Kedarnath K Bhat bhatkedar17@gmail.com

```
from google.colab import files
uploaded=files.upload()
for fn in uploaded.keys():
  print('User uploaded file "{name}" with length {length} bytes'.format(name=fn, length=le
```

Choose Files heart.csv

- **heart.csv**(application/vnd.ms-excel) - 11328 bytes, last modified: 10/24/2021 - 100% done

```
Saving heart.csv to heart (1).csv
User uploaded file "heart.csv" with length 11328 bytes
```

```
import numpy as np
import pandas as pd
import keras
import matplotlib as plt
```

```
data=pd.read_csv("heart.csv")
```

```
data.head()
```

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

```
X = data.iloc[:,:-1].values
y = data.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 = 0.15, random_state =
```

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

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


classifier=Sequential()

classifier.add(Dense(units = 6, kernel_initializer = 'uniform', activation='sigmoid', inpu

classifier.add(Dense(units=6, kernel_initializer='uniform', activation='sigmoid'))

classifier.add(Dense(units=1, kernel_initializer='uniform', activation='sigmoid'))


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

classifier.fit(X_train, y_train, batch_size =15, epochs=100)#batch_size of 15 gives more a
```

```
Epoch 1/100
18/18 [==============================] - 0s 1ms/step - loss: 0.6934 - accuracy: 0
Epoch 2/100
18/18 [==============================] - 0s 1ms/step - loss: 0.6918 - accuracy: 0
Epoch 3/100
18/18 [==============================] - 0s 1ms/step - loss: 0.6912 - accuracy: 0
Epoch 4/100
18/18 [==============================] - 0s 1ms/step - loss: 0.6908 - accuracy: 0
Epoch 5/100
18/18 [==============================] - 0s 2ms/step - loss: 0.6901 - accuracy: 0
Epoch 6/100
18/18 [==============================] - 0s 2ms/step - loss: 0.6893 - accuracy: 0
Epoch 7/100
18/18 [==============================] - 0s 1ms/step - loss: 0.6885 - accuracy: 0
Epoch 8/100
18/18 [==============================] - 0s 1ms/step - loss: 0.6874 - accuracy: 0
Epoch 9/100
18/18 [==============================] - 0s 1ms/step - loss: 0.6863 - accuracy: 0
Epoch 10/100
18/18 [==============================] - 0s 1ms/step - loss: 0.6855 - accuracy: 0
Epoch 11/100
18/18 [==============================] - 0s 1ms/step - loss: 0.6843 - accuracy: 0
Epoch 12/100
18/18 [==============================] - 0s 1ms/step - loss: 0.6831 - accuracy: 0
Epoch 13/100
18/18 [==============================] - 0s 1ms/step - loss: 0.6815 - accuracy: 0
Epoch 14/100
18/18 [==============================] - 0s 1ms/step - loss: 0.6798 - accuracy: 0
Epoch 15/100
18/18 [==============================] - 0s 2ms/step - loss: 0.6782 - accuracy: 0
Epoch 16/100
18/18 [==============================] - 0s 1ms/step - loss: 0.6763 - accuracy: 0
Epoch 17/100
18/18 [==============================] - 0s 1ms/step - loss: 0.6748 - accuracy: 0
Epoch 18/100
18/18 [==============================] - 0s 1ms/step - loss: 0.6727 - accuracy: 0
Epoch 19/100
18/18 [==============================] - 0s 1ms/step - loss: 0.6702 - accuracy: 0
```

```
Epoch 20/100
18/18 [==============================] - 0s 1ms/step - loss: 0.6675 - accuracy: 0
Epoch 21/100
18/18 [==============================] - 0s 1ms/step - loss: 0.6642 - accuracy: 0
Epoch 22/100
18/18 [==============================] - 0s 1ms/step - loss: 0.6608 - accuracy: 0
Epoch 23/100
18/18 [==============================] - 0s 1ms/step - loss: 0.6568 - accuracy: 0
Epoch 24/100
18/18 [==============================] - 0s 1ms/step - loss: 0.6526 - accuracy: 0
Epoch 25/100
18/18 [==============================] - 0s 1ms/step - loss: 0.6472 - accuracy: 0
Epoch 26/100
18/18 [==============================] - 0s 1ms/step - loss: 0.6417 - accuracy: 0
Epoch 27/100
18/18 [==============================] - 0s 1ms/step - loss: 0.6357 - accuracy: 0
Epoch 28/100
18/18 [==============================] - 0s 1ms/step - loss: 0.6299 - accuracy: 0
Epoch 29/100
```

```
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)
```

```
[[19  4]
 [ 3 20]]
```

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

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

```
84.78260869565217
```

```
classifier.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_6 (Dense)              (None, 6)                 84
_________________________________________________________________
dense_7 (Dense)              (None, 6)                 42
_________________________________________________________________
dense_8 (Dense)              (None, 1)                 7
=================================================================
Total params: 133
Trainable params: 133
```

```
Non-trainable params: 0
```

✓ 0s completed at 2:12 PM