

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

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: tensorflow-gpu in /usr/local/lib/python3.8/dist-packages (2.11.0)
Requirement already satisfied: protobuf<3.20,>=3.9.2 in /usr/local/lib/python3.8/dist-packages (from tensorflow-gpu) (3.19.6)
Requirement already satisfied: termcolor>=1.1.0 in /usr/local/lib/python3.8/dist-packages (from tensorflow-gpu) (2.1.1)
Requirement already satisfied: absl-py>=1.0.0 in /usr/local/lib/python3.8/dist-packages (from tensorflow-gpu) (1.3.0)
Requirement already satisfied: libclang>=13.0.0 in /usr/local/lib/python3.8/dist-packages (from tensorflow-gpu) (14.0.6)
Requirement already satisfied: numpy>=1.20 in /usr/local/lib/python3.8/dist-packages (from tensorflow-gpu) (1.21.6)
Requirement already satisfied: gast<=0.4.0,>=0.2.1 in /usr/local/lib/python3.8/dist-packages (from tensorflow-gpu) (0.4.0)
Requirement already satisfied: setuptools in /usr/local/lib/python3.8/dist-packages (from tensorflow-gpu) (57.4.0)
Requirement already satisfied: google-pasta>=0.1.1 in /usr/local/lib/python3.8/dist-packages (from tensorflow-gpu) (0.2.0)
Requirement already satisfied: keras<2.12,>=2.11.0 in /usr/local/lib/python3.8/dist-packages (from tensorflow-gpu) (2.11.0)
Requirement already satisfied: six>=1.12.0 in /usr/local/lib/python3.8/dist-packages (from tensorflow-gpu) (1.15.0)
Requirement already satisfied: packaging in /usr/local/lib/python3.8/dist-packages (from tensorflow-gpu) (21.3)
Requirement already satisfied: flatbuffers>=2.0 in /usr/local/lib/python3.8/dist-packages (from tensorflow-gpu) (23.1.4)
Requirement already satisfied: tensorboard<2.12,>=2.11 in /usr/local/lib/python3.8/dist-packages (from tensorflow-gpu) (2.11.0)
Requirement already satisfied: typing-extensions>=3.6.6 in /usr/local/lib/python3.8/dist-packages (from tensorflow-gpu) (4.4.0)
Requirement already satisfied: opt-einsum>=2.3.2 in /usr/local/lib/python3.8/dist-packages (from tensorflow-gpu) (3.3.0)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in /usr/local/lib/python3.8/dist-packages (from tensorflow-gpu) (1.51.1)
Requirement already satisfied: h5py>=2.9.0 in /usr/local/lib/python3.8/dist-packages (from tensorflow-gpu) (3.1.0)
Requirement already satisfied: tensorflow-estimator<2.12,>=2.11.0 in /usr/local/lib/python3.8/dist-packages (from tensorflow-gpu) (2.11.0)
Requirement already satisfied: astunparse>=1.6.0 in /usr/local/lib/python3.8/dist-packages (from tensorflow-gpu) (1.6.3)
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in /usr/local/lib/python3.8/dist-packages (from tensorflow-gpu) (0.24.0)
Requirement already satisfied: wrapt>=1.11.0 in /usr/local/lib/python3.8/dist-packages (from tensorflow-gpu) (1.14.1)
Requirement already satisfied: wheel<1.0,>=0.23.0 in /usr/local/lib/python3.8/dist-packages (from astunparse>=1.6.0->tensorflow-gpu) (0.38.1)
Requirement already satisfied: google-auth<3,>=1.6.3 in /usr/local/lib/python3.8/dist-packages (from tensorboard<2.12,>=2.11->tensorflow-gpu) (2.22.0)
Requirement already satisfied: tensorboard-data-server<0.7.0,>=0.6.0 in /usr/local/lib/python3.8/dist-packages (from tensorboard<2.12,>=2.11->tensorflow-gpu) (0.6.0)
Requirement already satisfied: tensorboard-plugin-wit>=1.6.0 in /usr/local/lib/python3.8/dist-packages (from tensorboard<2.12,>=2.11->tensorflow-gpu) (1.8.0)
Requirement already satisfied: requests<3,>=2.21.0 in /usr/local/lib/python3.8/dist-packages (from tensorboard<2.12,>=2.11->tensorflow-gpu) (2.28.1)
Requirement already satisfied: google-auth-oauthlib<0.5,>=0.4.1 in /usr/local/lib/python3.8/dist-packages (from tensorboard<2.12,>=2.11->tensorflow-gpu) (0.4.6)
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.8/dist-packages (from tensorboard<2.12,>=2.11->tensorflow-gpu) (3.4.3)
Requirement already satisfied: werkzeug>=1.0.1 in /usr/local/lib/python3.8/dist-packages (from tensorboard<2.12,>=2.11->tensorflow-gpu) (2.2.3)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /usr/local/lib/python3.8/dist-packages (from packaging->tensorflow-gpu) (3.0.9)
Requirement already satisfied: pyasn1-modules>=0.2.1 in /usr/local/lib/python3.8/dist-packages (from google-auth<3,>=1.6.3->tensorboard<2.12,>=2.11->tensorflow-gpu) (0.3.1)
Requirement already satisfied: rsa<5,>=3.1.4 in /usr/local/lib/python3.8/dist-packages (from google-auth<3,>=1.6.3->tensorboard<2.12,>=2.11->tensorflow-gpu) (4.9)
Requirement already satisfied: cachetools<6.0,>=2.0.0 in /usr/local/lib/python3.8/dist-packages (from google-auth<3,>=1.6.3->tensorboard<2.12,>=2.11->tensorflow-gpu) (5.2.1)
Requirement already satisfied: requests-oauthlib>=0.7.0 in /usr/local/lib/python3.8/dist-packages (from google-auth-oauthlib<0.5,>=0.4.1->tensorboard<2.12,>=2.11->tensorflow-gpu) (1.3.1)
Requirement already satisfied: importlib-metadata>=4.4 in /usr/local/lib/python3.8/dist-packages (from markdown>=2.6.8->tensorboard<2.12,>=2.11->tensorflow-gpu) (6.7.0)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.8/dist-packages (from requests<3,>=2.21.0->tensorboard<2.12,>=2.11->tensorflow-gpu) (3.4)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /usr/local/lib/python3.8/dist-packages (from requests<3,>=2.21.0->tensorboard<2.12,>=2.11->tensorflow-gpu) (1.26.15)
Requirement already satisfied: chardet<5,>=3.0.2 in /usr/local/lib/python3.8/dist-packages (from requests<3,>=2.21.0->tensorboard<2.12,>=2.11->tensorflow-gpu) (5.1.0)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.8/dist-packages (from requests<3,>=2.21.0->tensorboard<2.12,>=2.11->tensorflow-gpu) (2023.7.22)
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.8/dist-packages (from importlib-metadata>=4.4->markdown>=2.6.8->tensorflow-gpu) (3.17.0)
Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in /usr/local/lib/python3.8/dist-packages (from pyasn1-modules>=0.2.1->google-auth<3,>=1.6.3->tensorboard<2.12,>=2.11->tensorflow-gpu) (0.4.8)
Requirement already satisfied: oauthlib>=3.0.0 in /usr/local/lib/python3.8/dist-packages (from requests-oauthlib>=0.7.0->google-auth-oauthlib<0.5,>=0.4.1->tensorboard<2.12,>=2.11->tensorflow-gpu) (3.2.2)
```

```
#importing tensorflow
import tensorflow as tf
```

```
#checking tensorflow version
print(tf.__version__)
```

```
2.11.0
```

```
# import necessary libraries
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
```

```
# reading the dataset
```

```
data = pd.read_csv('/content/Churn_Modelling.csv')
```

```
#checking first five rows
data.head()
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary
0	1	15634602	Hargrave	619	France	Female	42	2	0.00	1	1	1	
1	2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	1	
2	3	15619304	Onio	502	France	Female	42	8	159660.80	3	1	0	
3	4	15701354	Boni	699	France	Female	39	1	0.00	2	0	0	

```
#checking null values if any
data.isnull().sum()

RowNumber      0
CustomerId      0
Surname         0
CreditScore    0
Geography       0
Gender          0
Age             0
Tenure          0
Balance         0
NumOfProducts  0
HasCrCard       0
IsActiveMember  0
EstimatedSalary 0
Exited          0
dtype: int64

#observation: No null values found here

#checking duplicate values
data.duplicated().sum()

0

#observation: No duplicate values found here

#checking columns
data.columns

Index(['RowNumber', 'CustomerId', 'Surname', 'CreditScore', 'Geography',
      'Gender', 'Age', 'Tenure', 'Balance', 'NumOfProducts', 'HasCrCard',
      'IsActiveMember', 'EstimatedSalary', 'Exited'],
      dtype='object')

#dropping unnecessary columns
data = data.drop(['RowNumber','CustomerId','Surname'],axis=1)

from pandas.io.pytables import DataCol
#splitting the data
x = data.iloc[:, :-1]
y = data.iloc[:, -1]

x.head()
```

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

geography = pd.get_dummies(x['Geography'],drop_first=True)
```

geography

	Germany	Spain
0	0	0
1	0	1
2	0	0
3	0	0
4	0	1
...
9995	0	0
9996	0	0
9997	0	0
9998	1	0
9999	0	0

10000 rows × 2 columns

```
gender = pd.get_dummies(x['Gender'],drop_first=True)
```

gender

	Male
0	0
1	0
2	0
3	0
4	0
...	...
9995	1
9996	1
9997	0
9998	1
9999	0

10000 rows × 1 columns

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

x.head()
```

```
#concatinating the geography and gender

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

x.head()
```

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

```
#splitting the dataset into x_train, y_train
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)
```

```
x_train.shape

(8000, 11)
```

```
#performing feature scaling
```

```
from sklearn.preprocessing import StandardScaler
```

```
sc = StandardScaler()
```

```
x_train = sc.fit_transform(x_train)
```

```
x_test = sc.transform(x_test)
```

```
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([[ -0.55204276, -0.36890377,  1.04473698, ...,  1.75486502,
        -0.57369368, -1.09168714],
       [-1.31490297,  0.10961719, -1.031415 , ..., -0.5698444 ,
        -0.57369368, -1.09168714],
       [ 0.57162971,  0.30102557,  1.04473698, ..., -0.5698444 ,
         1.74309049, -1.09168714],
       ...,
       [-0.74791227, -0.27319958, -1.37744033, ..., -0.5698444 ,
         1.74309049,  0.91601335],
       [-0.00566991, -0.46460796, -0.33936434, ...,  1.75486502,
        -0.57369368,  0.91601335],
       [-0.79945688, -0.84742473,  1.04473698, ...,  1.75486502,
        -0.57369368,  0.91601335]])
```

```

x_train.shape

(8000, 11)

#part2 = now let's create the ANN

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import ReLU, PReLU, LeakyReLU, ELU
from tensorflow.keras.layers import Dropout

# let's initialize the ANN

classifier = Sequential()

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

#adding the first hidden layer
classifier.add(Dense(units=7,activation='relu'))
#classifier.add(Dropout(0.2)), we can also check with this

#adding the 2nd hidden layer
classifier.add(Dense(units=5,activation='relu'))
#classifier.add(Dropout(0.2)), we can also check with this

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

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

#Observations: in adam optimizer we will get the learning rate automatically as 0.01

#creating our own learning rate
import tensorflow
tensorflow.keras.optimizers.Adam(learning_rate=0.01)

-----
NameError                                Traceback (most recent call last)
<ipython-input-169-d162903c8642> in <module>
      1 #creating our own learning rate
      2 import tensorflow
----> 3 tensorflow.keras.optimizers.Adam(learning_rate=0.01)

NameError: name 'learning_rate' is not defined

```

SEARCH STACK OVERFLOW

```

model_history = classifier.fit(x_train,y_train,validation_split=0.33,batch_size=10,epochs=100)

```

```

536/536 [=====] - 2s 3ms/step - loss: 0.3235 - accuracy: 0.8643 - val_loss: 0.3662 - val_accuracy: 0.8558
Epoch 83/100
536/536 [=====] - 2s 4ms/step - loss: 0.3238 - accuracy: 0.8675 - val_loss: 0.3627 - val_accuracy: 0.8591
Epoch 84/100
536/536 [=====] - 2s 3ms/step - loss: 0.3229 - accuracy: 0.8662 - val_loss: 0.3633 - val_accuracy: 0.8546
Epoch 85/100
536/536 [=====] - 2s 4ms/step - loss: 0.3229 - accuracy: 0.8651 - val_loss: 0.3629 - val_accuracy: 0.8535
Epoch 86/100
536/536 [=====] - 2s 4ms/step - loss: 0.3233 - accuracy: 0.8651 - val_loss: 0.3615 - val_accuracy: 0.8599
Epoch 87/100
536/536 [=====] - 2s 4ms/step - loss: 0.3238 - accuracy: 0.8679 - val_loss: 0.3631 - val_accuracy: 0.8561
Epoch 88/100
536/536 [=====] - 2s 3ms/step - loss: 0.3247 - accuracy: 0.8658 - val_loss: 0.3652 - val_accuracy: 0.8550
Epoch 89/100
536/536 [=====] - 2s 3ms/step - loss: 0.3233 - accuracy: 0.8649 - val_loss: 0.3635 - val_accuracy: 0.8561
Epoch 90/100
536/536 [=====] - 2s 4ms/step - loss: 0.3213 - accuracy: 0.8698 - val_loss: 0.3791 - val_accuracy: 0.8383
Epoch 91/100
536/536 [=====] - 2s 3ms/step - loss: 0.3229 - accuracy: 0.8670 - val_loss: 0.3651 - val_accuracy: 0.8557
Epoch 92/100
536/536 [=====] - 2s 3ms/step - loss: 0.3226 - accuracy: 0.8683 - val_loss: 0.3652 - val_accuracy: 0.8565
Epoch 93/100
536/536 [=====] - 2s 3ms/step - loss: 0.3233 - accuracy: 0.8640 - val_loss: 0.3657 - val_accuracy: 0.8516
Epoch 94/100
536/536 [=====] - 2s 3ms/step - loss: 0.3220 - accuracy: 0.8653 - val_loss: 0.3654 - val_accuracy: 0.8542
Epoch 95/100
536/536 [=====] - 2s 4ms/step - loss: 0.3226 - accuracy: 0.8655 - val_loss: 0.3660 - val_accuracy: 0.8523
Epoch 96/100
536/536 [=====] - 3s 5ms/step - loss: 0.3217 - accuracy: 0.8642 - val_loss: 0.3690 - val_accuracy: 0.8565
Epoch 97/100
536/536 [=====] - 2s 4ms/step - loss: 0.3217 - accuracy: 0.8673 - val_loss: 0.3735 - val_accuracy: 0.8527
Epoch 98/100
536/536 [=====] - 5s 9ms/step - loss: 0.3228 - accuracy: 0.8668 - val_loss: 0.3738 - val_accuracy: 0.8527
Epoch 99/100
536/536 [=====] - 3s 5ms/step - loss: 0.3213 - accuracy: 0.8651 - val_loss: 0.3646 - val_accuracy: 0.8550
Epoch 100/100
536/536 [=====] - 2s 5ms/step - loss: 0.3205 - accuracy: 0.8668 - val_loss: 0.3731 - val_accuracy: 0.8429

```

```

#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,batch_size=10,epochs=100,callbacks = early_stopping)
```

```

Epoch 1/100
536/536 [=====] - 2s 4ms/step - loss: 0.3216 - accuracy: 0.8660 - val_loss: 0.3692 - val_accuracy: 0.8557
Epoch 2/100
536/536 [=====] - 2s 4ms/step - loss: 0.3211 - accuracy: 0.8653 - val_loss: 0.3687 - val_accuracy: 0.8542
Epoch 3/100
536/536 [=====] - 2s 3ms/step - loss: 0.3212 - accuracy: 0.8671 - val_loss: 0.3670 - val_accuracy: 0.8538
Epoch 4/100
536/536 [=====] - 2s 4ms/step - loss: 0.3212 - accuracy: 0.8668 - val_loss: 0.3708 - val_accuracy: 0.8482
Epoch 5/100
536/536 [=====] - 2s 4ms/step - loss: 0.3212 - accuracy: 0.8683 - val_loss: 0.3668 - val_accuracy: 0.8554
Epoch 6/100
536/536 [=====] - 2s 4ms/step - loss: 0.3205 - accuracy: 0.8679 - val_loss: 0.3643 - val_accuracy: 0.8550
Epoch 7/100
536/536 [=====] - 2s 4ms/step - loss: 0.3223 - accuracy: 0.8684 - val_loss: 0.3664 - val_accuracy: 0.8516
Epoch 8/100
536/536 [=====] - 2s 3ms/step - loss: 0.3207 - accuracy: 0.8681 - val_loss: 0.3677 - val_accuracy: 0.8523
Epoch 9/100
536/536 [=====] - 2s 3ms/step - loss: 0.3206 - accuracy: 0.8671 - val_loss: 0.3720 - val_accuracy: 0.8497
Epoch 10/100
536/536 [=====] - 2s 3ms/step - loss: 0.3216 - accuracy: 0.8675 - val_loss: 0.3690 - val_accuracy: 0.8573
Epoch 11/100
536/536 [=====] - 2s 3ms/step - loss: 0.3194 - accuracy: 0.8664 - val_loss: 0.3656 - val_accuracy: 0.8516
Epoch 12/100
536/536 [=====] - 2s 3ms/step - loss: 0.3188 - accuracy: 0.8662 - val_loss: 0.3661 - val_accuracy: 0.8550
Epoch 13/100
536/536 [=====] - 2s 4ms/step - loss: 0.3191 - accuracy: 0.8673 - val_loss: 0.3682 - val_accuracy: 0.8523
Epoch 14/100
536/536 [=====] - 2s 4ms/step - loss: 0.3194 - accuracy: 0.8690 - val_loss: 0.3711 - val_accuracy: 0.8538

```

```

Epoch 15/100
536/536 [=====] - 2s 4ms/step - loss: 0.3213 - accuracy: 0.8673 - val_loss: 0.3723 - val_accuracy: 0.8542
Epoch 16/100
536/536 [=====] - 2s 3ms/step - loss: 0.3186 - accuracy: 0.8681 - val_loss: 0.3677 - val_accuracy: 0.8538
Epoch 17/100
536/536 [=====] - 2s 3ms/step - loss: 0.3179 - accuracy: 0.8686 - val_loss: 0.3701 - val_accuracy: 0.8504
Epoch 18/100
536/536 [=====] - 2s 4ms/step - loss: 0.3201 - accuracy: 0.8655 - val_loss: 0.3666 - val_accuracy: 0.8542
Epoch 19/100
536/536 [=====] - 2s 4ms/step - loss: 0.3184 - accuracy: 0.8692 - val_loss: 0.3665 - val_accuracy: 0.8554
Epoch 20/100
536/536 [=====] - 2s 3ms/step - loss: 0.3186 - accuracy: 0.8664 - val_loss: 0.3708 - val_accuracy: 0.8595
Epoch 21/100
536/536 [=====] - 2s 4ms/step - loss: 0.3187 - accuracy: 0.8671 - val_loss: 0.3657 - val_accuracy: 0.8546
Epoch 22/100
536/536 [=====] - 2s 3ms/step - loss: 0.3179 - accuracy: 0.8711 - val_loss: 0.3755 - val_accuracy: 0.8535
Epoch 23/100
536/536 [=====] - 2s 3ms/step - loss: 0.3188 - accuracy: 0.8673 - val_loss: 0.3728 - val_accuracy: 0.8455
Epoch 24/100
536/536 [=====] - 2s 4ms/step - loss: 0.3191 - accuracy: 0.8699 - val_loss: 0.3741 - val_accuracy: 0.8482
Epoch 25/100
536/536 [=====] - 2s 4ms/step - loss: 0.3182 - accuracy: 0.8694 - val_loss: 0.3724 - val_accuracy: 0.8474
Epoch 26/100
536/536 [=====] - 2s 3ms/step - loss: 0.3183 - accuracy: 0.8716 - val_loss: 0.3660 - val_accuracy: 0.8554
Epoch 26: early stopping

```

#observation: when we apply early stopping, the models has stopped in 26th epochs where we got the high accuracy continuously

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

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

```
#summerize history of 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()

```

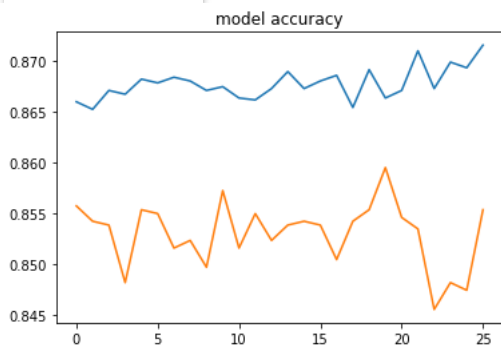
```

-----
AttributeError                                Traceback (most recent call last)
<ipython-input-178-6c93442be282> in <module>
      4 plt.plot(model_history.history['val_accuracy'])
      5 plt.title('model accuracy')
----> 6 plt.ylabel('accuracy')
      7 plt.xlabel('epoch')
      8 plt.legend(['train', 'test'], loc='upper left')

```

```
AttributeError: module 'matplotlib.pyplot' has no attribute 'ylabel'
```

SEARCH STACK OVERFLOW



```
#summerize history of 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()
```

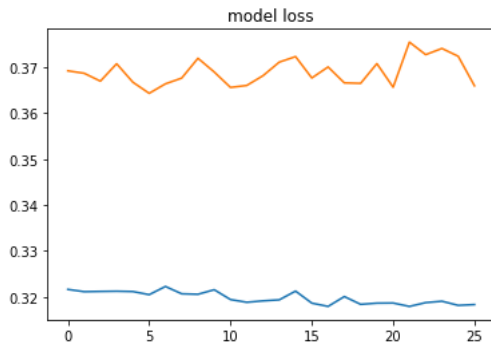
```

AttributeError                                Traceback (most recent call last)
<ipython-input-179-73e2301a56ad> in <module>
      4 plt.plot(model_history.history['val_loss'])
      5 plt.title('model loss')
----> 6 plt.ylabel('loss')
      7 plt.xlabel('epoch')
      8 plt.legend(['train','test'],loc='upper left')

```

AttributeError: module 'matplotlib.pyplot' has no attribute 'ylabel'

SEARCH STACK OVERFLOW



```
#making the predictions validating the model
#predicting the test set results
```

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

```
63/63 [=====] - 0s 3ms/step
```

```
y_pred = (y_pred >= 0.5)
```

```
#make the confusion matrix
from sklearn.metrics import confusion_matrix
```

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

```
cm
```

```
array([[1499,  96],
       [ 191, 214]])
```

```
from sklearn.metrics import accuracy_score
```

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

```
score
```

```
0.8565
```

```
# getting the weights
```

```
classifier.get_weights()
```



```
-0.34245377, 0.20330705, 0.34210020, 0.00017005, 0.03038503,
0.0695022 ], dtype=float32),
array([[ 0.05446914, -0.0271513 , -0.49611044, -0.1992463 , 1.0337577 ,
-0.11983801, 0.08127924],
[ -0.04218744, 0.6845132 , 0.5409455 , -0.11985739, -0.11944453,
0.43122166, -0.05408784],
[ 0.24662411, 0.25438663, -0.16986974, -0.60736585, 0.263576 ,
0.27018544, -1.0614408 ],
[ -0.6984371 , 0.05909414, -0.40886375, -0.07667132, -1.1021444 ,
0.65153563, 0.20674604],
[ 0.49887642, 0.66835535, 0.45850047, 0.5757199 , 0.26261824,
0.54127806, -0.05531849],
[ 0.35465476, 0.29024136, -0.54554623, 0.55335283, -0.12601252,
0.6390836 , 0.15706158],
[ 0.4035916 , -0.38544887, -1.0651339 , 0.5149539 , 0.06408582,
-0.15341984, 0.29750165],
[ -0.6021392 , 0.07206472, 0.6419946 , 0.18929596, 0.38350916,
-0.3143208 , -0.19367662],
[ 0.1587106 , -0.11467145, -0.8828561 , 0.15993945, 0.11302575,
-0.03498594, 0.4827117 ],
[ -0.1820494 , -0.2571856 , -0.31944674, -1.4793364 , -0.01877056,
0.24992792, 0.5423601 ],
[ 0.36178312, 0.20521167, -0.31077975, 0.02699341, 0.5932222 ,
0.25637475, 0.6116501 ]], dtype=float32),
array([ 0.55158865, 0.34859917, 0.23198739, 0.0491035 , -0.0942897 ,
-0.63105196, 0.08686335], dtype=float32),
array([[ 0.88825965, 0.24235576, 0.2726831 , -0.1001079 , 0.03549778],
[ 0.39849687, -0.36546496, 0.477124 , 0.01393195, 0.5723573 ],
[ 0.2154519 , 0.83851314, 0.1967897 , -2.218216 , -0.76328075],
[ -0.23566042, 0.5106066 , -0.5420726 , 0.5148286 , -0.44234285],
[ 0.26049614, -0.46540615, 0.22289342, 0.5889023 , -0.03098134],
[ -0.28139395, 0.6182708 , -0.4890441 , 0.42980397, -0.31701675],
[ -0.13721392, -0.16644026, 0.46767297, 0.53952897, 0.63457507]],
dtype=float32),
array([ 0.40654385, -0.086804 , 0.37736547, -0.56656563, 0.44847205],
dtype=float32),
array([[ 0.4908445 , -0.47190037, 0.6066686 , 0.07257682, 0.13227247],
[ -0.60433793, -0.76282346, -0.7777411 , 0.47370833, -0.6816994 ],
[ 0.03423654, -0.46542445, 0.8120542 , 0.13453387, -0.6313754 ],
[ -0.33680356, -0.11890525, -0.18988997, -1.4624531 , -0.03748875],
[ 0.79268265, -0.12498891, 0.6637929 , 0.8535326 , 0.18120623]],
dtype=float32),
array([ 0.37637845, 0. , 0.33879015, 0.21803184, -0.01149005],
dtype=float32),
array([[-1.2212205 ],
[ 0.29727983],
[ -0.7134965 ],
[ -1.2959719 ],
[ 0.9470253 ]], dtype=float32),
array([2.4014473], dtype=float32)]
```