PROJECT : PREDICT LUNG CANCER DIESEAS

Import Required Libraries

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

Out[ ]:

	Name	Surname	Age	Smokes	AreaQ	Alkhol	Result
0	John	Wick	35	3	5	4	1
1	John	Constantine	27	20	2	5	1
2	Camela	Anderson	30	0	5	2	0
3	Alex	Telles	28	0	8	1	0
4	Diego	Maradona	68	4	5	6	1
5	Cristiano	Ronaldo	34	0	10	0	0
6	Mihail	Tal	58	15	10	0	0
7	Kathy	Bates	22	12	5	2	0
8	Nicole	Kidman	45	2	6	0	0
9	Ray	Milland	52	18	4	5	1
10	Fredric	March	33	4	8	0	0
11	Yul	Brynner	18	10	6	3	0
12	Joan	Crawford	25	2	5	1	0
13	Jane	Wyman	28	20	2	8	1
14	Anna	Magnani	34	25	4	8	1
15	Katharine	Hepburn	39	18	8	1	0
16	Katharine	Hepburn	42	22	3	5	1
17	Barbra	Streisand	19	12	8	0	0
18	Maggie	Smith	62	5	4	3	1
19	Glenda	Jackson	73	10	7	6	1
20	Jane	Fonda	55	15	1	3	1
21	Maximilian	Schell	33	8	8	1	0
22	Gregory	Peck	22	20	6	2	0
23	Sidney	Poitier	44	5	8	1	0
24	Rex	Harrison	77	3	2	6	1
25	Lee	Marvin	21	20	5	3	0
26	Paul	Scofield	37	15	6	2	0
27	Rod	Steiger	34	12	8	0	0
28	John	Wayne	55	20	1	4	1
29	Gene	Hackman	40	20	2	7	1
30	Marlon	Brando	36	13	5	2	0
31	Jack	Lemmon	56	20	3	3	1
32	Jack	Nicholson	47	15	1	8	1
33	Peter	Finch	62	25	3	4	1
34	Richard	Dreyfuss	26	10	7	2	0
35	Dustin	Hoffman	25	20	8	2	0
36	Henry	Henry	59	20	3	4	1
37	Robert	Duvall	62	15	5	5	1
38	Ellen	Burstyn	33	25	8	2	0
39	Faye	Dunaway	37	10	5	3	0
40	Diane	Keaton	50	20	2	4	1
41	Jane	Fonda	47	12	8	0	0
42	Sally	Field	69	20	5	4	1
43	Sissy	Spacek	63	20	4	5	1
44	Jessica	Lange	39	15	7	2	0
45	Gwyneth	Paltrow	21	20	8	3	0
46	Halle	Berry	31	20	9	4	0
47	Nicole	Kidman	28	10	4	1	0
48	Charlize	Theron	53	20	6	3	1
49	Katharine	Hepburn	62	20	5	6	1
50	Katharine	Hepburn	42	12	6	2	0

	Name	Surname	Age	Smokes	AreaQ	Alkhol	Result
51	Barbra	Streisand	44	30	1	6	1
52	Maggie	Smith	26	34	1	8	1
53	Glenda	Jackson	35	20	5	1	0
54	Ernest	Borgnine	26	13	6	1	0
55	Alec	Guinness	77	20	5	4	1
56	Charlton	Heston	75	15	3	5	1
57	Gregory	Peck	43	30	3	8	1

In [1]:
```
#help(tf)
```

Checking missing values

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

Out[ ]:
```
Name       0
Surname    0
Age        0
Smokes     0
AreaQ      0
Alkhol     0
Result     0
dtype: int64
```

Separate X and Y

In [ ]:
```
x=df.iloc[:,2:-1].values
y=df.iloc[:,-1].values
```

Data into Training 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([[26, 13,  6,  1],
       [73, 10,  7,  6],
       [36, 13,  5,  2],
       [42, 12,  6,  2],
       [39, 15,  7,  2],
       [77, 20,  5,  4],
       [39, 18,  8,  1],
       [52, 18,  4,  5],
       [34, 12,  8,  0],
       [37, 15,  6,  2],
       [42, 22,  3,  5],
       [77,  3,  2,  6],
       [62, 25,  3,  4],
       [75, 15,  3,  5],
       [47, 12,  8,  0],
       [18, 10,  6,  3],
       [47, 15,  1,  8],
       [26, 34,  1,  8],
       [50, 20,  2,  4],
       [62, 15,  5,  5],
       [40, 20,  2,  7],
       [63, 20,  4,  5],
       [27, 20,  2,  5],
       [33,  8,  8,  1],
       [30,  0,  5,  2],
       [31, 20,  9,  4],
       [37, 10,  5,  3],
       [25, 20,  8,  2],
       [44,  5,  8,  1],
       [53, 20,  6,  3],
       [33,  4,  8,  0],
       [22, 20,  6,  2],
       [62,  5,  4,  3],
       [51, 25,  9,  0],
       [55, 15,  1,  3],
       [22, 12,  5,  2],
       [69, 20,  5,  4],
       [34, 25,  4,  8],
       [55, 20,  1,  4],
       [44, 30,  1,  6],
       [33, 25,  8,  2]])
```

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

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

Data Normalization

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

Out[ ]:
```
array([[26, 13,  6,  1],
       [73, 10,  7,  6],
       [36, 13,  5,  2],
       [42, 12,  6,  2],
       [39, 15,  7,  2],
       [77, 20,  5,  4],
       [39, 18,  8,  1],
       [52, 18,  4,  5],
       [34, 12,  8,  0],
       [37, 15,  6,  2],
       [42, 22,  3,  5],
       [77,  3,  2,  6],
       [62, 25,  3,  4],
       [75, 15,  3,  5],
       [47, 12,  8,  0],
       [18, 10,  6,  3],
       [47, 15,  1,  8],
       [26, 34,  1,  8],
       [50, 20,  2,  4],
       [62, 15,  5,  5],
       [40, 20,  2,  7],
       [63, 20,  4,  5],
       [27, 20,  2,  5],
       [33,  8,  8,  1],
       [30,  0,  5,  2],
       [31, 20,  9,  4],
       [37, 10,  5,  3],
       [25, 20,  8,  2],
       [44,  5,  8,  1],
       [53, 20,  6,  3],
       [33,  4,  8,  0],
       [22, 20,  6,  2],
       [62,  5,  4,  3],
       [51, 25,  9,  0],
       [55, 15,  1,  3],
       [22, 12,  5,  2],
       [69, 20,  5,  4],
       [34, 25,  4,  8],
       [55, 20,  1,  4],
       [44, 30,  1,  6],
       [33, 25,  8,  2]])
```

ANN

In [ ]:
```
#Initialize Artificial Nueral Network(ANN)
ann=tf.keras.models.Sequential()
```

First Hidden layer used RELU Activation function

In [ ]:
```
#Create a First hidden layer
ann.add(tf.keras.layers.Dense(6,activation='relu'))
```

Second Hidden layer used RELU Activation function

In [ ]:
```
#Create a Second hidden layer
ann.add(tf.keras.layers.Dense(6,activation='relu'))
```

Output layer used Sigmoid Activation function

In [ ]:
```
#Output layer
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=32,epochs=100)
```

```
Epoch 1/100
2/2 [==============================] - 0s 8ms/step - loss: 0.3911 - accuracy: 0.6829
Epoch 2/100
2/2 [==============================] - 0s 7ms/step - loss: 0.3889 - accuracy: 0.7073
Epoch 3/100
2/2 [==============================] - 0s 7ms/step - loss: 0.3865 - accuracy: 0.7317
Epoch 4/100
2/2 [==============================] - 0s 8ms/step - loss: 0.3845 - accuracy: 0.7317
Epoch 5/100
2/2 [==============================] - 0s 8ms/step - loss: 0.3821 - accuracy: 0.7317
Epoch 6/100
2/2 [==============================] - 0s 6ms/step - loss: 0.3801 - accuracy: 0.7317
Epoch 7/100
2/2 [==============================] - 0s 6ms/step - loss: 0.3783 - accuracy: 0.7317
Epoch 8/100
2/2 [==============================] - 0s 7ms/step - loss: 0.3764 - accuracy: 0.7317
Epoch 9/100
2/2 [==============================] - 0s 7ms/step - loss: 0.3744 - accuracy: 0.7317
Epoch 10/100
2/2 [==============================] - 0s 6ms/step - loss: 0.3726 - accuracy: 0.7317
Epoch 11/100
2/2 [==============================] - 0s 7ms/step - loss: 0.3708 - accuracy: 0.7317
Epoch 12/100
2/2 [==============================] - 0s 7ms/step - loss: 0.3690 - accuracy: 0.7317
Epoch 13/100
2/2 [==============================] - 0s 7ms/step - loss: 0.3673 - accuracy: 0.7317
Epoch 14/100
2/2 [==============================] - 0s 6ms/step - loss: 0.3657 - accuracy: 0.7317
Epoch 15/100
2/2 [==============================] - 0s 6ms/step - loss: 0.3639 - accuracy: 0.7317
Epoch 16/100
2/2 [==============================] - 0s 6ms/step - loss: 0.3622 - accuracy: 0.7317
Epoch 17/100
2/2 [==============================] - 0s 6ms/step - loss: 0.3602 - accuracy: 0.7317
Epoch 18/100
2/2 [==============================] - 0s 7ms/step - loss: 0.3585 - accuracy: 0.7317
Epoch 19/100
2/2 [==============================] - 0s 8ms/step - loss: 0.3566 - accuracy: 0.7805
Epoch 20/100
2/2 [==============================] - 0s 8ms/step - loss: 0.3546 - accuracy: 0.7805
Epoch 21/100
2/2 [==============================] - 0s 7ms/step - loss: 0.3527 - accuracy: 0.7805
Epoch 22/100
2/2 [==============================] - 0s 7ms/step - loss: 0.3508 - accuracy: 0.7805
Epoch 23/100
2/2 [==============================] - 0s 7ms/step - loss: 0.3490 - accuracy: 0.7805
Epoch 24/100
2/2 [==============================] - 0s 8ms/step - loss: 0.3471 - accuracy: 0.7805
Epoch 25/100
2/2 [==============================] - 0s 7ms/step - loss: 0.3451 - accuracy: 0.7805
Epoch 26/100
2/2 [==============================] - 0s 7ms/step - loss: 0.3430 - accuracy: 0.7805
Epoch 27/100
2/2 [==============================] - 0s 8ms/step - loss: 0.3414 - accuracy: 0.7805
Epoch 28/100
2/2 [==============================] - 0s 8ms/step - loss: 0.3396 - accuracy: 0.8049
Epoch 29/100
2/2 [==============================] - 0s 6ms/step - loss: 0.3376 - accuracy: 0.8293
Epoch 30/100
2/2 [==============================] - 0s 7ms/step - loss: 0.3362 - accuracy: 0.8293
Epoch 31/100
2/2 [==============================] - 0s 8ms/step - loss: 0.3348 - accuracy: 0.8537
Epoch 32/100
2/2 [==============================] - 0s 8ms/step - loss: 0.3335 - accuracy: 0.8537
Epoch 33/100
2/2 [==============================] - 0s 7ms/step - loss: 0.3321 - accuracy: 0.8537
Epoch 34/100
2/2 [==============================] - 0s 9ms/step - loss: 0.3304 - accuracy: 0.8537
Epoch 35/100
2/2 [==============================] - 0s 6ms/step - loss: 0.3285 - accuracy: 0.8537
Epoch 36/100
2/2 [==============================] - 0s 7ms/step - loss: 0.3270 - accuracy: 0.8780
Epoch 37/100
2/2 [==============================] - 0s 7ms/step - loss: 0.3254 - accuracy: 0.9024
Epoch 38/100
2/2 [==============================] - 0s 7ms/step - loss: 0.3244 - accuracy: 0.9024
Epoch 39/100
2/2 [==============================] - 0s 8ms/step - loss: 0.3228 - accuracy: 0.9024
Epoch 40/100
2/2 [==============================] - 0s 7ms/step - loss: 0.3210 - accuracy: 0.9024
Epoch 41/100
2/2 [==============================] - 0s 7ms/step - loss: 0.3193 - accuracy: 0.9024
Epoch 42/100
2/2 [==============================] - 0s 7ms/step - loss: 0.3179 - accuracy: 0.9024
Epoch 43/100
2/2 [==============================] - 0s 8ms/step - loss: 0.3163 - accuracy: 0.9024
Epoch 44/100
```

```
2/2 [==============================] - 0s 13ms/step - loss: 0.3149 - accuracy: 0.9024
Epoch 45/100
2/2 [==============================] - 0s 10ms/step - loss: 0.3138 - accuracy: 0.9024
Epoch 46/100
2/2 [==============================] - 0s 8ms/step - loss: 0.3125 - accuracy: 0.9024
Epoch 47/100
2/2 [==============================] - 0s 6ms/step - loss: 0.3110 - accuracy: 0.9024
Epoch 48/100
2/2 [==============================] - 0s 8ms/step - loss: 0.3099 - accuracy: 0.9024
Epoch 49/100
2/2 [==============================] - 0s 8ms/step - loss: 0.3086 - accuracy: 0.9024
Epoch 50/100
2/2 [==============================] - 0s 9ms/step - loss: 0.3072 - accuracy: 0.9024
Epoch 51/100
2/2 [==============================] - 0s 7ms/step - loss: 0.3060 - accuracy: 0.9024
Epoch 52/100
2/2 [==============================] - 0s 7ms/step - loss: 0.3049 - accuracy: 0.9024
Epoch 53/100
2/2 [==============================] - 0s 8ms/step - loss: 0.3038 - accuracy: 0.9024
Epoch 54/100
2/2 [==============================] - 0s 8ms/step - loss: 0.3026 - accuracy: 0.9024
Epoch 55/100
2/2 [==============================] - 0s 7ms/step - loss: 0.3015 - accuracy: 0.9024
Epoch 56/100
2/2 [==============================] - 0s 7ms/step - loss: 0.3005 - accuracy: 0.9024
Epoch 57/100
2/2 [==============================] - 0s 9ms/step - loss: 0.2994 - accuracy: 0.9024
Epoch 58/100
2/2 [==============================] - 0s 7ms/step - loss: 0.2983 - accuracy: 0.9024
Epoch 59/100
2/2 [==============================] - 0s 7ms/step - loss: 0.2974 - accuracy: 0.9024
Epoch 60/100
2/2 [==============================] - 0s 7ms/step - loss: 0.2964 - accuracy: 0.9024
Epoch 61/100
2/2 [==============================] - 0s 6ms/step - loss: 0.2953 - accuracy: 0.9024
Epoch 62/100
2/2 [==============================] - 0s 9ms/step - loss: 0.2944 - accuracy: 0.9268
Epoch 63/100
2/2 [==============================] - 0s 8ms/step - loss: 0.2934 - accuracy: 0.9268
Epoch 64/100
2/2 [==============================] - 0s 9ms/step - loss: 0.2924 - accuracy: 0.9268
Epoch 65/100
2/2 [==============================] - 0s 8ms/step - loss: 0.2914 - accuracy: 0.9268
Epoch 66/100
2/2 [==============================] - 0s 9ms/step - loss: 0.2904 - accuracy: 0.9268
Epoch 67/100
2/2 [==============================] - 0s 9ms/step - loss: 0.2895 - accuracy: 0.9268
Epoch 68/100
2/2 [==============================] - 0s 9ms/step - loss: 0.2887 - accuracy: 0.9268
Epoch 69/100
2/2 [==============================] - 0s 9ms/step - loss: 0.2876 - accuracy: 0.9268
Epoch 70/100
2/2 [==============================] - 0s 9ms/step - loss: 0.2867 - accuracy: 0.9268
Epoch 71/100
2/2 [==============================] - 0s 8ms/step - loss: 0.2859 - accuracy: 0.9268
Epoch 72/100
2/2 [==============================] - 0s 7ms/step - loss: 0.2850 - accuracy: 0.9268
Epoch 73/100
2/2 [==============================] - 0s 7ms/step - loss: 0.2843 - accuracy: 0.9268
Epoch 74/100
2/2 [==============================] - 0s 8ms/step - loss: 0.2835 - accuracy: 0.9268
Epoch 75/100
2/2 [==============================] - 0s 7ms/step - loss: 0.2828 - accuracy: 0.9512
Epoch 76/100
2/2 [==============================] - 0s 7ms/step - loss: 0.2820 - accuracy: 0.9512
Epoch 77/100
2/2 [==============================] - 0s 8ms/step - loss: 0.2814 - accuracy: 0.9512
Epoch 78/100
2/2 [==============================] - 0s 8ms/step - loss: 0.2803 - accuracy: 0.9512
Epoch 79/100
2/2 [==============================] - 0s 8ms/step - loss: 0.2798 - accuracy: 0.9512
Epoch 80/100
2/2 [==============================] - 0s 8ms/step - loss: 0.2783 - accuracy: 0.9512
Epoch 81/100
2/2 [==============================] - 0s 7ms/step - loss: 0.2774 - accuracy: 0.9512
Epoch 82/100
2/2 [==============================] - 0s 7ms/step - loss: 0.2765 - accuracy: 0.9512
Epoch 83/100
2/2 [==============================] - 0s 8ms/step - loss: 0.2756 - accuracy: 0.9512
Epoch 84/100
2/2 [==============================] - 0s 8ms/step - loss: 0.2746 - accuracy: 0.9512
Epoch 85/100
2/2 [==============================] - 0s 10ms/step - loss: 0.2737 - accuracy: 0.9512
Epoch 86/100
2/2 [==============================] - 0s 7ms/step - loss: 0.2729 - accuracy: 0.9512
Epoch 87/100
2/2 [==============================] - 0s 9ms/step - loss: 0.2720 - accuracy: 0.9512
```

```
Epoch 88/100
2/2 [==============================] - 0s 8ms/step - loss: 0.2713 - accuracy: 0.9512
Epoch 89/100
2/2 [==============================] - 0s 8ms/step - loss: 0.2705 - accuracy: 0.9512
Epoch 90/100
2/2 [==============================] - 0s 7ms/step - loss: 0.2696 - accuracy: 0.9512
Epoch 91/100
2/2 [==============================] - 0s 7ms/step - loss: 0.2689 - accuracy: 0.9512
Epoch 92/100
2/2 [==============================] - 0s 7ms/step - loss: 0.2682 - accuracy: 0.9512
Epoch 93/100
2/2 [==============================] - 0s 7ms/step - loss: 0.2676 - accuracy: 0.9512
Epoch 94/100
2/2 [==============================] - 0s 9ms/step - loss: 0.2668 - accuracy: 0.9756
Epoch 95/100
2/2 [==============================] - 0s 8ms/step - loss: 0.2660 - accuracy: 0.9756
Epoch 96/100
2/2 [==============================] - 0s 8ms/step - loss: 0.2650 - accuracy: 0.9756
Epoch 97/100
2/2 [==============================] - 0s 7ms/step - loss: 0.2642 - accuracy: 0.9756
Epoch 98/100
2/2 [==============================] - 0s 7ms/step - loss: 0.2634 - accuracy: 0.9756
Epoch 99/100
2/2 [==============================] - 0s 6ms/step - loss: 0.2626 - accuracy: 0.9756
Epoch 100/100
2/2 [==============================] - 0s 8ms/step - loss: 0.2618 - accuracy: 1.0000
```

Out[ ]: <keras.src.callbacks.History at 0x782b9578b640>

In [ ]:
```
pred=ann.predict(scaler.transform([[30,0,5,2]]))
print(pred)
if(pred>0.5):
  print(1)
else:
  print(0)
```

```
1/1 [==============================] - 0s 33ms/step
[[0.51077026]]
1
```

Total number of Edges

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

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_6 (Dense)             (None, 6)                 30

 dense_7 (Dense)             (None, 6)                 42

 dense_8 (Dense)             (None, 1)                 7

=================================================================
Total params: 79 (316.00 Byte)
Trainable params: 79 (316.00 Byte)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```