tensorboard<2.16,>=2.15->tensorflow) (3.2.2)
WARNING: Running pip as the 'root' user can result in broken
permissions and conflicting behaviour with the system package manager.
It is recommended to use a virtual environment instead:
https://pip.pypa.io/warnings/venv
```

```
!nvidia-smi
```

```
Thu Feb  1 08:01:13 2024
+----------------------------------------------------------------------
-----------------+
| NVIDIA-SMI 535.104.05             Driver Version: 535.104.05   CUDA
Version: 12.2     |
|-----------------------------------------+----------------------
+----------------------+
| GPU  Name                 Persistence-M | Bus-Id        Disp.A |
Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |         Memory-Usage |
GPU-Util  Compute M. |
|                                         |                      |
MIG M. |
|
=========================================+======================+======
================|
|   0  Tesla T4                       Off | 00000000:00:04.0 Off |
0 |
| N/A   39C    P8               9W /  70W |      0MiB / 15360MiB |
0%      Default |
|                                         |                      |
N/A |
+-----------------------------------------+----------------------
+----------------------+


+----------------------------------------------------------------------
-----------------+
| Processes:
|
|  GPU   GI   CI        PID   Type   Process name
GPU Memory |
|        ID   ID
Usage      |
|
=======================================================================
================|
|  No running processes found
```

```
|
+---------------------------------------------------------------------
------------------+
```

```
!pip install tensorflow-gpu
```

```
Collecting tensorflow-gpu
  Using cached tensorflow-gpu-2.12.0.tar.gz (2.6 kB)
  error: subprocess-exited-with-error

  × python setup.py egg_info did not run successfully.
  │ exit code: 1
  ╰─> See above for output.

  note: This error originates from a subprocess, and is likely not a
problem with pip.
  Preparing metadata (setup.py) ... error: metadata-generation-failed

× Encountered error while generating package metadata.
╰─> See above for output.

note: This is an issue with the package mentioned above, not pip.
hint: See above for details.
```

```python
import tensorflow as tf
print("TensorFlow version:", tf.__version__)
print("GPU available:", tf.config.list_physical_devices('GPU'))
```

```
TensorFlow version: 2.15.0
GPU available: [PhysicalDevice(name='/physical_device:GPU:0',
device_type='GPU')]
```

```python
import tensorflow as tf

# Create a random matrix
a = tf.random.normal((1000, 1000))

# Perform a matrix multiplication on GPU
b = tf.matmul(a, a)

print("GPU computation time:", b)
```

```
GPU computation time: tf.Tensor(
[[ 23.644073  -53.09984     4.4855604 ...  44.762833   15.747591
  -27.254604 ]
 [ 17.465328  -10.247104  -48.122856  ...    9.175959   20.489845
   40.124718 ]
 [ -5.5265207  32.026146   31.498093  ...  40.996006   36.62998
   69.20412  ]
 ...
 [  3.4558582  45.48342     5.707222  ... -24.857723   46.83235
```

```
  -18.666574 ]
 [ -1.7733288 -16.848602    9.835978  ...  57.113823    7.558341
  -18.197208 ]
 [  6.84245     9.575052   66.461      ...   9.252407   49.378075
   30.718555 ]], shape=(1000, 1000), dtype=float32)
```

```python
import tensorflow as tf
print(tf.__version__)
```

```
2.15.0
```

```python
## import some basics libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

```python
dataset=pd.read_csv('Churn_Modelling.csv')
dataset.head()
```

```
   RowNumber  CustomerId   Surname  CreditScore Geography  Gender  Age
\
0          1    15634602  Hargrave          619    France  Female   42

1          2    15647311      Hill          608     Spain  Female   41

2          3    15619304      Onio          502    France  Female   42

3          4    15701354      Boni          699    France  Female   39

4          5    15737888  Mitchell          850     Spain  Female   43


   Tenure    Balance  NumOfProducts  HasCrCard  IsActiveMember  \
0       2       0.00              1          1               1
1       1   83807.86              1          0               1
2       8  159660.80              3          1               0
3       1       0.00              2          0               0
4       2  125510.82              1          1               1

   EstimatedSalary  Exited
0        101348.88       1
1        112542.58       0
2        113931.57       1
3         93826.63       0
4         79084.10       0
```

```python
##Divide the dataset into independent and dependent features
X=dataset.iloc[:,3:13]
y=dataset.iloc[:,13]
```

```python
X.head()
```

```
   CreditScore Geography  Gender  Age  Tenure    Balance
NumOfProducts  \
0          619    France  Female   42       2       0.00
1
1          608     Spain  Female   41       1   83807.86
1
2          502    France  Female   42       8  159660.80
3
3          699    France  Female   39       1       0.00
2
4          850     Spain  Female   43       2  125510.82
1

   HasCrCard  IsActiveMember  EstimatedSalary
0          1               1        101348.88
1          0               1        112542.58
2          1               0        113931.57
3          0               0         93826.63
4          1               1         79084.10
```

```
y
```

```
0       1
1       0
2       1
3       0
4       0
       ..
9995    0
9996    0
9997    1
9998    1
9999    0
Name: Exited, Length: 10000, dtype: int64
```

```
##Feature Engineering

geography=pd.get_dummies(X['Geography'],drop_first=True)
gender=pd.get_dummies(X['Gender'],drop_first=True)

##Concatenate these variables with dataframe

X=X.drop(['Geography','Gender'],axis=1)

X.head()
```

```
   CreditScore  Age  Tenure    Balance  NumOfProducts  HasCrCard  \
0          619   42       2       0.00              1          1
1          608   41       1   83807.86              1          0
2          502   42       8  159660.80              3          1
3          699   39       1       0.00              2          0
```

```
4          850   43       2  125510.82              1          1

   IsActiveMember  EstimatedSalary
0               1        101348.88
1               1        112542.58
2               0        113931.57
3               0         93826.63
4               1         79084.10
```

```
X=pd.concat([X,geography,gender],axis=1)

#Splitting the dataset into Training set and Test Set

from sklearn.model_selection import train_test_split

X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,rando
m_state=0)

##Feature scaling

from sklearn.preprocessing import StandardScaler
sc= StandardScaler()
X_train=sc.fit_transform(X_train)
X_test=sc.transform(X_test)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:432:
UserWarning: X has feature names, but StandardScaler was fitted
without feature names
  warnings.warn(
```

```
X_train
```

```
array([[ 0.16958176, -0.46460796,  0.00666099, ..., -0.5698444 ,
         1.74309049, -1.09168714],
       [-2.30455945,  0.30102557, -1.37744033, ...,  1.75486502,
        -0.57369368,  0.91601335],
       [-1.19119591, -0.94312892, -1.031415  , ..., -0.5698444 ,
        -0.57369368, -1.09168714],
       ...,
       [ 0.9015152 , -0.36890377,  0.00666099, ..., -0.5698444 ,
        -0.57369368,  0.91601335],
       [-0.62420521, -0.08179119,  1.39076231, ..., -0.5698444 ,
         1.74309049, -1.09168714],
       [-0.28401079,  0.87525072, -1.37744033, ...,  1.75486502,
        -0.57369368, -1.09168714]])
```

```
X_test
```

```
array([[ 5.97000000e+02,  3.50000000e+01,  8.00000000e+00, ...,
         1.00000000e+00, -7.10542736e-18, -4.17443857e-17],
       [ 5.23000000e+02,  4.00000000e+01,  2.00000000e+00, ...,
         5.77315973e-17, -7.10542736e-18, -4.17443857e-17],
       [ 7.06000000e+02,  4.20000000e+01,  8.00000000e+00, ...,
         5.77315973e-17,  1.00000000e+00, -4.17443857e-17],
       ...,
       [ 5.78000000e+02,  3.60000000e+01,  1.00000000e+00, ...,
         5.77315973e-17,  1.00000000e+00,  1.00000000e+00],
       [ 6.50000000e+02,  3.40000000e+01,  4.00000000e+00, ...,
         1.00000000e+00, -7.10542736e-18,  1.00000000e+00],
       [ 5.73000000e+02,  3.00000000e+01,  8.00000000e+00, ...,
         1.00000000e+00, -7.10542736e-18,  1.00000000e+00]])
```

```
X_train.shape
```

```
(8000, 11)
```

```
## Part2- Now lets create the ANN
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import LeakyReLU,PReLU,ELU,ReLU
from tensorflow.keras.layers import Dropout
```

```
###lets initialize the ANN
```

```
Classifier=Sequential()
```

```
## Adding the imput Layer
Classifier.add(Dense(units=11,activation='relu'))
```

```
##Adding the first hidden layer
Classifier.add(Dense(units=7,activation='relu'))
```

```
#adding the second hudden layer
Classifier.add(Dense(units=6,activation='relu'))
```

```
#adding output layer
Classifier.add(Dense(1,activation='sigmoid'))
```

```
Classifier.compile(optimizer=opt,loss='binary_crossentropy',metrics=['
accuracy'])
```

```
import tensorflow
opt=tensorflow.keras.optimizers.Adam(learning_rate=0.01)
```

```
##Early Stopping
import tensorflow as tf
early_stopping=tf.keras.callbacks.EarlyStopping(
    monitor="val_loss",
    min_delta=0.0001,
    patience=20,
    verbose=1,
    mode="auto",
    baseline=None,
    restore_best_weights=False,
    start_from_epoch=0,
)

model_history=Classifier.fit(X_train,y_train,validation_split=0.33,bat
ch_size=10,epochs=1000,callbacks=early_stopping)
```

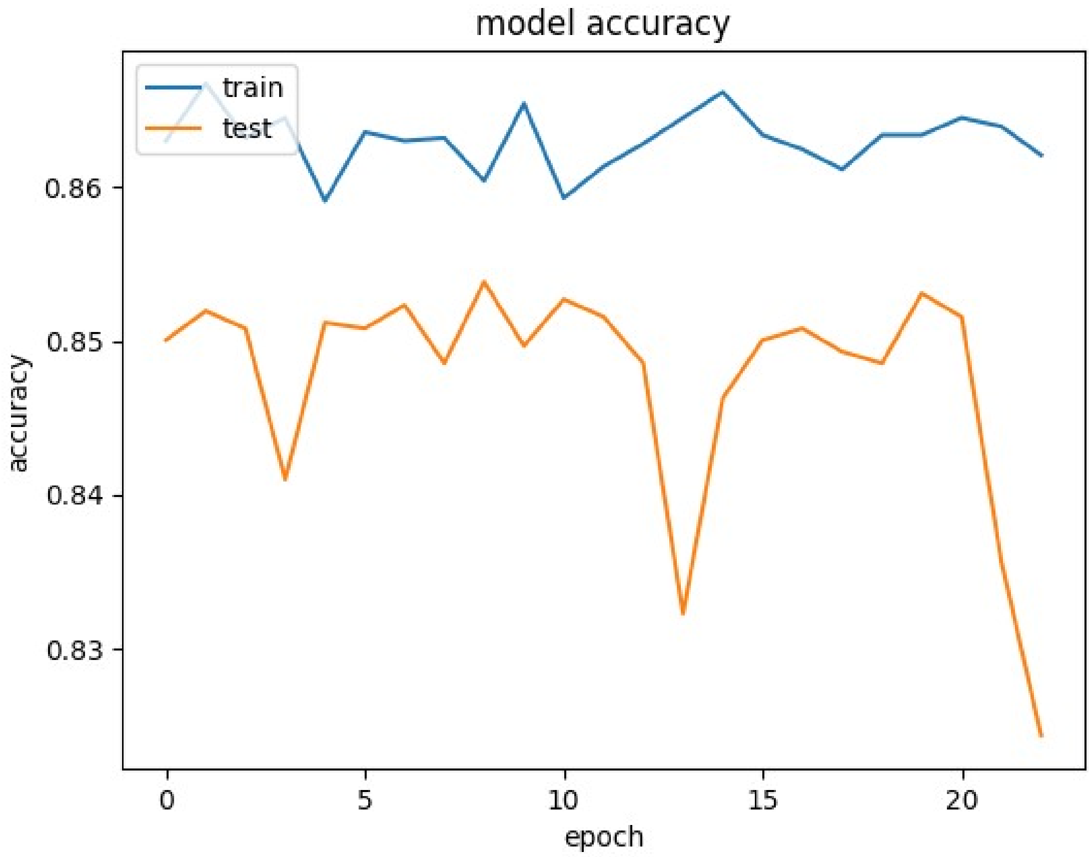
```
Epoch 1/1000
536/536 [==============================] - 2s 4ms/step - loss: 0.3216
- accuracy: 0.8630 - val_loss: 0.3729 - val_accuracy: 0.8501
Epoch 2/1000
536/536 [==============================] - 3s 6ms/step - loss: 0.3203
- accuracy: 0.8668 - val_loss: 0.3670 - val_accuracy: 0.8519
Epoch 3/1000
536/536 [==============================] - 2s 4ms/step - loss: 0.3216
- accuracy: 0.8632 - val_loss: 0.3639 - val_accuracy: 0.8508
Epoch 4/1000
536/536 [==============================] - 3s 5ms/step - loss: 0.3201
- accuracy: 0.8645 - val_loss: 0.3787 - val_accuracy: 0.8410
Epoch 5/1000
536/536 [==============================] - 3s 5ms/step - loss: 0.3211
- accuracy: 0.8591 - val_loss: 0.3655 - val_accuracy: 0.8512
Epoch 6/1000
536/536 [==============================] - 3s 5ms/step - loss: 0.3202
- accuracy: 0.8636 - val_loss: 0.3651 - val_accuracy: 0.8508
Epoch 7/1000
536/536 [==============================] - 4s 8ms/step - loss: 0.3209
- accuracy: 0.8630 - val_loss: 0.3669 - val_accuracy: 0.8523
Epoch 8/1000
536/536 [==============================] - 2s 4ms/step - loss: 0.3194
- accuracy: 0.8632 - val_loss: 0.3682 - val_accuracy: 0.8485
Epoch 9/1000
536/536 [==============================] - 2s 4ms/step - loss: 0.3205
- accuracy: 0.8604 - val_loss: 0.3690 - val_accuracy: 0.8538
Epoch 10/1000
536/536 [==============================] - 2s 4ms/step - loss: 0.3175
- accuracy: 0.8655 - val_loss: 0.3760 - val_accuracy: 0.8497
Epoch 11/1000
536/536 [==============================] - 3s 6ms/step - loss: 0.3204
- accuracy: 0.8593 - val_loss: 0.3687 - val_accuracy: 0.8527
Epoch 12/1000
536/536 [==============================] - 3s 5ms/step - loss: 0.3198
- accuracy: 0.8614 - val_loss: 0.3666 - val_accuracy: 0.8516
Epoch 13/1000
536/536 [==============================] - 2s 4ms/step - loss: 0.3212
- accuracy: 0.8628 - val_loss: 0.3723 - val_accuracy: 0.8485
Epoch 14/1000
536/536 [==============================] - 2s 4ms/step - loss: 0.3201
- accuracy: 0.8645 - val_loss: 0.3796 - val_accuracy: 0.8323
Epoch 15/1000
```

```
536/536 [==============================] - 3s 5ms/step - loss: 0.3193
- accuracy: 0.8662 - val_loss: 0.3677 - val_accuracy: 0.8463
Epoch 16/1000
536/536 [==============================] - 3s 6ms/step - loss: 0.3208
- accuracy: 0.8634 - val_loss: 0.3693 - val_accuracy: 0.8501
Epoch 17/1000
536/536 [==============================] - 3s 5ms/step - loss: 0.3209
- accuracy: 0.8625 - val_loss: 0.3713 - val_accuracy: 0.8508
Epoch 18/1000
536/536 [==============================] - 2s 5ms/step - loss: 0.3219
- accuracy: 0.8612 - val_loss: 0.3716 - val_accuracy: 0.8493
Epoch 19/1000
536/536 [==============================] - 3s 5ms/step - loss: 0.3204
- accuracy: 0.8634 - val_loss: 0.3687 - val_accuracy: 0.8485
Epoch 20/1000
536/536 [==============================] - 3s 6ms/step - loss: 0.3193
- accuracy: 0.8634 - val_loss: 0.3683 - val_accuracy: 0.8531
Epoch 21/1000
536/536 [==============================] - 3s 5ms/step - loss: 0.3205
- accuracy: 0.8645 - val_loss: 0.3756 - val_accuracy: 0.8516
Epoch 22/1000
536/536 [==============================] - 3s 5ms/step - loss: 0.3167
- accuracy: 0.8640 - val_loss: 0.3793 - val_accuracy: 0.8357
Epoch 23/1000
536/536 [==============================] - 2s 4ms/step - loss: 0.3195
- accuracy: 0.8621 - val_loss: 0.3795 - val_accuracy: 0.8243
Epoch 23: early stopping
```

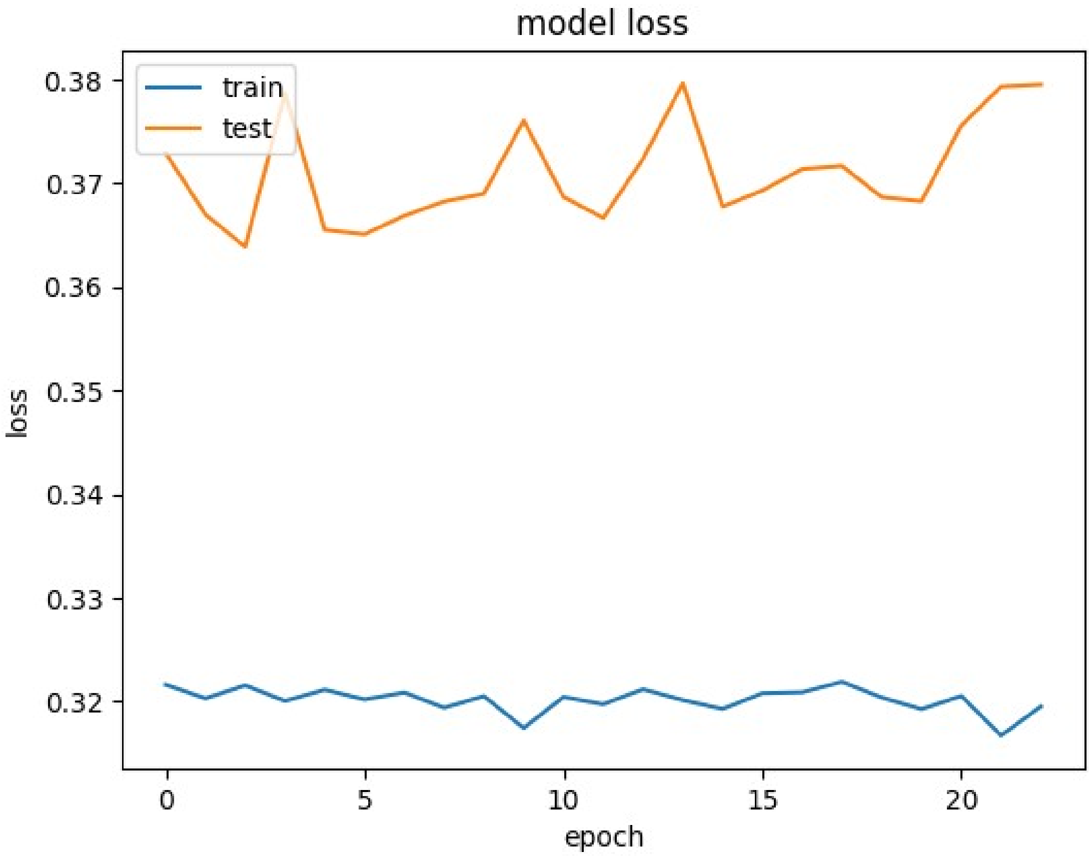
```
model_history.history.keys()
```

```
dict_keys(['loss', 'accuracy', 'val_loss', 'val_accuracy'])
```

```
#summarize history for accuracy
plt.plot(model_history.history['accuracy'])
plt.plot(model_history.history['val_accuracy'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
```



```
# summarize history for loss
plt.plot(model_history.history['loss'])
plt.plot(model_history.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
```



```
# Part 3 - Making the predictions and evaluating the model

# Predicting the Test set results
y_pred = Classifier.predict(X_test)
y_pred = (y_pred > 0.5)

63/63 [==============================] - 0s 4ms/step

## make the confusion metrix

from sklearn.metrics import confusion_matrix
cm=confusion_matrix(y_test,y_pred)
cm

array([[730, 865],
       [139, 266]])

# Calculate the Accuracy
from sklearn.metrics import accuracy_score
score=accuracy_score(y_pred,y_test)

score

0.498
```

```python
## get the weights
Classifier.get_weights()
```

```
[array([[-6.72509551e-01, -1.84281871e-01,  1.12500034e-01,
         -1.81433892e+00, -8.09982002e-01,  1.76370156e+00,
          2.31279278e+00, -5.47874928e-01,  2.78982568e+00,
         -8.15676376e-02, -2.19630742e+00],
        [ 4.70728397e+00, -3.89807850e-01, -4.09404993e+00,
         -2.07181364e-01,  2.88580871e+00, -1.16329718e+00,
         -3.05263829e+00, -2.11800814e+00,  1.81204987e+00,
          1.05808757e-03, -2.98090363e+00],
        [-2.85697073e-01,  6.20624185e-01,  4.66723025e-01,
          1.75992215e+00,  4.37421829e-01,  1.16838582e-01,
         -8.64167035e-01,  2.99453640e+00,  3.56366873e+00,
         -1.84047893e-01,  2.27878761e+00],
        [-1.02047884e+00, -4.06635571e+00, -8.81114721e-01,
          1.57106626e+00, -1.12913764e+00, -2.75600582e-01,
          2.89875650e+00,  9.60043252e-01,  7.88069069e-01,
          1.57982016e+00,  3.26408362e+00],
        [-3.52437019e+00, -5.48682070e+00,  9.12325382e-01,
          3.52715421e+00, -1.69791281e-01,  3.48134851e+00,
          2.25111961e+00, -1.78163588e+00,  1.97066784e+00,
          5.48902845e+00, -1.61501467e+00],
        [-2.34298706e+00, -6.70243979e-01, -1.40832591e+00,
         -7.77164519e-01, -6.49422407e-01,  1.55517626e+00,
          9.26993638e-02, -1.32727683e+00,  1.07449472e+00,
         -2.03437224e-01, -1.94583404e+00],
        [-1.00281692e+00,  6.86129570e-01,  9.08577815e-03,
         -9.94411290e-01,  4.81117868e+00,  1.80351293e+00,
          3.01582432e+00,  3.72218132e+00,  9.33219045e-02,
          1.67438760e-01, -9.32064414e-01],
        [ 2.16485828e-01,  1.30235478e-01, -6.87126100e-01,
         -3.62331939e+00, -5.15279591e-01,  3.00986767e+00,
          1.08365655e+00, -9.81289268e-01, -1.16710246e+00,
          1.47214696e-01, -1.26870525e+00],
        [ 3.76084298e-01,  2.87395501e+00, -1.94282091e+00,
          4.97392595e-01, -7.32620060e-01,  3.05467248e-01,
         -2.46112287e-01,  3.67432326e-01,  1.04490042e+00,
         -1.77958167e+00, -1.45957160e+00],
        [-6.41324759e-01, -2.74460584e-01, -7.71972775e-01,
         -6.38196409e-01,  5.41176915e-01,  5.50887465e-01,
          4.25735079e-02,  1.83118045e+00, -1.89027357e+00,
          3.71938318e-01,  1.40154719e+00],
        [-1.10428102e-01, -1.19963318e-01,  1.17026711e+00,
         -6.04610741e-01,  1.42019665e+00,  1.30864692e+00,
          1.63064048e-01,  5.19107401e-01, -1.28584743e-01,
          2.91206777e-01, -1.20407677e+00]], dtype=float32),
 array([-0.18619463, -2.4928243 ,  5.1261234 , -2.9508593 , -4.0754614
,
        -3.6271155 ,  0.01662346, -2.4706192 , -4.8959513 , -4.277336
```

```
,
        -1.7559004 ], dtype=float32),
 array([[-0.92239684,  0.37103936, -1.3784449 , -7.501407  , -
0.8615784 ,
          1.4035454 ],
        [-0.7046321 , -2.8108506 , -4.5866075 ,  0.36675444, -
0.21041748,
         -0.20127408],
        [ 1.8358886 ,  0.53034216,  1.8308427 ,  0.35606104, -
0.50012803,
         -2.221816  ],
        [-0.09373069, -2.33576   ,  1.568848  , -3.0296187 , -
0.01621949,
          0.9282872 ],
        [-0.53145057,  1.7057312 ,  0.65117496, -1.2876368 , -
0.02406495,
         -3.862307  ],
        [-0.4172956 ,  1.6052992 ,  1.5952787 ,  1.3609582 , -
0.46453026,
          2.1103034 ],
        [-3.4181988 , -0.96016973,  0.24366865,  0.5355145 , -
0.53002965,
         -2.3825674 ],
        [ 1.9648957 , -2.0153368 ,  0.90328765,  1.0305336 , -
0.4076225 ,
          0.98575205],
        [-2.371814  ,  1.8528981 , -1.8947605 , -5.2339053 , -
0.4787469 ,
          0.08510146],
        [ 0.06613193, -1.8432208 , -2.4533746 , -1.3874533 , -
0.05306538,
          4.3157873 ],
        [-0.39467245, -2.9727259 , -1.0642824 , -3.4855046 , -
0.27543354,
         -1.4027644 ]], dtype=float32),
 array([ 1.6702005 , -6.5485625 ,  3.44491   , -0.92504317, -0.6787864
,
        -3.050061  ], dtype=float32),
 array([[-2.3024502 , -0.5547372 ,  0.27347672, -0.00821126, -
0.06514273,
         -0.45833448, -1.4710728 ],
        [ 0.10025582, -0.00645378,  1.5795236 , -0.7306909 , -
0.2858292 ,
         -0.76354474, -1.0672929 ],
        [-0.72517025,  0.0380387 ,  0.8398574 , -0.10435436, -
0.00438156,
         -0.20708339, -0.9149316 ],
        [ 0.65167207, -0.30606323,  1.3041909 , -0.77460784, -
0.1830079 ,
          0.02212565, -1.0740519 ],
        [-0.6097542 ,  0.58453035, -0.39472187, -0.52702916,
0.4870162 ,
          0.06253976, -0.28517613],
        [-0.86673176, -0.11276332, -1.7392315 , -0.2503454 , -
0.42943946,
         -0.6649588 ,  0.883663  ]], dtype=float32),
 array([-3.9933953 , -3.5602343 , -0.5923214 , -0.66967607, -0.1898797
,
        -0.53514206,  1.9213245 ], dtype=float32),
 array([[-0.21315075, -1.8544565 , -0.60921353, -0.28420576, -
0.15054648,
         -0.31307536],
        [-0.05213704, -0.4249323 , -0.20810893, -0.32871756, -
0.62844694,
          0.42495382],
        [ 0.36464247, -0.94677836, -0.17811972, -0.56175035, -
0.83719474,
         -0.39334235],
        [ 0.20393509, -0.16592333, -0.60960275,  0.04187553, -
0.2376461 ,
          0.01504612],
        [-0.38531703,  0.07927102,  0.07064295,  0.20331907, -
0.36400813,
          0.47444415],
        [-0.22164404,  0.18270329, -0.1682508 ,  0.1844728 , -
0.07435948,
         -0.53298396],
        [-3.442308  ,  0.5775392 , -0.39769346, -0.0858357 , -
0.5590735 ,
         -0.5944817 ]], dtype=float32),
 array([ 0.12267611,  0.82916784,  0.        , -0.13760321, -
0.42449898,
        -0.39950797], dtype=float32),
 array([[-0.32271966],
        [ 0.5535804 ],
        [-0.40824097],
        [-0.30143762],
        [-0.2743126 ],
        [ 0.08382902]], dtype=float32),
 array([-0.99778324], dtype=float32)]
```