PROJECT : PREDICT HEART DIESEAS

Import Required Libraries

In [ ]:
```
import numpy as np
import pandas as pd
import tensorflow as tf
df=pd.read_csv('/content/heart.csv')
df
```

Out[ ]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3	0
1	53	1	0	140	203	1	0	155	1	3.1	0	0	3	0
2	70	1	0	145	174	0	1	125	1	2.6	0	0	3	0
3	61	1	0	148	203	0	1	161	0	0.0	2	1	3	0
4	62	0	0	138	294	1	1	106	0	1.9	1	3	2	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1020	59	1	1	140	221	0	1	164	1	0.0	2	0	2	1
1021	60	1	0	125	258	0	0	141	1	2.8	1	1	3	0
1022	47	1	0	110	275	0	0	118	1	1.0	1	1	2	0
1023	50	0	0	110	254	0	0	159	0	0.0	2	0	2	1
1024	54	1	0	120	188	0	1	113	0	1.4	1	1	3	0

1025 rows × 14 columns

In [ ]:
```
df.head()
```

Out[ ]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3	0
1	53	1	0	140	203	1	0	155	1	3.1	0	0	3	0
2	70	1	0	145	174	0	1	125	1	2.6	0	0	3	0
3	61	1	0	148	203	0	1	161	0	0.0	2	1	3	0
4	62	0	0	138	294	1	1	106	0	1.9	1	3	2	0

In [ ]:
```
df.tail()
```

Out[ ]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
1020	59	1	1	140	221	0	1	164	1	0.0	2	0	2	1
1021	60	1	0	125	258	0	0	141	1	2.8	1	1	3	0
1022	47	1	0	110	275	0	0	118	1	1.0	1	1	2	0
1023	50	0	0	110	254	0	0	159	0	0.0	2	0	2	1
1024	54	1	0	120	188	0	1	113	0	1.4	1	1	3	0

Checking missing values

In [ ]:
```
df.isna().sum()
```

Out[ ]:
```
age         0
sex         0
cp          0
trestbps    0
chol        0
fbs         0
restecg     0
thalach     0
exang       0
oldpeak     0
slope       0
ca          0
thal        0
target      0
dtype: int64
```

In [ ]:
```
df.dtypes
```

```
Out[ ]: age           int64
        sex           int64
        cp            int64
        trestbps      int64
        chol          int64
        fbs           int64
        restecg       int64
        thalach       int64
        exang         int64
        oldpeak     float64
        slope         int64
        ca            int64
        thal          int64
        target        int64
        dtype: object
```

Separate X and Y

```
In [ ]: x=df.iloc[:,:-1].values
        y=df.iloc[:,-1].values
```

Data into Traning and Testing

```
In [ ]: from sklearn.model_selection import train_test_split
        x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.30,random_state=42)
        x_train
```

```
Out[ ]: array([[59.,  1.,  1., ...,  2.,  0.,  2.],
               [58.,  1.,  0., ...,  1.,  3.,  3.],
               [44.,  0.,  2., ...,  1.,  1.,  2.],
               ...,
               [51.,  1.,  0., ...,  2.,  0.,  3.],
               [43.,  1.,  0., ...,  2.,  0.,  3.],
               [52.,  1.,  0., ...,  2.,  1.,  2.]])
```

```
In [ ]: x_test
        y_train
        y_test
```

```
Out[ ]: array([1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 1, 0, 1, 0,
               0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0,
               0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0,
               1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0,
               0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1,
               0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0,
               1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1,
               0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1,
               1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0,
               1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0,
               1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0,
               1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0,
               0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0,
               1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0])
```

Data Normalization using MinMaxScaler

```
In [ ]: from sklearn.preprocessing import MinMaxScaler
        scaler=MinMaxScaler()
        scaler.fit_transform(x_train)
        x_test=scaler.transform(x_test)
```

ANN

```
In [ ]: ann=tf.keras.models.Sequential()
```

First Hidden layer used RELU Activation function

```
In [ ]: ann.add(tf.keras.layers.Dense(6,activation='relu'))
```

Second Hidden layer used RELU Activation function

```
In [ ]: ann.add(tf.keras.layers.Dense(6,activation='relu'))
```

Output layer used Sigmoid Activation function

```
In [ ]: ann.add(tf.keras.layers.Dense(1,activation='sigmoid'))
```

Compiling ANN

```
In [ ]: ann.compile(optimizer='adam',loss="binary_crossentropy",metrics=['accuracy'])
```

```
In [ ]: ann.fit(x_train,y_train,batch_size=64,epochs=100)
```

```
Epoch 1/100
12/12 [==============================] - 1s 3ms/step - loss: 14.9127 - accuracy: 0.5258
Epoch 2/100
12/12 [==============================] - 0s 5ms/step - loss: 11.7842 - accuracy: 0.5258
Epoch 3/100
12/12 [==============================] - 0s 5ms/step - loss: 8.7426 - accuracy: 0.5258
Epoch 4/100
12/12 [==============================] - 0s 4ms/step - loss: 5.2108 - accuracy: 0.5328
Epoch 5/100
12/12 [==============================] - 0s 4ms/step - loss: 1.9649 - accuracy: 0.6053
Epoch 6/100
12/12 [==============================] - 0s 4ms/step - loss: 1.0755 - accuracy: 0.6444
Epoch 7/100
12/12 [==============================] - 0s 3ms/step - loss: 1.0337 - accuracy: 0.6722
Epoch 8/100
12/12 [==============================] - 0s 5ms/step - loss: 0.9143 - accuracy: 0.6709
Epoch 9/100
12/12 [==============================] - 0s 3ms/step - loss: 0.8780 - accuracy: 0.6834
Epoch 10/100
12/12 [==============================] - 0s 3ms/step - loss: 0.8388 - accuracy: 0.6932
Epoch 11/100
12/12 [==============================] - 0s 3ms/step - loss: 0.8089 - accuracy: 0.7001
Epoch 12/100
12/12 [==============================] - 0s 3ms/step - loss: 0.7869 - accuracy: 0.7029
Epoch 13/100
12/12 [==============================] - 0s 4ms/step - loss: 0.7670 - accuracy: 0.7280
Epoch 14/100
12/12 [==============================] - 0s 3ms/step - loss: 0.7520 - accuracy: 0.7280
Epoch 15/100
12/12 [==============================] - 0s 3ms/step - loss: 0.7351 - accuracy: 0.7211
Epoch 16/100
12/12 [==============================] - 0s 3ms/step - loss: 0.7200 - accuracy: 0.7280
Epoch 17/100
12/12 [==============================] - 0s 3ms/step - loss: 0.7098 - accuracy: 0.7336
Epoch 18/100
12/12 [==============================] - 0s 3ms/step - loss: 0.6991 - accuracy: 0.7266
Epoch 19/100
12/12 [==============================] - 0s 3ms/step - loss: 0.6901 - accuracy: 0.7252
Epoch 20/100
12/12 [==============================] - 0s 3ms/step - loss: 0.6812 - accuracy: 0.7252
Epoch 21/100
12/12 [==============================] - 0s 4ms/step - loss: 0.6731 - accuracy: 0.7238
Epoch 22/100
12/12 [==============================] - 0s 4ms/step - loss: 0.6669 - accuracy: 0.7197
Epoch 23/100
12/12 [==============================] - 0s 4ms/step - loss: 0.6577 - accuracy: 0.7225
Epoch 24/100
12/12 [==============================] - 0s 4ms/step - loss: 0.6501 - accuracy: 0.7225
Epoch 25/100
12/12 [==============================] - 0s 4ms/step - loss: 0.6432 - accuracy: 0.7211
Epoch 26/100
12/12 [==============================] - 0s 4ms/step - loss: 0.6403 - accuracy: 0.7252
Epoch 27/100
12/12 [==============================] - 0s 4ms/step - loss: 0.6370 - accuracy: 0.7169
Epoch 28/100
12/12 [==============================] - 0s 4ms/step - loss: 0.6276 - accuracy: 0.7183
Epoch 29/100
12/12 [==============================] - 0s 3ms/step - loss: 0.6224 - accuracy: 0.7252
Epoch 30/100
12/12 [==============================] - 0s 4ms/step - loss: 0.6156 - accuracy: 0.7266
Epoch 31/100
12/12 [==============================] - 0s 4ms/step - loss: 0.6130 - accuracy: 0.7266
Epoch 32/100
12/12 [==============================] - 0s 4ms/step - loss: 0.6072 - accuracy: 0.7238
Epoch 33/100
12/12 [==============================] - 0s 4ms/step - loss: 0.6029 - accuracy: 0.7238
Epoch 34/100
12/12 [==============================] - 0s 5ms/step - loss: 0.5982 - accuracy: 0.7266
Epoch 35/100
12/12 [==============================] - 0s 4ms/step - loss: 0.5922 - accuracy: 0.7211
Epoch 36/100
12/12 [==============================] - 0s 3ms/step - loss: 0.5882 - accuracy: 0.7280
Epoch 37/100
12/12 [==============================] - 0s 3ms/step - loss: 0.5853 - accuracy: 0.7238
Epoch 38/100
12/12 [==============================] - 0s 3ms/step - loss: 0.5787 - accuracy: 0.7322
Epoch 39/100
12/12 [==============================] - 0s 3ms/step - loss: 0.5759 - accuracy: 0.7294
Epoch 40/100
12/12 [==============================] - 0s 4ms/step - loss: 0.5730 - accuracy: 0.7252
Epoch 41/100
12/12 [==============================] - 0s 4ms/step - loss: 0.5701 - accuracy: 0.7364
Epoch 42/100
12/12 [==============================] - 0s 3ms/step - loss: 0.5704 - accuracy: 0.7183
Epoch 43/100
12/12 [==============================] - 0s 4ms/step - loss: 0.5612 - accuracy: 0.7294
Epoch 44/100
```

```
12/12 [==============================] - 0s 4ms/step - loss: 0.5576 - accuracy: 0.7294
Epoch 45/100
12/12 [==============================] - 0s 3ms/step - loss: 0.5578 - accuracy: 0.7350
Epoch 46/100
12/12 [==============================] - 0s 3ms/step - loss: 0.5545 - accuracy: 0.7350
Epoch 47/100
12/12 [==============================] - 0s 4ms/step - loss: 0.5485 - accuracy: 0.7378
Epoch 48/100
12/12 [==============================] - 0s 3ms/step - loss: 0.5471 - accuracy: 0.7336
Epoch 49/100
12/12 [==============================] - 0s 3ms/step - loss: 0.5439 - accuracy: 0.7378
Epoch 50/100
12/12 [==============================] - 0s 4ms/step - loss: 0.5409 - accuracy: 0.7392
Epoch 51/100
12/12 [==============================] - 0s 4ms/step - loss: 0.5398 - accuracy: 0.7308
Epoch 52/100
12/12 [==============================] - 0s 5ms/step - loss: 0.5413 - accuracy: 0.7392
Epoch 53/100
12/12 [==============================] - 0s 4ms/step - loss: 0.5352 - accuracy: 0.7350
Epoch 54/100
12/12 [==============================] - 0s 4ms/step - loss: 0.5407 - accuracy: 0.7364
Epoch 55/100
12/12 [==============================] - 0s 4ms/step - loss: 0.5461 - accuracy: 0.7127
Epoch 56/100
12/12 [==============================] - 0s 4ms/step - loss: 0.5341 - accuracy: 0.7406
Epoch 57/100
12/12 [==============================] - 0s 3ms/step - loss: 0.5264 - accuracy: 0.7364
Epoch 58/100
12/12 [==============================] - 0s 4ms/step - loss: 0.5306 - accuracy: 0.7406
Epoch 59/100
12/12 [==============================] - 0s 3ms/step - loss: 0.5261 - accuracy: 0.7392
Epoch 60/100
12/12 [==============================] - 0s 4ms/step - loss: 0.5283 - accuracy: 0.7392
Epoch 61/100
12/12 [==============================] - 0s 4ms/step - loss: 0.5187 - accuracy: 0.7462
Epoch 62/100
12/12 [==============================] - 0s 3ms/step - loss: 0.5171 - accuracy: 0.7462
Epoch 63/100
12/12 [==============================] - 0s 3ms/step - loss: 0.5139 - accuracy: 0.7490
Epoch 64/100
12/12 [==============================] - 0s 4ms/step - loss: 0.5148 - accuracy: 0.7462
Epoch 65/100
12/12 [==============================] - 0s 4ms/step - loss: 0.5121 - accuracy: 0.7476
Epoch 66/100
12/12 [==============================] - 0s 4ms/step - loss: 0.5149 - accuracy: 0.7434
Epoch 67/100
12/12 [==============================] - 0s 4ms/step - loss: 0.5089 - accuracy: 0.7490
Epoch 68/100
12/12 [==============================] - 0s 4ms/step - loss: 0.5064 - accuracy: 0.7531
Epoch 69/100
12/12 [==============================] - 0s 4ms/step - loss: 0.5053 - accuracy: 0.7476
Epoch 70/100
12/12 [==============================] - 0s 4ms/step - loss: 0.5066 - accuracy: 0.7503
Epoch 71/100
12/12 [==============================] - 0s 4ms/step - loss: 0.5035 - accuracy: 0.7517
Epoch 72/100
12/12 [==============================] - 0s 4ms/step - loss: 0.4994 - accuracy: 0.7587
Epoch 73/100
12/12 [==============================] - 0s 4ms/step - loss: 0.5010 - accuracy: 0.7559
Epoch 74/100
12/12 [==============================] - 0s 4ms/step - loss: 0.4990 - accuracy: 0.7476
Epoch 75/100
12/12 [==============================] - 0s 4ms/step - loss: 0.4947 - accuracy: 0.7601
Epoch 76/100
12/12 [==============================] - 0s 4ms/step - loss: 0.4974 - accuracy: 0.7503
Epoch 77/100
12/12 [==============================] - 0s 4ms/step - loss: 0.4930 - accuracy: 0.7531
Epoch 78/100
12/12 [==============================] - 0s 3ms/step - loss: 0.4914 - accuracy: 0.7615
Epoch 79/100
12/12 [==============================] - 0s 4ms/step - loss: 0.4900 - accuracy: 0.7615
Epoch 80/100
12/12 [==============================] - 0s 3ms/step - loss: 0.4898 - accuracy: 0.7573
Epoch 81/100
12/12 [==============================] - 0s 4ms/step - loss: 0.4898 - accuracy: 0.7531
Epoch 82/100
12/12 [==============================] - 0s 3ms/step - loss: 0.4868 - accuracy: 0.7629
Epoch 83/100
12/12 [==============================] - 0s 5ms/step - loss: 0.4871 - accuracy: 0.7559
Epoch 84/100
12/12 [==============================] - 0s 3ms/step - loss: 0.4849 - accuracy: 0.7517
Epoch 85/100
12/12 [==============================] - 0s 3ms/step - loss: 0.4831 - accuracy: 0.7657
Epoch 86/100
12/12 [==============================] - 0s 3ms/step - loss: 0.4841 - accuracy: 0.7531
Epoch 87/100
12/12 [==============================] - 0s 3ms/step - loss: 0.4813 - accuracy: 0.7559
```

```
Epoch 88/100
12/12 [==============================] - 0s 3ms/step - loss: 0.4836 - accuracy: 0.7559
Epoch 89/100
12/12 [==============================] - 0s 3ms/step - loss: 0.4760 - accuracy: 0.7643
Epoch 90/100
12/12 [==============================] - 0s 4ms/step - loss: 0.4769 - accuracy: 0.7601
Epoch 91/100
12/12 [==============================] - 0s 4ms/step - loss: 0.4764 - accuracy: 0.7573
Epoch 92/100
12/12 [==============================] - 0s 4ms/step - loss: 0.4727 - accuracy: 0.7629
Epoch 93/100
12/12 [==============================] - 0s 4ms/step - loss: 0.4728 - accuracy: 0.7643
Epoch 94/100
12/12 [==============================] - 0s 4ms/step - loss: 0.4738 - accuracy: 0.7643
Epoch 95/100
12/12 [==============================] - 0s 4ms/step - loss: 0.4725 - accuracy: 0.7587
Epoch 96/100
12/12 [==============================] - 0s 4ms/step - loss: 0.4693 - accuracy: 0.7643
Epoch 97/100
12/12 [==============================] - 0s 4ms/step - loss: 0.4674 - accuracy: 0.7559
Epoch 98/100
12/12 [==============================] - 0s 4ms/step - loss: 0.4677 - accuracy: 0.7685
Epoch 99/100
12/12 [==============================] - 0s 4ms/step - loss: 0.4657 - accuracy: 0.7629
Epoch 100/100
12/12 [==============================] - 0s 4ms/step - loss: 0.4632 - accuracy: 0.7685
```

```
Out[ ]:  <keras.src.callbacks.History at 0x79b085229240>
```

```
In [ ]: pred=ann.predict(scaler.transform([[52,1,0,125,212,0,1,168,0,1.0,2,2,3]]))
        print(pred)
        if(pred>0.5):
          print(1)
        else:
          print(0)
```

```
1/1 [==============================] - 0s 58ms/step
[[0.50485814]]
1
```

Total number of Edges

```
In [ ]: ann.summary()
```

```
Model: "sequential_20"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_73 (Dense)            (None, 6)                 84

 dense_74 (Dense)            (None, 6)                 42

 dense_75 (Dense)            (None, 1)                 7

=================================================================
Total params: 133 (532.00 Byte)
Trainable params: 133 (532.00 Byte)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

```
In [ ]:
```