```
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 (1).csv
```

```
for fn in uploaded.keys():
  print('User uploaded file "{name}" with {length} bytes'.format(name=fn,length=len(uploaded[
```

```
User uploaded file "heart.csv" with 11328 bytes
```

```
import numpy as np
import pandas as pd
```

```
dataset = pd.read_csv('/content/heart.csv')
dataset.head()
```

| | age | sex | cp | trestbps | chol | fbs | restecg | thalach | exang | oldpeak | slope | ca | thal |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 63 | 1 | 3 | 145 | 233 | 1 | 0 | 150 | 0 | 2.3 | 0 | 0 | 1 |
| **1** | 37 | 1 | 2 | 130 | 250 | 0 | 1 | 187 | 0 | 3.5 | 0 | 0 | 2 |
| **2** | 41 | 0 | 1 | 130 | 204 | 0 | 0 | 172 | 0 | 1.4 | 2 | 0 | 2 |
| **3** | 56 | 1 | 1 | 120 | 236 | 0 | 1 | 178 | 0 | 0.8 | 2 | 0 | 2 |
| **4** | 57 | 0 | 0 | 120 | 354 | 0 | 1 | 163 | 1 | 0.6 | 2 | 0 | 2 |

```
X = dataset.iloc[:,:-1].values
Y = dataset.iloc[:,-1].values
```

```
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)
```

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

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

```
classifier = Sequential()

classifier.add(Dense(units=6,kernel_initializer='uniform',activation='relu',input_dim=13))
classifier.add(Dense(units=6,kernel_initializer='uniform',activation='relu'))
classifier.add(Dense(units=1,kernel_initializer='uniform',activation='sigmoid'))

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

classifier.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_3 (Dense)              (None, 6)                 84
_________________________________________________________________
dense_4 (Dense)              (None, 6)                 42
_________________________________________________________________
dense_5 (Dense)              (None, 1)                 7
=================================================================
Total params: 133
Trainable params: 133
Non-trainable params: 0
_________________________________________________________________
```

```
classifier.fit(X_train,Y_train,batch_size=10,epochs=50)
```

```
Epoch 1/50
25/25 [==============================] - 1s 2ms/step - loss: 0.3200 - accuracy: 0.863
Epoch 2/50
25/25 [==============================] - 0s 2ms/step - loss: 0.3176 - accuracy: 0.876
Epoch 3/50
25/25 [==============================] - 0s 2ms/step - loss: 0.3172 - accuracy: 0.876
Epoch 4/50
25/25 [==============================] - 0s 2ms/step - loss: 0.3164 - accuracy: 0.880
Epoch 5/50
25/25 [==============================] - 0s 2ms/step - loss: 0.3152 - accuracy: 0.876
Epoch 6/50
25/25 [==============================] - 0s 2ms/step - loss: 0.3149 - accuracy: 0.876
Epoch 7/50
25/25 [==============================] - 0s 2ms/step - loss: 0.3143 - accuracy: 0.884
Epoch 8/50
25/25 [==============================] - 0s 2ms/step - loss: 0.3129 - accuracy: 0.884
Epoch 9/50
25/25 [==============================] - 0s 2ms/step - loss: 0.3124 - accuracy: 0.880
Epoch 10/50
25/25 [==============================] - 0s 2ms/step - loss: 0.3110 - accuracy: 0.884
Epoch 11/50
25/25 [==============================] - 0s 2ms/step - loss: 0.3106 - accuracy: 0.884
Epoch 12/50
25/25 [==============================] - 0s 2ms/step - loss: 0.3102 - accuracy: 0.884
Epoch 13/50
25/25 [==============================] - 0s 2ms/step - loss: 0.3094 - accuracy: 0.884
Epoch 14/50
25/25 [==============================] - 0s 2ms/step - loss: 0.3081 - accuracy: 0.884
```

```
Epoch 15/50
25/25 [==============================] - 0s 2ms/step - loss: 0.3077 - accuracy: 0.884
Epoch 16/50
25/25 [==============================] - 0s 2ms/step - loss: 0.3067 - accuracy: 0.884
Epoch 17/50
25/25 [==============================] - 0s 2ms/step - loss: 0.3057 - accuracy: 0.884
Epoch 18/50
25/25 [==============================] - 0s 2ms/step - loss: 0.3056 - accuracy: 0.884
Epoch 19/50
25/25 [==============================] - 0s 2ms/step - loss: 0.3050 - accuracy: 0.884
Epoch 20/50
25/25 [==============================] - 0s 2ms/step - loss: 0.3035 - accuracy: 0.884
Epoch 21/50
25/25 [==============================] - 0s 2ms/step - loss: 0.3030 - accuracy: 0.884
Epoch 22/50
25/25 [==============================] - 0s 2ms/step - loss: 0.3027 - accuracy: 0.884
Epoch 23/50
25/25 [==============================] - 0s 2ms/step - loss: 0.3020 - accuracy: 0.884
Epoch 24/50
25/25 [==============================] - 0s 2ms/step - loss: 0.3020 - accuracy: 0.888
Epoch 25/50
25/25 [==============================] - 0s 2ms/step - loss: 0.3008 - accuracy: 0.884
Epoch 26/50
25/25 [==============================] - 0s 2ms/step - loss: 0.3001 - accuracy: 0.884
Epoch 27/50
25/25 [==============================] - 0s 2ms/step - loss: 0.2988 - accuracy: 0.884
Epoch 28/50
25/25 [==============================] - 0s 2ms/step - loss: 0.2981 - accuracy: 0.888
Epoch 29/50
25/25 [==============================] - 0s 2ms/step - loss: 0.2976 - accuracy: 0.888
Epoch 30/50
```

```
Y_pred = classifier.predict(X_test)
Y_pred = (Y_pred>0.5)

from sklearn.metrics import confusion_matrix
cm = confusion_matrix(Y_test,Y_pred)
```

```
print(cm)
```

```
[[24  3]
 [ 3 31]]
```

```
tn,fp,fn,tp = cm.ravel()
(tn,fp,fn,tp)
```

```
(24, 3, 3, 31)
```

```
acc = (((tp+tn)/(tn+fp+fn+tp))*100)
print(acc)
```

```
90.1639344262295
```

✓ 0s completed at 11:38