In [1]:

```
!pip install --upgrade pip setuptools wheel
```

```
Requirement already satisfied: pip in /usr/local/lib/python3.10/dist-packages (24.2)
Requirement already satisfied: setuptools in /usr/local/lib/python3.10/dist-packages (75.2.0)
Requirement already satisfied: wheel in /usr/local/lib/python3.10/dist-packages (0.44.0)
```

In [2]:

```
!pip cache purge
```

```
Files removed: 18
```

In [1]:

```
!pip install tensorflow-gpu==2.10.0
```

```
Requirement already satisfied: tensorflow-gpu==2.10.0 in /usr/local/lib/python3.10/dist-packages (2.10.0)
Requirement already satisfied: absl-py>=1.0.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow-gpu==2.10.0) (1.4.0)
Requirement already satisfied: astunparse>=1.6.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow-gpu==2.10.0) (1.6.3)
Requirement already satisfied: flatbuffers>=2.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow-gpu==2.10.0) (24.3.25)
Requirement already satisfied: gast<=0.4.0,>=0.2.1 in /usr/local/lib/python3.10/dist-packages (from tensorflow-gpu==2.10.0) (0.4.0
)
Requirement already satisfied: google-pasta>=0.1.1 in /usr/local/lib/python3.10/dist-packages (from tensorflow-gpu==2.10.0) (0.2.0
)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in /usr/local/lib/python3.10/dist-packages (from tensorflow-gpu==2.10.0) (1.64.
1)
Requirement already satisfied: h5py>=2.9.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow-gpu==2.10.0) (3.11.0)
Requirement already satisfied: keras<2.11,>=2.10.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow-gpu==2.10.0) (2.10.
0)
Requirement already satisfied: keras-preprocessing>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from tensorflow-gpu==2.10.0)
(1.1.2)
Requirement already satisfied: libclang>=13.0.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow-gpu==2.10.0) (18.1.1)
Requirement already satisfied: numpy>=1.20 in /usr/local/lib/python3.10/dist-packages (from tensorflow-gpu==2.10.0) (1.26.4)
Requirement already satisfied: opt-einsum>=2.3.2 in /usr/local/lib/python3.10/dist-packages (from tensorflow-gpu==2.10.0) (3.4.0)
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages (from tensorflow-gpu==2.10.0) (24.1)
Requirement already satisfied: protobuf<3.20,>=3.9.2 in /usr/local/lib/python3.10/dist-packages (from tensorflow-gpu==2.10.0) (3.1
9.6)
Requirement already satisfied: setuptools in /usr/local/lib/python3.10/dist-packages (from tensorflow-gpu==2.10.0) (75.2.0)
Requirement already satisfied: six>=1.12.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow-gpu==2.10.0) (1.16.0)
Requirement already satisfied: tensorboard<2.11,>=2.10 in /usr/local/lib/python3.10/dist-packages (from tensorflow-gpu==2.10.0) (2
.10.1)
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in /usr/local/lib/python3.10/dist-packages (from tensorflow-gp
u==2.10.0) (0.37.1)
Requirement already satisfied: tensorflow-estimator<2.11,>=2.10.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow-gpu=
=2.10.0) (2.10.0)
Requirement already satisfied: termcolor>=1.1.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow-gpu==2.10.0) (2.5.0)
```

```
Requirement already satisfied: typing-extensions>=3.6.6 in /usr/local/lib/python3.10/dist-packages (from tensorflow-gpu==2.10.0) (
4.12.2)
Requirement already satisfied: wrapt>=1.11.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow-gpu==2.10.0) (1.16.0)
Requirement already satisfied: wheel<1.0,>=0.23.0 in /usr/local/lib/python3.10/dist-packages (from astunparse>=1.6.0->tensorflow-g
pu==2.10.0) (0.44.0)
Requirement already satisfied: google-auth<3,>=1.6.3 in /usr/local/lib/python3.10/dist-packages (from tensorboard<2.11,>=2.10->ten
sorflow-gpu==2.10.0) (2.27.0)
Requirement already satisfied: google-auth-oauthlib<0.5,>=0.4.1 in /usr/local/lib/python3.10/dist-packages (from tensorboard<2.11,
>=2.10->tensorflow-gpu==2.10.0) (0.4.6)
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.10/dist-packages (from tensorboard<2.11,>=2.10->tensorflo
w-gpu==2.10.0) (3.7)
Requirement already satisfied: requests<3,>=2.21.0 in /usr/local/lib/python3.10/dist-packages (from tensorboard<2.11,>=2.10->tenso
rflow-gpu==2.10.0) (2.32.3)
Requirement already satisfied: tensorboard-data-server<0.7.0,>=0.6.0 in /usr/local/lib/python3.10/dist-packages (from tensorboard<
2.11,>=2.10->tensorflow-gpu==2.10.0) (0.6.1)
Requirement already satisfied: tensorboard-plugin-wit>=1.6.0 in /usr/local/lib/python3.10/dist-packages (from tensorboard<2.11,>=2
.10->tensorflow-gpu==2.10.0) (1.8.1)
Requirement already satisfied: werkzeug>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from tensorboard<2.11,>=2.10->tensorflo
w-gpu==2.10.0) (3.0.4)
Requirement already satisfied: cachetools<6.0,>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from google-auth<3,>=1.6.3->tens
orboard<2.11,>=2.10->tensorflow-gpu==2.10.0) (5.5.0)
Requirement already satisfied: pyasn1-modules>=0.2.1 in /usr/local/lib/python3.10/dist-packages (from google-auth<3,>=1.6.3->tenso
rboard<2.11,>=2.10->tensorflow-gpu==2.10.0) (0.4.1)
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
.11,>=2.10->tensorflow-gpu==2.10.0) (4.9)
Requirement already satisfied: requests-oauthlib>=0.7.0 in /usr/local/lib/python3.10/dist-packages (from google-auth-oauthlib<0.5,
>=0.4.1->tensorboard<2.11,>=2.10->tensorflow-gpu==2.10.0) (1.3.1)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2.21.0->tens
orboard<2.11,>=2.10->tensorflow-gpu==2.10.0) (3.4.0)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2.21.0->tensorboard<2.11
,>=2.10->tensorflow-gpu==2.10.0) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2.21.0->tensorboar
d<2.11,>=2.10->tensorflow-gpu==2.10.0) (2.2.3)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2.21.0->tensorboar
d<2.11,>=2.10->tensorflow-gpu==2.10.0) (2024.8.30)
Requirement already satisfied: MarkupSafe>=2.1.1 in /usr/local/lib/python3.10/dist-packages (from werkzeug>=1.0.1->tensorboard<2.1
1,>=2.10->tensorflow-gpu==2.10.0) (3.0.1)
Requirement already satisfied: pyasn1<0.7.0,>=0.4.6 in /usr/local/lib/python3.10/dist-packages (from pyasn1-modules>=0.2.1->google
-auth<3,>=1.6.3->tensorboard<2.11,>=2.10->tensorflow-gpu==2.10.0) (0.6.1)
Requirement already satisfied: oauthlib>=3.0.0 in /usr/local/lib/python3.10/dist-packages (from requests-oauthlib>=0.7.0->google-a
uth-oauthlib<0.5,>=0.4.1->tensorboard<2.11,>=2.10->tensorflow-gpu==2.10.0) (3.2.2)
```

In [1]:

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

```
Collecting tensorflow-gpu
  Downloading 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
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.
```

In [2]:

```
import tensorflow as tf
```

In [3]:

```
print(tf.__version__)
```

```
2.10.0
```

In [4]:

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

In [5]:

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

Out[5]:

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0	1	15634602	Hargrave	619	France	Female	42	2	0.00	1	1	1	101348.88	1
1	2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	1	112542.58	0
2	3	15619304	Onio	502	France	Female	42	8	159660.80	3	1	0	113931.57	1
3	4	15701354	Boni	699	France	Female	39	1	0.00	2	0	0	93826.63	0
4	5	15737888	Mitchell	850	Spain	Female	43	2	125510.82	1	1	1	79084.10	0

In [9]:

```
X = dataset.iloc[:,3:13]
y= dataset.iloc[:,13]
```

In [11]:

```
y.head()
```

Out[11]:

	Exited
0	**1**
1	**0**
2	**1**
3	**0**
4	**0**

dtype: int64

In [12]:

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

In [13]:

```
##concatenate these variables with dataframe
X.drop(['Geography','Gender'],inplace =True, axis=1)
```

In [14]:

```
X.head()
```

Out[14]:

	CreditScore	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary
0	**619**	**42**	**2**	**0.00**	**1**	**1**	**1**	**101348.88**
1	**608**	**41**	**1**	**83807.86**	**1**	**0**	**1**	**112542.58**
2	**502**	**42**	**8**	**159660.80**	**3**	**1**	**0**	**113931.57**
3	**699**	**39**	**1**	**0.00**	**2**	**0**	**0**	**93826.63**
4	**850**	**43**	**2**	**125510.82**	**1**	**1**	**1**	**79084.10**

In [16]:

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

In [17]:

```
X.head()
```

Out[17]:

	CreditScore	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Germany	Spain	Male
0	619	42	2	0.00	1	1	1	101348.88	False	False	False
1	608	41	1	83807.86	1	0	1	112542.58	False	True	False
2	502	42	8	159660.80	3	1	0	113931.57	False	False	False
3	699	39	1	0.00	2	0	0	93826.63	False	False	False
4	850	43	2	125510.82	1	1	1	79084.10	False	True	False

In [18]:

```
##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, random_state =0)
```

In [19]:

```
#feature scaling

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

In [22]:

```
X_test.shape
```

Out[22]:

```
(2000, 11)
```

In [25]:

```
### 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
```

In [55]:

```
###Let's initialize the ANN

classifier = Sequential()
```

In [56]:

```
## Adding the input layer
classifier.add(Dense(units = 11,activation='relu'))
```

In [57]:

```
## Adding the first hidden layer
classifier.add(Dense(units =7,activation ='relu'))
classifier.add(Dropout(0.2))
```

In [58]:

```
## adding the second hidden layer
classifier.add(Dense(units =6,activation ='relu'))
classifier.add(Dropout(0.3))
```

In [59]:

```
#Adding the output layer
classifier.add(Dense(units =1,activation='sigmoid'))
```

In [34]:

```

```

In [60]:

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

In [61]:

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

In [39]:

```
model_history = classifier.fit(X_train,y_train,validation_split=0.33, batch_size =10,epochs =20)
```

```
Epoch 1/20
536/536 [==============================] - 2s 4ms/step - loss: 0.2902 - accuracy: 0.8830 - val_loss: 0.4345 - val_accuracy: 0.8330
Epoch 2/20
536/536 [==============================] - 1s 3ms/step - loss: 0.2946 - accuracy: 0.8796 - val_loss: 0.4160 - val_accuracy: 0.8349
Epoch 3/20
536/536 [==============================] - 1s 2ms/step - loss: 0.2926 - accuracy: 0.8781 - val_loss: 0.4311 - val_accuracy: 0.8357
Epoch 4/20
536/536 [==============================] - 1s 2ms/step - loss: 0.2902 - accuracy: 0.8804 - val_loss: 0.4389 - val_accuracy: 0.8387
Epoch 5/20
536/536 [==============================] - 1s 2ms/step - loss: 0.2888 - accuracy: 0.8796 - val_loss: 0.4553 - val_accuracy: 0.8360
Epoch 6/20
536/536 [==============================] - 1s 2ms/step - loss: 0.2922 - accuracy: 0.8809 - val_loss: 0.4532 - val_accuracy: 0.8342
Epoch 7/20
536/536 [==============================] - 1s 3ms/step - loss: 0.2933 - accuracy: 0.8778 - val_loss: 0.4395 - val_accuracy: 0.8342
Epoch 8/20
536/536 [==============================] - 1s 2ms/step - loss: 0.2911 - accuracy: 0.8802 - val_loss: 0.4165 - val_accuracy: 0.8360
Epoch 9/20
536/536 [==============================] - 2s 3ms/step - loss: 0.2909 - accuracy: 0.8793 - val_loss: 0.4576 - val_accuracy: 0.8387
Epoch 10/20
536/536 [==============================] - 3s 5ms/step - loss: 0.2934 - accuracy: 0.8819 - val_loss: 0.4557 - val_accuracy: 0.8315
Epoch 11/20
536/536 [==============================] - 2s 4ms/step - loss: 0.2910 - accuracy: 0.8802 - val_loss: 0.4277 - val_accuracy: 0.8395
Epoch 12/20
536/536 [==============================] - 1s 2ms/step - loss: 0.2913 - accuracy: 0.8800 - val_loss: 0.4482 - val_accuracy: 0.8368
Epoch 13/20
536/536 [==============================] - 1s 2ms/step - loss: 0.2893 - accuracy: 0.8823 - val_loss: 0.4500 - val_accuracy: 0.8410
Epoch 14/20
536/536 [==============================] - 1s 2ms/step - loss: 0.2988 - accuracy: 0.8798 - val_loss: 0.4252 - val_accuracy: 0.8398
Epoch 15/20
536/536 [==============================] - 1s 2ms/step - loss: 0.2923 - accuracy: 0.8791 - val_loss: 0.4156 - val_accuracy: 0.8315
Epoch 16/20
536/536 [==============================] - 1s 2ms/step - loss: 0.2905 - accuracy: 0.8800 - val_loss: 0.4412 - val_accuracy: 0.8342
Epoch 17/20
536/536 [==============================] - 1s 2ms/step - loss: 0.2897 - accuracy: 0.8791 - val_loss: 0.4487 - val_accuracy: 0.8383
Epoch 18/20
536/536 [==============================] - 1s 2ms/step - loss: 0.2893 - accuracy: 0.8800 - val_loss: 0.4170 - val_accuracy: 0.8360
Epoch 19/20
536/536 [==============================] - 2s 3ms/step - loss: 0.2940 - accuracy: 0.8796 - val_loss: 0.4135 - val_accuracy: 0.8421
Epoch 20/20
536/536 [==============================] - 2s 4ms/step - loss: 0.2916 - accuracy: 0.8776 - val_loss: 0.4340 - val_accuracy: 0.8360
```

In [41]:

```
model_history.history
```

Out[41]:

```
{'loss': [0.29016390442848206,
  0.2945784032344818,
  0.292606383562088,
  0.290227472782135,
```

```
 0.2887743413448334,
 0.29215991497039795,
 0.29333874583244324,
 0.2911457419395447,
 0.2909436523914337,
 0.29336604475975037,
 0.2910258173942566,
 0.29132258892059326,
 0.2892898917198181,
 0.2987709641456604,
 0.2922622859477997,
 0.2904742360115051,
 0.2896958291530609,
 0.28933313488960266,
 0.29396653175354004,
 0.29161694645881653],
'accuracy': [0.8830005526542664,
 0.8796417117118835,
 0.8781489133834839,
 0.8803881406784058,
 0.8796417117118835,
 0.8809479475021362,
 0.8777757287025452,
 0.880201518535614,
 0.8792685270309448,
 0.8818809390068054,
 0.880201518535614,
 0.880014955997467,
 0.8822541236877441,
 0.8798283338546753,
 0.8790819048881531,
 0.880014955997467,
 0.8790819048881531,
 0.880014955997467,
 0.8796417117118835,
 0.8775891065597534],
'val_loss': [0.4344862401485443,
 0.416005402803421,
 0.4310718774795532,
 0.4389435350894928,
 0.45525386929512024,
 0.4532302916049957,
 0.43947622179985046,
 0.41651248931884766,
 0.4575900137424469,
 0.4556894600391388,
 0.42771726846694946,
 0.4481942653656006,
 0.4500405192375183,
 0.4252464771270752,
```

```
  0.41558647155761 72,
  0.4411742091178894,
  0.4486519396305084,
  0.4170237183570862,
  0.4135344624519348,
  0.4340309500694275],
 'val_accuracy': [0.8330178260803223,
  0.8349109888076782,
  0.8356683254241943,
  0.8386974334716797,
  0.8360469341278076,
  0.8341537117958069,
  0.8341537117958069,
  0.8360469341278076,
  0.8386974334716797,
  0.8315032124519348,
  0.8394547700881958,
  0.836804211139679,
  0.8409693241119385,
  0.8398333787918091,
  0.8315032124519348,
  0.8341537117958069,
  0.8383188247680664,
  0.8360469341278076,
  0.8421052694320679,
  0.8360469341278076]}
```

In [62]:

```
##Early stopping (when the accuracy is not increaing automatically training of the model will stop)
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
)

#ref- https://keras.io/api/callbacks/early_stopping/
```

In [63]:

```
model_history = classifier.fit(X_train,y_train,validation_split=0.33, batch_size =10,epochs =1000, callbacks=early_stopping)
```

```
Epoch 1/1000
536/536 [==============================] - 3s 5ms/step - loss: 0.4691 - accuracy: 0.7975 - val_loss: 0.4467 - val_accuracy: 0.7747
Epoch 2/1000
```

```
536/536 [==============================] - 3s 5ms/step - loss: 0.4288 - accuracy: 0.8097 - val_loss: 0.4189 - val_accuracy: 0.7982
Epoch 3/1000
536/536 [==============================] - 2s 3ms/step - loss: 0.4002 - accuracy: 0.8242 - val_loss: 0.4133 - val_accuracy: 0.8270
Epoch 4/1000
536/536 [==============================] - 1s 3ms/step - loss: 0.3979 - accuracy: 0.8339 - val_loss: 0.3863 - val_accuracy: 0.8353
Epoch 5/1000
536/536 [==============================] - 1s 2ms/step - loss: 0.3854 - accuracy: 0.8358 - val_loss: 0.3800 - val_accuracy: 0.8395
Epoch 6/1000
536/536 [==============================] - 1s 2ms/step - loss: 0.3873 - accuracy: 0.8364 - val_loss: 0.3887 - val_accuracy: 0.8357
Epoch 7/1000
536/536 [==============================] - 1s 2ms/step - loss: 0.3867 - accuracy: 0.8412 - val_loss: 0.4074 - val_accuracy: 0.8239
Epoch 8/1000
536/536 [==============================] - 1s 2ms/step - loss: 0.3887 - accuracy: 0.8386 - val_loss: 0.3793 - val_accuracy: 0.8413
Epoch 9/1000
536/536 [==============================] - 1s 2ms/step - loss: 0.3814 - accuracy: 0.8365 - val_loss: 0.3924 - val_accuracy: 0.8311
Epoch 10/1000
536/536 [==============================] - 1s 2ms/step - loss: 0.3821 - accuracy: 0.8442 - val_loss: 0.3873 - val_accuracy: 0.8345
Epoch 11/1000
536/536 [==============================] - 1s 2ms/step - loss: 0.3773 - accuracy: 0.8451 - val_loss: 0.3919 - val_accuracy: 0.8376
Epoch 12/1000
536/536 [==============================] - 2s 3ms/step - loss: 0.3795 - accuracy: 0.8410 - val_loss: 0.3798 - val_accuracy: 0.8413
Epoch 13/1000
536/536 [==============================] - 2s 4ms/step - loss: 0.3854 - accuracy: 0.8380 - val_loss: 0.3822 - val_accuracy: 0.8402
Epoch 14/1000
536/536 [==============================] - 2s 3ms/step - loss: 0.3812 - accuracy: 0.8410 - val_loss: 0.3930 - val_accuracy: 0.8357
Epoch 15/1000
536/536 [==============================] - 1s 2ms/step - loss: 0.3796 - accuracy: 0.8375 - val_loss: 0.3818 - val_accuracy: 0.8410
Epoch 16/1000
536/536 [==============================] - 1s 2ms/step - loss: 0.3698 - accuracy: 0.8427 - val_loss: 0.3898 - val_accuracy: 0.8463
Epoch 17/1000
536/536 [==============================] - 1s 2ms/step - loss: 0.3790 - accuracy: 0.8406 - val_loss: 0.3759 - val_accuracy: 0.8417
Epoch 18/1000
536/536 [==============================] - 1s 2ms/step - loss: 0.3663 - accuracy: 0.8507 - val_loss: 0.3777 - val_accuracy: 0.8440
Epoch 19/1000
536/536 [==============================] - 1s 2ms/step - loss: 0.3768 - accuracy: 0.8410 - val_loss: 0.3748 - val_accuracy: 0.8387
Epoch 20/1000
536/536 [==============================] - 1s 2ms/step - loss: 0.3704 - accuracy: 0.8483 - val_loss: 0.3853 - val_accuracy: 0.8387
Epoch 21/1000
536/536 [==============================] - 1s 2ms/step - loss: 0.3715 - accuracy: 0.8474 - val_loss: 0.3904 - val_accuracy: 0.8387
Epoch 22/1000
536/536 [==============================] - 2s 3ms/step - loss: 0.3738 - accuracy: 0.8427 - val_loss: 0.3853 - val_accuracy: 0.8429
Epoch 23/1000
536/536 [==============================] - 2s 4ms/step - loss: 0.3703 - accuracy: 0.8459 - val_loss: 0.3813 - val_accuracy: 0.8413
Epoch 24/1000
536/536 [==============================] - 1s 2ms/step - loss: 0.3672 - accuracy: 0.8498 - val_loss: 0.3727 - val_accuracy: 0.8425
Epoch 25/1000
536/536 [==============================] - 1s 2ms/step - loss: 0.3686 - accuracy: 0.8459 - val_loss: 0.3804 - val_accuracy: 0.8323
Epoch 26/1000
536/536 [==============================] - 1s 2ms/step - loss: 0.3683 - accuracy: 0.8470 - val_loss: 0.3844 - val_accuracy: 0.8406
Epoch 27/1000
```

```
536/536 [==============================] - 1s 2ms/step - loss: 0.3663 - accuracy: 0.8472 - val_loss: 0.3849 - val_accuracy: 0.8383
Epoch 28/1000
536/536 [==============================] - 2s 3ms/step - loss: 0.3669 - accuracy: 0.8455 - val_loss: 0.3827 - val_accuracy: 0.8338
Epoch 29/1000
536/536 [==============================] - 2s 3ms/step - loss: 0.3654 - accuracy: 0.8474 - val_loss: 0.3847 - val_accuracy: 0.8395
Epoch 30/1000
536/536 [==============================] - 2s 3ms/step - loss: 0.3673 - accuracy: 0.8462 - val_loss: 0.3925 - val_accuracy: 0.8296
Epoch 31/1000
536/536 [==============================] - 2s 3ms/step - loss: 0.3631 - accuracy: 0.8487 - val_loss: 0.3895 - val_accuracy: 0.8421
Epoch 32/1000
536/536 [==============================] - 2s 4ms/step - loss: 0.3642 - accuracy: 0.8457 - val_loss: 0.3794 - val_accuracy: 0.8372
Epoch 33/1000
536/536 [==============================] - 1s 2ms/step - loss: 0.3696 - accuracy: 0.8442 - val_loss: 0.3781 - val_accuracy: 0.8436
Epoch 34/1000
536/536 [==============================] - 1s 2ms/step - loss: 0.3637 - accuracy: 0.8444 - val_loss: 0.3751 - val_accuracy: 0.8413
Epoch 35/1000
536/536 [==============================] - 1s 2ms/step - loss: 0.3589 - accuracy: 0.8464 - val_loss: 0.3874 - val_accuracy: 0.8417
Epoch 36/1000
536/536 [==============================] - 1s 2ms/step - loss: 0.3619 - accuracy: 0.8438 - val_loss: 0.3812 - val_accuracy: 0.8387
Epoch 37/1000
536/536 [==============================] - 1s 2ms/step - loss: 0.3600 - accuracy: 0.8451 - val_loss: 0.3867 - val_accuracy: 0.8425
Epoch 38/1000
536/536 [==============================] - 1s 2ms/step - loss: 0.3688 - accuracy: 0.8447 - val_loss: 0.3819 - val_accuracy: 0.8421
Epoch 39/1000
536/536 [==============================] - 1s 2ms/step - loss: 0.3622 - accuracy: 0.8447 - val_loss: 0.3707 - val_accuracy: 0.8432
Epoch 40/1000
536/536 [==============================] - 1s 2ms/step - loss: 0.3600 - accuracy: 0.8487 - val_loss: 0.3779 - val_accuracy: 0.8440
Epoch 41/1000
536/536 [==============================] - 2s 3ms/step - loss: 0.3619 - accuracy: 0.8444 - val_loss: 0.3805 - val_accuracy: 0.8421
Epoch 42/1000
536/536 [==============================] - 2s 4ms/step - loss: 0.3682 - accuracy: 0.8444 - val_loss: 0.3793 - val_accuracy: 0.8402
Epoch 43/1000
536/536 [==============================] - 1s 2ms/step - loss: 0.3633 - accuracy: 0.8438 - val_loss: 0.3939 - val_accuracy: 0.8338
Epoch 44/1000
536/536 [==============================] - 1s 3ms/step - loss: 0.3622 - accuracy: 0.8457 - val_loss: 0.3775 - val_accuracy: 0.8398
Epoch 45/1000
536/536 [==============================] - 1s 2ms/step - loss: 0.3617 - accuracy: 0.8507 - val_loss: 0.3752 - val_accuracy: 0.8398
Epoch 46/1000
536/536 [==============================] - 1s 2ms/step - loss: 0.3577 - accuracy: 0.8509 - val_loss: 0.3746 - val_accuracy: 0.8402
Epoch 47/1000
536/536 [==============================] - 1s 2ms/step - loss: 0.3619 - accuracy: 0.8477 - val_loss: 0.3833 - val_accuracy: 0.8391
Epoch 48/1000
536/536 [==============================] - 1s 2ms/step - loss: 0.3665 - accuracy: 0.8502 - val_loss: 0.3866 - val_accuracy: 0.8364
Epoch 49/1000
536/536 [==============================] - 1s 2ms/step - loss: 0.3678 - accuracy: 0.8431 - val_loss: 0.3925 - val_accuracy: 0.8413
Epoch 50/1000
536/536 [==============================] - 1s 2ms/step - loss: 0.3613 - accuracy: 0.8475 - val_loss: 0.3779 - val_accuracy: 0.8391
Epoch 51/1000
536/536 [==============================] - 2s 4ms/step - loss: 0.3583 - accuracy: 0.8507 - val_loss: 0.3796 - val_accuracy: 0.8395
Epoch 52/1000
536/536 [==============================] - 1s 3ms/step - loss: 0.3593 - accuracy: 0.8455 - val_loss: 0.3883 - val_accuracy: 0.8432
```

```
Epoch 53/1000
536/536 [==============================] - 1s 2ms/step - loss: 0.3629 - accuracy: 0.8474 - val_loss: 0.3760 - val_accuracy: 0.8417
Epoch 54/1000
536/536 [==============================] - 1s 2ms/step - loss: 0.3598 - accuracy: 0.8483 - val_loss: 0.3815 - val_accuracy: 0.8436
Epoch 55/1000
536/536 [==============================] - 1s 2ms/step - loss: 0.3635 - accuracy: 0.8472 - val_loss: 0.3834 - val_accuracy: 0.8417
Epoch 56/1000
536/536 [==============================] - 1s 2ms/step - loss: 0.3610 - accuracy: 0.8433 - val_loss: 0.3797 - val_accuracy: 0.8455
Epoch 57/1000
536/536 [==============================] - 1s 2ms/step - loss: 0.3619 - accuracy: 0.8461 - val_loss: 0.3824 - val_accuracy: 0.8470
Epoch 58/1000
536/536 [==============================] - 1s 2ms/step - loss: 0.3630 - accuracy: 0.8440 - val_loss: 0.3763 - val_accuracy: 0.8466
Epoch 59/1000
536/536 [==============================] - 1s 2ms/step - loss: 0.3625 - accuracy: 0.8459 - val_loss: 0.3896 - val_accuracy: 0.8387
Epoch 59: early stopping
```

In [64]:

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

Out[64]:

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

In [65]:

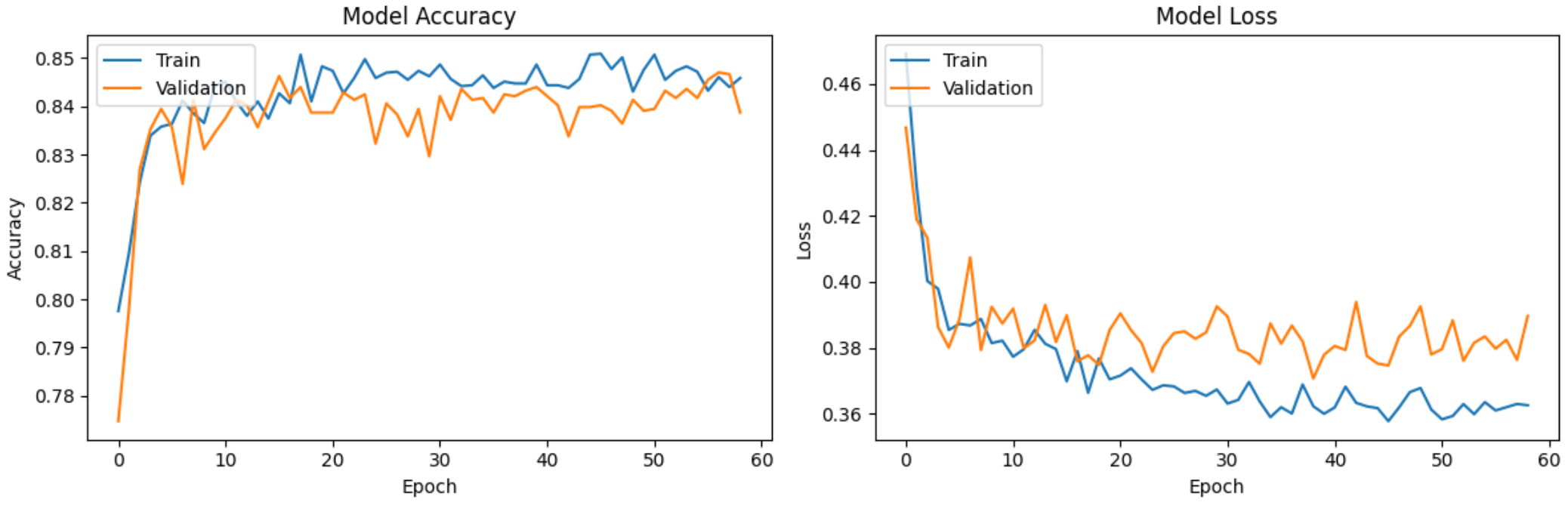
```
import matplotlib.pyplot as plt

# Assuming model_history is the history object from your model training
history = model_history.history

# Plotting training & validation accuracy values
plt.figure(figsize=(12, 4))
plt.subplot(1, 2, 1)
plt.plot(history['accuracy'])
plt.plot(history['val_accuracy'])
plt.title('Model Accuracy')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend(['Train', 'Validation'], loc='upper left')

# Plotting training & validation loss values
plt.subplot(1, 2, 2)
plt.plot(history['loss'])
plt.plot(history['val_loss'])
plt.title('Model Loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Validation'], loc='upper left')
```

```
plt.tight_layout()
plt.show()
```



In [66]:

```
##Making predictions for the test data
y_pred = classifier.predict(X_test)
y_pred =(y_pred>=0.5)
```

```
63/63 [==============================] - 0s 1ms/step
```

In [67]:

```
##confusion metrics
from sklearn.metrics import confusion_matrix

cm = confusion_matrix(y_test, y_pred)
cm
```

Out[67]:

```
array([[1587,   29],
       [ 256,  128]])
```

In [68]:

```
#calculate the accuracy
```

```
from sklearn.metrics import accuracy_score

score = accuracy_score(y_pred,y_test)
score
```

Out[68]:

```
0.8575
```

In [69]:

```
## get the weights store in pickle
classifier.get_weights()
```

Out[69]:

```
[array([[ 2.2004254e+00,  1.9332002e+00,  1.7303725e-01, -1.8794774e-01,
          9.5149353e-02,  7.7215672e-02,  8.9216143e-01, -4.2458412e-01,
         -8.1445026e-01,  3.7007439e-01,  3.0413099e-02],
        [-4.1016918e-03, -2.8584118e+00,  1.1839387e+00,  2.2792106e+00,
         -2.6591551e+00,  2.7383659e+00, -3.6347618e+00, -2.9518035e-01,
         -2.0268962e+00, -2.8307855e+00, -3.5196011e+00],
        [ 2.1887434e-01,  1.3614721e+00,  1.9822623e-01,  5.6371468e-01,
          1.0556438e+00, -3.2949943e-02, -2.2061501e-01, -2.8926608e-01,
          2.3679996e-01, -8.2679570e-01,  3.0304375e-01],
        [-1.5465878e+00, -1.0534208e+00, -3.8656442e+00,  1.1283741e+00,
          3.4647176e-01, -7.9473633e-01, -1.1359247e+00,  9.4928074e-01,
          2.5931866e+00, -7.1383035e-01, -1.3972195e+00],
        [ 3.9857453e-01, -5.3440595e-01, -4.1505246e+00, -9.2173660e-01,
         -3.6926544e-01, -2.4598654e-01, -1.0585510e+00,  4.2999978e+00,
         -3.2729024e-01,  1.2105113e+00,  5.4735935e-01],
        [ 6.1103773e-01, -3.8528240e-01, -3.4709117e-01,  4.7636607e-01,
          1.4641877e-01,  1.7871058e-01,  1.3361884e+00, -3.9370558e-01,
          7.0327598e-01,  1.0633464e+00, -5.7647580e-01],
        [ 9.4912440e-01,  1.8681848e-01, -5.7814795e-01, -2.6518886e+00,
          4.0346342e-03,  1.9795613e+00,  1.2409623e+00,  1.4174433e-01,
         -2.5837305e+00, -5.9706134e-01, -3.6651209e-01],
        [-8.4151097e-02,  5.3832269e-01,  6.3130862e-01,  1.2172312e-01,
          1.9682970e-02,  3.1386986e-02,  6.7307293e-01,  4.0746659e-02,
          2.2453654e-01, -1.9039582e+00,  1.2188110e-01],
        [-5.5554503e-01,  1.4771353e-01,  1.9922380e+00, -5.8441913e-01,
         -2.8836775e+00,  3.1583837e-01,  2.1082149e+00, -4.3935740e-01,
          9.0494484e-01, -1.1984712e-01,  1.4136431e-01],
        [ 1.3268495e-01, -2.3264050e+00, -3.2028067e-01,  1.5146647e-01,
          5.6561959e-01,  1.3421734e-01, -6.2697595e-01,  6.6289699e-01,
          9.3345135e-01,  7.7784228e-01, -6.2257219e-02],
        [ 1.5364350e+00, -8.1729084e-01, -5.6603193e-01, -1.0584930e+00,
         -7.7876911e-02, -6.1676875e-02, -2.5400308e-01, -4.7864699e-01,
         -9.8980641e-01,  1.5621022e+00,  3.7419957e-01]], dtype=float32),
 array([-2.403252 , -2.6347258, -1.0716164, -2.160227 ,  1.1529032,
```

```
        -4.90253  , -1.9564831, -5.113029 , -3.1557212, -3.5590944,
         3.1479294], dtype=float32),
 array([[ 0.10436518,  1.0036105 ,  0.9807755 ,  0.6555331 , -0.14224064,
         -0.5683991 ,  0.87253076],
        [-0.3239811 ,  0.48419607, -0.890777  , -0.8642794 ,  0.17934859,
         -1.0597662 , -0.7778416 ],
        [ 1.1802466 ,  1.1080325 , -1.7646296 , -3.9420257 , -0.9210848 ,
         -1.276027  , -1.9091287 ],
        [ 0.55593354,  0.42306998, -1.8407978 , -2.8010762 , -0.10155237,
          0.31222406, -0.13247631],
        [-1.6114855 , -1.0204035 ,  0.25413764,  0.3212783 , -0.54135275,
          1.0195493 ,  0.03844799],
        [-2.4349575 , -3.1175797 , -0.50261664,  0.05860481, -0.11561491,
          2.0610158 ,  1.5056928 ],
        [-1.6932385 , -3.180799  ,  0.51483274,  0.30560437,  0.04058821,
         -1.1685208 , -0.21702763],
        [ 3.048175  ,  3.4889743 , -1.2732706 , -2.2082412 , -0.12531677,
         -1.4543037 , -2.561604  ],
        [-0.29808313, -0.37960398, -1.2015309 , -3.0375514 , -0.04039068,
         -2.2848115 ,  0.07014931],
        [-1.1254883 , -1.5130044 , -0.3310763 , -2.0600111 , -0.6342547 ,
          1.1110755 , -1.7220602 ],
        [-1.9270498 , -1.0113522 ,  0.8247402 ,  0.37766019, -0.4506791 ,
          0.26526234,  1.2612394 ]], dtype=float32),
 array([ 1.2200994 , -0.34654886,  0.5853467 ,  0.02294951, -0.4124335 ,
        -1.9242957 , -0.61362666], dtype=float32),
 array([[-1.0967549 , -0.91878974, -0.33977914, -3.2540274 , -0.28609747,
          0.32850334],
        [-1.329424  , -1.3913674 , -0.5879582 , -1.1073284 , -0.06342156,
          0.36710948],
        [ 0.19983341,  0.27650437, -0.29943925,  0.36451414, -0.5822177 ,
         -2.5042002 ],
        [ 0.5515937 ,  0.46177578, -0.5739764 ,  0.4287029 , -0.70267427,
         -4.3690777 ],
        [ 0.47045842, -0.43380505, -0.05732127, -0.28128424,  0.0488058 ,
         -0.72834426],
        [ 0.30601364,  0.4181038 , -0.05442235,  0.37899718, -0.65186   ,
         -0.2664855 ],
        [ 0.1995398 ,  0.25233245, -0.06094968,  0.13355607, -0.16511916,
         -0.9760894 ]], dtype=float32),
 array([ 0.72017527,  0.7216329 , -0.31944504,  0.61424637, -0.4642663 ,
         0.08282294], dtype=float32),
 array([[-0.20780405],
        [-0.25537932],
        [-0.1695681 ],
        [-0.2303269 ],
        [-0.01416992],
        [ 0.22989334]], dtype=float32),
 array([-0.7617152], dtype=float32)]

In [ ]:
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: