```
# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd



from google.colab import files


uploaded = files.upload()
```

Choose Files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.
Saving heart.csv to heart.csv

```
import io
dataset = pd.read_csv(io.BytesIO(uploaded['heart.csv']))


# Importing the dataset
#dataset = pd.read_csv('heart.csv')
X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, 13].values


# Splitting the dataset into the 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 =


# Feature Scaling
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)



# Part 2 - Now let's make the ANN!

# Importing the Keras libraries and packages
import keras
from keras.models import Sequential
from keras.layers import Dense


# Initialising the ANN
classifier = Sequential()


# Adding the input layer and the first hidden layer
classifier.add(Dense(11,kernel_initializer = 'uniform', activation = 'relu', input_dim = 1


# Adding the second hidden layer
```

```
classifier.add(Dense(11,kernel_initializer = 'uniform', activation = 'relu'))

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



# Compiling the ANN
classifier.compile(optimizer = 'adam', loss = 'binary_crossentropy', metrics = ['accuracy'


# Fitting the ANN to the Training set
classifier.fit(X_train, y_train, batch_size = 10, epochs = 500)
```

```
Epoch 1/500
25/25 [==============================] - 0s 2ms/step - loss: 0.0174 - accuracy: 0.
Epoch 2/500
25/25 [==============================] - 0s 2ms/step - loss: 0.0174 - accuracy: 0.
Epoch 3/500
25/25 [==============================] - 0s 2ms/step - loss: 0.0174 - accuracy: 0.
Epoch 4/500
25/25 [==============================] - 0s 2ms/step - loss: 0.0174 - accuracy: 0.
Epoch 5/500
25/25 [==============================] - 0s 2ms/step - loss: 0.0174 - accuracy: 0.
Epoch 6/500
25/25 [==============================] - 0s 2ms/step - loss: 0.0174 - accuracy: 0.
Epoch 7/500
25/25 [==============================] - 0s 2ms/step - loss: 0.0174 - accuracy: 0.
Epoch 8/500
25/25 [==============================] - 0s 2ms/step - loss: 0.0174 - accuracy: 0.
Epoch 9/500
25/25 [==============================] - 0s 2ms/step - loss: 0.0174 - accuracy: 0.
Epoch 10/500
25/25 [==============================] - 0s 2ms/step - loss: 0.0174 - accuracy: 0.
Epoch 11/500
25/25 [==============================] - 0s 2ms/step - loss: 0.0174 - accuracy: 0.
Epoch 12/500
25/25 [==============================] - 0s 2ms/step - loss: 0.0174 - accuracy: 0.
Epoch 13/500
25/25 [==============================] - 0s 2ms/step - loss: 0.0174 - accuracy: 0.
Epoch 14/500
25/25 [==============================] - 0s 2ms/step - loss: 0.0174 - accuracy: 0.
Epoch 15/500
25/25 [==============================] - 0s 2ms/step - loss: 0.0174 - accuracy: 0.
Epoch 16/500
25/25 [==============================] - 0s 2ms/step - loss: 0.0174 - accuracy: 0.
Epoch 17/500
25/25 [==============================] - 0s 2ms/step - loss: 0.0174 - accuracy: 0.
Epoch 18/500
25/25 [==============================] - 0s 2ms/step - loss: 0.0174 - accuracy: 0.
Epoch 19/500
25/25 [==============================] - 0s 2ms/step - loss: 0.0174 - accuracy: 0.
Epoch 20/500
25/25 [==============================] - 0s 2ms/step - loss: 0.0174 - accuracy: 0.
Epoch 21/500
25/25 [==============================] - 0s 2ms/step - loss: 0.0174 - accuracy: 0.
Epoch 22/500
25/25 [==============================] - 0s 2ms/step - loss: 0.0174 - accuracy: 0.
Epoch 23/500
```

```
25/25 [==============================] - 0s 2ms/step - loss: 0.0174 - accuracy: 0.
Epoch 24/500
25/25 [==============================] - 0s 2ms/step - loss: 0.0174 - accuracy: 0.
Epoch 25/500
25/25 [==============================] - 0s 2ms/step - loss: 0.0174 - accuracy: 0.
Epoch 26/500
25/25 [==============================] - 0s 2ms/step - loss: 0.0174 - accuracy: 0.
Epoch 27/500
25/25 [==============================] - 0s 2ms/step - loss: 0.0174 - accuracy: 0.
Epoch 28/500
25/25 [==============================] - 0s 2ms/step - loss: 0.0174 - accuracy: 0.
Epoch 29/500
```

```
# 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)


# Making the Confusion Matrix
from sklearn.metrics import confusion_matrix
from sklearn.metrics import confusion_matrix, accuracy_score, classification_report
cm = confusion_matrix(y_test, y_pred)
print(cm)
#print(accuracy_score(X_test, y_pred))
print(classification_report(y_test, y_pred))
```

```
[[24  3]
 [ 8 26]]
              precision    recall  f1-score   support

           0       0.75      0.89      0.81        27
           1       0.90      0.76      0.83        34

    accuracy                           0.82        61
   macro avg       0.82      0.83      0.82        61
weighted avg       0.83      0.82      0.82        61
```