In [1]:
```
import warnings
warnings.filterwarnings('ignore')
```

In [2]:
```
import tensorflow as tf
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.datasets import load_breast_cancer
```

load dataset

In [3]:
```
df=load_breast_cancer()
```

Split-Train-Test

In [4]:
```
X_train,X_test,y_train,y_test=train_test_split(df.data,df.target,test_size=
0.20,random_state=42)
```

Standardization

In [5]:
```
sc=StandardScaler()
X_train=sc.fit_transform(X_train)
X_test=sc.transform(X_test)
```

model training

In [6]:
```
model=tf.keras.models.Sequential([tf.keras.layers.Dense(1,activation='sigmo
id',input_shape=(X_train.shape[1],))])
```

In [7]:
```
model.compile(optimizer='adam',loss='binary_crossentropy',metrics=['accurac
y'])
```

In [8]:
```
model.fit(X_train,y_train,epochs=10)
```

```
Epoch 1/10
15/15 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.5040 - loss: 0.8651
Epoch 2/10
15/15 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.5222 - loss: 0.8216
Epoch 3/10
15/15 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.6134 - loss: 0.6736
Epoch 4/10
15/15 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.6383 - loss: 0.6226
Epoch 5/10
15/15 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.6760 - loss: 0.5601
Epoch 6/10
15/15 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - accuracy: 0.7518 - loss: 0.5022
Epoch 7/10
15/15 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - accuracy: 0.7847 - loss: 0.4685
Epoch 8/10
15/15 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.7892 - loss: 0.4443
Epoch 9/10
15/15 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8548 - loss: 0.3854
Epoch 10/10
15/15 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8599 - loss: 0.3666
```

Out[8]:
```
<keras.src.callbacks.history.History at 0x2591d10aee0>
```

Prediction

In [9]:
```
y_pred=model.predict(X_test)
```

```
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 11ms/step
```

evalution of model

In [10]:
```
test_loss,test_accuracy=model.evaluate(X_test,y_test)
```

```
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.9066 - loss: 0.3159
```

In [11]:
```
print("accuracy is",test_accuracy)
```

```
accuracy is 0.8862847089767456
```

In [ ]:
```

```