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

```
df=pd.read_csv('diabetes.csv')
```

```
df.head()
```

Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedi
6	148	72	35	0	33.6	
1	85	66	29	0	26.6	
8	183	64	0	0	23.3	
1	89	66	23	94	28.1	
0	137	40	35	168	43.1	

```
df.corr()['Outcome']
df.shape
```

```
    (768, 9)
```

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

```
    array([[  6.   , 148.   ,  72.   , ...,  33.6  ,   0.627,  50.   ],
           [  1.   ,  85.   ,  66.   , ...,  26.6  ,   0.351,  31.   ],
           [  8.   , 183.   ,  64.   , ...,  23.3  ,   0.672,  32.   ],
           ...,
           [  5.   , 121.   ,  72.   , ...,  26.2  ,   0.245,  30.   ],
           [  1.   , 126.   ,  60.   , ...,  30.1  ,   0.349,  47.   ],
           [  1.   ,  93.   ,  70.   , ...,  30.4  ,   0.315,  23.   ]])
```

```
from sklearn.preprocessing import StandardScaler
sc=StandardScaler()
X=sc.fit_transform(x)
```

```
 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=1)
```

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

```
model=Sequential()
model.add(Dense(32,activation='relu',input_dim=8))# first layer
model.add(Dense(1,activation='sigmoid'))#output layer
model.compile(optimizer='Adam',loss='binary_crossentropy',metrics=['accuracy'])
model.fit(X_train,y_train,batch_size=32,epochs=10,validation_data=(X_test,y_test))
```

```
    Epoch 1/10
    20/20 [==============================] - 1s 13ms/step - loss: 0.7505 - accuracy: 0.4186 - val_loss: 0.7066 - val_accuracy: 0
    Epoch 2/10
    20/20 [==============================] - 0s 4ms/step - loss: 0.6734 - accuracy: 0.6107 - val_loss: 0.6377 - val_accuracy: 0.
    Epoch 3/10
    20/20 [==============================] - 0s 5ms/step - loss: 0.6184 - accuracy: 0.7248 - val_loss: 0.5880 - val_accuracy: 0.
    Epoch 4/10
    20/20 [==============================] - 0s 4ms/step - loss: 0.5772 - accuracy: 0.7524 - val_loss: 0.5544 - val_accuracy: 0.
    Epoch 5/10
    20/20 [==============================] - 0s 3ms/step - loss: 0.5497 - accuracy: 0.7622 - val_loss: 0.5295 - val_accuracy: 0.
    Epoch 6/10
    20/20 [==============================] - 0s 4ms/step - loss: 0.5267 - accuracy: 0.7655 - val_loss: 0.5132 - val_accuracy: 0.
    Epoch 7/10
    20/20 [==============================] - 0s 4ms/step - loss: 0.5113 - accuracy: 0.7736 - val_loss: 0.5008 - val_accuracy: 0.
    Epoch 8/10
    20/20 [==============================] - 0s 4ms/step - loss: 0.4991 - accuracy: 0.7752 - val_loss: 0.4918 - val_accuracy: 0.
    Epoch 9/10
```

```
20/20 [==============================] - 0s 4ms/step - loss: 0.4893 - accuracy: 0.7769 - val_loss: 0.4848 - val_accuracy: 0.
Epoch 10/10
20/20 [==============================] - 0s 4ms/step - loss: 0.4824 - accuracy: 0.7818 - val_loss: 0.4780 - val_accuracy: 0.
<keras.src.callbacks.History at 0x78b5c8cde470>
```

```
pip install -U keras-tuner
```

```
Collecting keras-tuner
  Downloading keras_tuner-1.4.5-py3-none-any.whl (129 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 129.5/129.5 kB 3.4 MB/s eta 0:00:00
Collecting keras-core (from keras-tuner)
  Downloading keras_core-0.1.7-py3-none-any.whl (950 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 950.8/950.8 kB 10.3 MB/s eta 0:00:00
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages (from keras-tuner) (23.2)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from keras-tuner) (2.31.0)
Collecting kt-legacy (from keras-tuner)
  Downloading kt_legacy-1.0.5-py3-none-any.whl (9.6 kB)
Requirement already satisfied: absl-py in /usr/local/lib/python3.10/dist-packages (from keras-core->keras-tuner) (1.4.0)
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from keras-core->keras-tuner) (1.23.5)
Requirement already satisfied: rich in /usr/local/lib/python3.10/dist-packages (from keras-core->keras-tuner) (13.6.0)
Collecting namex (from keras-core->keras-tuner)
  Downloading namex-0.0.7-py3-none-any.whl (5.8 kB)
Requirement already satisfied: h5py in /usr/local/lib/python3.10/dist-packages (from keras-core->keras-tuner) (3.9.0)
Requirement already satisfied: dm-tree in /usr/local/lib/python3.10/dist-packages (from keras-core->keras-tuner) (0.1.8)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests->keras-tun
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests->keras-tuner) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests->keras-tuner) (2
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests->keras-tuner) (2
Requirement already satisfied: markdown-it-py>=2.2.0 in /usr/local/lib/python3.10/dist-packages (from rich->keras-core->kera
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in /usr/local/lib/python3.10/dist-packages (from rich->keras-core->ke
Requirement already satisfied: mdurl~=0.1 in /usr/local/lib/python3.10/dist-packages (from markdown-it-py>=2.2.0->rich->kera
Installing collected packages: namex, kt-legacy, keras-core, keras-tuner
Successfully installed keras-core-0.1.7 keras-tuner-1.4.5 kt-legacy-1.0.5 namex-0.0.7
```

```
import kerastuner as kt
```

```
Using TensorFlow backend
<ipython-input-14-5fd8096cdee5>:1: DeprecationWarning: `import kerastuner` is deprecated, please use `import keras_tuner`.
  import kerastuner as kt
```

```
def build_model(hp):
  model=Sequential()
  model.add(Dense(32,activation='relu',input_dim=8))
  model.add(Dense(1,activation='sigmoid'))
  opt=hp.Choice('optimizer',values=['sgd','adam','rmsprop',
                                   'adadelta'])
  model.compile(optimizer=opt,loss='binary_crossentropy',
                metrics=['accuracy'])
  return  model
```

```
tuner=kt.RandomSearch(build_model,objective='val_accuracy',
                  max_trials=6)
```

```
tuner.search(X_train,y_train,epochs=10,
             validation_data=(X_test,y_test))
```

```
Trial 4 Complete [00h 00m 02s]
val_accuracy: 0.33766233921051025

Best val_accuracy So Far: 0.8051947951316833
Total elapsed time: 00h 00m 14s
```

```
tuner.get_best_hyperparameters()[0].values
```

```
{'optimizer': 'adam'}
```

```
model=tuner.get_best_models(num_models=1)[0]
```

```
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
```

```
=================================================================
 dense (Dense)               (None, 32)                288       
                                                                 
 dense_1 (Dense)             (None, 1)                 33        
                                                                 
=================================================================
Total params: 321 (1.25 KB)
Trainable params: 321 (1.25 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

```
model.fit(X_train,y_train,epochs=100,initial_epoch=11,
          validation_data=(X_test,y_test))
```

```
Epoch 12/100
20/20 [==============================] - 1s 12ms/step - loss: 0.4791 - accuracy: 0.7736 - val_loss: 0.4748 - val_accuracy: 0
Epoch 13/100
20/20 [==============================] - 0s 5ms/step - loss: 0.4729 - accuracy: 0.7638 - val_loss: 0.4701 - val_accuracy: 0.
Epoch 14/100
20/20 [==============================] - 0s 4ms/step - loss: 0.4679 - accuracy: 0.7720 - val_loss: 0.4678 - val_accuracy: 0.
Epoch 15/100
20/20 [==============================] - 0s 4ms/step - loss: 0.4631 - accuracy: 0.7785 - val_loss: 0.4659 - val_accuracy: 0.
Epoch 16/100
20/20 [==============================] - 0s 4ms/step - loss: 0.4598 - accuracy: 0.7834 - val_loss: 0.4642 - val_accuracy: 0.
Epoch 17/100
20/20 [==============================] - 0s 5ms/step - loss: 0.4566 - accuracy: 0.7850 - val_loss: 0.4625 - val_accuracy: 0.
Epoch 18/100
20/20 [==============================] - 0s 6ms/step - loss: 0.4545 - accuracy: 0.7883 - val_loss: 0.4615 - val_accuracy: 0.
Epoch 19/100
20/20 [==============================] - 0s 5ms/step - loss: 0.4523 - accuracy: 0.7883 - val_loss: 0.4593 - val_accuracy: 0.
Epoch 20/100
20/20 [==============================] - 0s 6ms/step - loss: 0.4513 - accuracy: 0.7850 - val_loss: 0.4590 - val_accuracy: 0.
Epoch 21/100
20/20 [==============================] - 0s 6ms/step - loss: 0.4495 - accuracy: 0.7850 - val_loss: 0.4578 - val_accuracy: 0.
Epoch 22/100
20/20 [==============================] - 0s 6ms/step - loss: 0.4483 - accuracy: 0.7850 - val_loss: 0.4579 - val_accuracy: 0.
Epoch 23/100
20/20 [==============================] - 0s 6ms/step - loss: 0.4469 - accuracy: 0.7818 - val_loss: 0.4590 - val_accuracy: 0.
Epoch 24/100
20/20 [==============================] - 0s 6ms/step - loss: 0.4465 - accuracy: 0.7801 - val_loss: 0.4572 - val_accuracy: 0.
Epoch 25/100
20/20 [==============================] - 0s 6ms/step - loss: 0.4453 - accuracy: 0.7785 - val_loss: 0.4572 - val_accuracy: 0.
Epoch 26/100
20/20 [==============================] - 0s 7ms/step - loss: 0.4441 - accuracy: 0.7834 - val_loss: 0.4583 - val_accuracy: 0.
Epoch 27/100
20/20 [==============================] - 0s 6ms/step - loss: 0.4440 - accuracy: 0.7818 - val_loss: 0.4565 - val_accuracy: 0.
Epoch 28/100
20/20 [==============================] - 0s 5ms/step - loss: 0.4425 - accuracy: 0.7850 - val_loss: 0.4570 - val_accuracy: 0.
Epoch 29/100
20/20 [==============================] - 0s 5ms/step - loss: 0.4416 - accuracy: 0.7834 - val_loss: 0.4581 - val_accuracy: 0.
Epoch 30/100
20/20 [==============================] - 0s 6ms/step - loss: 0.4404 - accuracy: 0.7850 - val_loss: 0.4564 - val_accuracy: 0.
Epoch 31/100
20/20 [==============================] - 0s 6ms/step - loss: 0.4395 - accuracy: 0.7818 - val_loss: 0.4558 - val_accuracy: 0.
Epoch 32/100
20/20 [==============================] - 0s 5ms/step - loss: 0.4387 - accuracy: 0.7834 - val_loss: 0.4563 - val_accuracy: 0.
Epoch 33/100
20/20 [==============================] - 0s 5ms/step - loss: 0.4379 - accuracy: 0.7883 - val_loss: 0.4566 - val_accuracy: 0.
Epoch 34/100
20/20 [==============================] - 0s 6ms/step - loss: 0.4370 - accuracy: 0.7899 - val_loss: 0.4570 - val_accuracy: 0.
Epoch 35/100
20/20 [==============================] - 0s 6ms/step - loss: 0.4365 - accuracy: 0.7850 - val_loss: 0.4575 - val_accuracy: 0.
Epoch 36/100
20/20 [==============================] - 0s 5ms/step - loss: 0.4359 - accuracy: 0.7915 - val_loss: 0.4594 - val_accuracy: 0.
Epoch 37/100
20/20 [==============================] - 0s 5ms/step - loss: 0.4347 - accuracy: 0.7932 - val_loss: 0.4583 - val_accuracy: 0.
Epoch 38/100
20/20 [==============================] - 0s 6ms/step - loss: 0.4344 - accuracy: 0.7899 - val_loss: 0.4569 - val_accuracy: 0.
Epoch 39/100
20/20 [==============================] - 0s 5ms/step - loss: 0.4339 - accuracy: 0.7915 - val_loss: 0.4574 - val_accuracy: 0.
Epoch 40/100
20/20 [==============================] - 0s 5ms/step - loss: 0.4327 - accuracy: 0.7948 - val_loss: 0.4576 - val_accuracy: 0.
```

```
# TUNING no of units in each layer
```

```
def build_model(hp):
  model=Sequential()
  units=hp.Int('units',min_value=8,max_value=128,step=8)
  model.add(Dense(units=units,activation='relu',input_dim=8))
  model.add(Dense(1,activation='sigmoid'))
  model.compile(optimizer='rmsprop',loss='binary_crossentropy',
                metrics=['accuracy'])
  return  model
```

```
tuner=kt.RandomSearch(build_model,
                      objective='val_accuracy',
                      max_trials=5,
                      directory='mydir',
                      project_name='srk'
    )
```

```
tuner.search(X_train,y_train,epochs=5,
             validation_data=(X_test,y_test))
```

```
    Trial 5 Complete [00h 00m 02s]
    val_accuracy: 0.798701286315918

    Best val_accuracy So Far: 0.798701286315918
    Total elapsed time: 00h 00m 12s
```

```
tuner.get_best_hyperparameters()[0].values
```

```
    {'units': 24}
```

```
model=tuner.get_best_models(num_models=1)[0]
```

```
model.fit(X_train,y_train,epochs=100,initial_epoch=6,
          validation_data=(X_test,y_test))
```

```
    Epoch 7/100
    20/20 [==============================] - 0s 6ms/step - loss: 0.4276 - accuracy: 0.7883 - val_loss: 0.4808 - val_accuracy: 0.
    Epoch 8/100
    20/20 [==============================] - 0s 3ms/step - loss: 0.4273 - accuracy: 0.7850 - val_loss: 0.4819 - val_accuracy: 0.
    Epoch 9/100
    20/20 [==============================] - 0s 3ms/step - loss: 0.4267 - accuracy: 0.7850 - val_loss: 0.4811 - val_accuracy: 0.
    Epoch 10/100
    20/20 [==============================] - 0s 3ms/step - loss: 0.4267 - accuracy: 0.7850 - val_loss: 0.4820 - val_accuracy: 0.
    Epoch 11/100
    20/20 [==============================] - 0s 3ms/step - loss: 0.4264 - accuracy: 0.7883 - val_loss: 0.4821 - val_accuracy: 0.
    Epoch 12/100
    20/20 [==============================] - 0s 4ms/step - loss: 0.4260 - accuracy: 0.7883 - val_loss: 0.4812 - val_accuracy: 0.
    Epoch 13/100
    20/20 [==============================] - 0s 3ms/step - loss: 0.4259 - accuracy: 0.7964 - val_loss: 0.4806 - val_accuracy: 0.
    Epoch 14/100
    20/20 [==============================] - 0s 3ms/step - loss: 0.4255 - accuracy: 0.7866 - val_loss: 0.4810 - val_accuracy: 0.
    Epoch 15/100
    20/20 [==============================] - 0s 4ms/step - loss: 0.4254 - accuracy: 0.7850 - val_loss: 0.4805 - val_accuracy: 0.
    Epoch 16/100
    20/20 [==============================] - 0s 4ms/step - loss: 0.4249 - accuracy: 0.7850 - val_loss: 0.4816 - val_accuracy: 0.
    Epoch 17/100
    20/20 [==============================] - 0s 4ms/step - loss: 0.4250 - accuracy: 0.7883 - val_loss: 0.4812 - val_accuracy: 0.
    Epoch 18/100
    20/20 [==============================] - 0s 4ms/step - loss: 0.4244 - accuracy: 0.7899 - val_loss: 0.4810 - val_accuracy: 0.
    Epoch 19/100
    20/20 [==============================] - 0s 4ms/step - loss: 0.4248 - accuracy: 0.7948 - val_loss: 0.4806 - val_accuracy: 0.
    Epoch 20/100
    20/20 [==============================] - 0s 4ms/step - loss: 0.4245 - accuracy: 0.7899 - val_loss: 0.4806 - val_accuracy: 0.
    Epoch 21/100
    20/20 [==============================] - 0s 4ms/step - loss: 0.4238 - accuracy: 0.7915 - val_loss: 0.4818 - val_accuracy: 0.
    Epoch 22/100
    20/20 [==============================] - 0s 4ms/step - loss: 0.4240 - accuracy: 0.7883 - val_loss: 0.4815 - val_accuracy: 0.
    Epoch 23/100
    20/20 [==============================] - 0s 3ms/step - loss: 0.4232 - accuracy: 0.7883 - val_loss: 0.4816 - val_accuracy: 0.
    Epoch 24/100
    20/20 [==============================] - 0s 4ms/step - loss: 0.4233 - accuracy: 0.7850 - val_loss: 0.4820 - val_accuracy: 0.
    Epoch 25/100
    20/20 [==============================] - 0s 4ms/step - loss: 0.4229 - accuracy: 0.7932 - val_loss: 0.4825 - val_accuracy: 0.
    Epoch 26/100
    20/20 [==============================] - 0s 4ms/step - loss: 0.4229 - accuracy: 0.7964 - val_loss: 0.4832 - val_accuracy: 0.
    Epoch 27/100
    20/20 [==============================] - 0s 4ms/step - loss: 0.4224 - accuracy: 0.7899 - val_loss: 0.4826 - val_accuracy: 0.
    Epoch 28/100
    20/20 [==============================] - 0s 4ms/step - loss: 0.4223 - accuracy: 0.7899 - val_loss: 0.4833 - val_accuracy: 0.
    Epoch 29/100
    20/20 [==============================] - 0s 3ms/step - loss: 0.4219 - accuracy: 0.7883 - val_loss: 0.4830 - val_accuracy: 0.
    Epoch 30/100
    20/20 [==============================] - 0s 4ms/step - loss: 0.4218 - accuracy: 0.7915 - val_loss: 0.4843 - val_accuracy: 0.
```

```
Epoch 31/100
20/20 [==============================] - 0s 4ms/step - loss: 0.4215 - accuracy: 0.7899 - val_loss: 0.4832 - val_accuracy: 0.
Epoch 32/100
20/20 [==============================] - 0s 4ms/step - loss: 0.4210 - accuracy: 0.7866 - val_loss: 0.4832 - val_accuracy: 0.
Epoch 33/100
20/20 [==============================] - 0s 3ms/step - loss: 0.4210 - accuracy: 0.7850 - val_loss: 0.4828 - val_accuracy: 0.
Epoch 34/100
20/20 [==============================] - 0s 3ms/step - loss: 0.4207 - accuracy: 0.7915 - val_loss: 0.4829 - val_accuracy: 0.
Epoch 35/100
20/20 [==============================] - 0s 3ms/step - loss: 0.4207 - accuracy: 0.7850 - val_loss: 0.4834 - val_accuracy: 0.
```

```
  # TUNIG no of layers in network
  def build_model(hp):
   model=Sequential()
   model.add(Dense(72,activation='relu',input_dim=8))
   for i in range(hp.Int('num_layers',min_value=1,max_value=11)):
     model.add(Dense(72,activation='relu'))
   model.add(Dense(1,activation='sigmoid'))
   model.compile(optimizer='rmsprop',loss='binary_crossentropy',
                metrics=['accuracy'])
   return  model
```

```
tuner=kt.RandomSearch(build_model,
                      objective='val_accuracy',
                      max_trials=5,
                      directory='mydir',
                      project_name='num_layers'
    )
```

```
tuner.search(X_train,y_train,epochs=5,
             validation_data=(X_test,y_test))
```

```
Trial 5 Complete [00h 00m 02s]
val_accuracy: 0.8246753215789795

Best val_accuracy So Far: 0.8246753215789795
Total elapsed time: 00h 00m 15s
```

```
tuner.get_best_hyperparameters()[0].values
```

```
{'num_layers': 5}
```

```
model=tuner.get_best_models(num_models=1)[0]
```

```
model.fit(X_train,y_train,epochs=100,initial_epoch=6,
          validation_data=(X_test,y_test))
```

```
Epoch 7/100
20/20 [==============================] - 2s 22ms/step - loss: 0.4374 - accuracy: 0.7915 - val_loss: 0.4678 - val_accuracy: 0
Epoch 8/100
20/20 [==============================] - 0s 7ms/step - loss: 0.4277 - accuracy: 0.8029 - val_loss: 0.5061 - val_accuracy: 0.
Epoch 9/100
20/20 [==============================] - 0s 8ms/step - loss: 0.4191 - accuracy: 0.7980 - val_loss: 0.4840 - val_accuracy: 0.
Epoch 10/100
20/20 [==============================] - 0s 8ms/step - loss: 0.4101 - accuracy: 0.8111 - val_loss: 0.4667 - val_accuracy: 0.
Epoch 11/100
20/20 [==============================] - 0s 6ms/step - loss: 0.4050 - accuracy: 0.8013 - val_loss: 0.4567 - val_accuracy: 0.
Epoch 12/100
20/20 [==============================] - 0s 6ms/step - loss: 0.3941 - accuracy: 0.8257 - val_loss: 0.4590 - val_accuracy: 0.
Epoch 13/100
20/20 [==============================] - 0s 6ms/step - loss: 0.3849 - accuracy: 0.8241 - val_loss: 0.5230 - val_accuracy: 0.
Epoch 14/100
20/20 [==============================] - 0s 7ms/step - loss: 0.3773 - accuracy: 0.8290 - val_loss: 0.4654 - val_accuracy: 0.
Epoch 15/100
20/20 [==============================] - 0s 6ms/step - loss: 0.3786 - accuracy: 0.8257 - val_loss: 0.4879 - val_accuracy: 0.
Epoch 16/100
20/20 [==============================] - 0s 8ms/step - loss: 0.3625 - accuracy: 0.8453 - val_loss: 0.5304 - val_accuracy: 0.
Epoch 17/100
20/20 [==============================] - 0s 7ms/step - loss: 0.3522 - accuracy: 0.8616 - val_loss: 0.5006 - val_accuracy: 0.
Epoch 18/100
20/20 [==============================] - 0s 7ms/step - loss: 0.3427 - accuracy: 0.8534 - val_loss: 0.5689 - val_accuracy: 0.
Epoch 19/100
20/20 [==============================] - 0s 8ms/step - loss: 0.3213 - accuracy: 0.8697 - val_loss: 0.6126 - val_accuracy: 0.
Epoch 20/100
20/20 [==============================] - 0s 7ms/step - loss: 0.3194 - accuracy: 0.8583 - val_loss: 0.5619 - val_accuracy: 0.
Epoch 21/100
20/20 [==============================] - 0s 7ms/step - loss: 0.3100 - accuracy: 0.8795 - val_loss: 0.5443 - val_accuracy: 0.
Epoch 22/100
```

```
20/20 [==============================] - 0s 9ms/step - loss: 0.3013 - accuracy: 0.8746 - val_loss: 0.5379 - val_accuracy: 0.
Epoch 23/100
20/20 [==============================] - 0s 10ms/step - loss: 0.2769 - accuracy: 0.8876 - val_loss: 0.7885 - val_accuracy: 0
Epoch 24/100
20/20 [==============================] - 0s 4ms/step - loss: 0.2993 - accuracy: 0.8779 - val_loss: 0.6036 - val_accuracy: 0.
Epoch 25/100
20/20 [==============================] - 0s 5ms/step - loss: 0.2536 - accuracy: 0.8974 - val_loss: 0.6204 - val_accuracy: 0.
Epoch 26/100
20/20 [==============================] - 0s 5ms/step - loss: 0.2750 - accuracy: 0.8860 - val_loss: 0.6518 - val_accuracy: 0.
Epoch 27/100
20/20 [==============================] - 0s 4ms/step - loss: 0.2179 - accuracy: 0.9283 - val_loss: 0.6873 - val_accuracy: 0.
Epoch 28/100
20/20 [==============================] - 0s 4ms/step - loss: 0.2385 - accuracy: 0.9104 - val_loss: 0.6403 - val_accuracy: 0.
Epoch 29/100
20/20 [==============================] - 0s 4ms/step - loss: 0.2256 - accuracy: 0.9121 - val_loss: 0.6991 - val_accuracy: 0.
Epoch 30/100
20/20 [==============================] - 0s 4ms/step - loss: 0.1939 - accuracy: 0.9381 - val_loss: 0.7440 - val_accuracy: 0.
Epoch 31/100
20/20 [==============================] - 0s 4ms/step - loss: 0.2254 - accuracy: 0.9202 - val_loss: 0.8054 - val_accuracy: 0.
Epoch 32/100
20/20 [==============================] - 0s 4ms/step - loss: 0.1948 - accuracy: 0.9316 - val_loss: 0.9085 - val_accuracy: 0.
Epoch 33/100
20/20 [==============================] - 0s 4ms/step - loss: 0.1799 - accuracy: 0.9349 - val_loss: 0.7580 - val_accuracy: 0.
Epoch 34/100
20/20 [==============================] - 0s 4ms/step - loss: 0.1668 - accuracy: 0.9463 - val_loss: 0.8760 - val_accuracy: 0.
Epoch 35/100
20/20 [==============================] - 0s 4ms/step - loss: 0.1804 - accuracy: 0.9349 - val_loss: 0.8233 - val_accuracy: 0
```

```
# Above model is highly overfitted if num of layers is taken is 5
# Now build different model with all tuning combined
def build_model(hp):
  model=Sequential()
  counter=0;
  for i in range(hp.Int('num_layers',min_value=1,max_value=10)):
    if counter==0:
      model.add( Dense( hp.Int('units'+str(i),min_value=8,max_value=128,step=8),activation=hp.Choice('activation'+str(i),values=[
    else:
       model.add( Dense(hp.Int('units'+str(i),min_value=8,max_value=128,step=8),activation=hp.Choice('activation'+str(i),values=[
    counter+=1
  model.add(Dense(1,activation='sigmoid'))
  model.compile(optimizer=hp.Choice('optimizer',values=['rmsprop','adam','sgd','nadam','adadelta']),
                loss='binary_crossentropy',
                metrics=['accuracy']
                )
  return  model
```

```
tuner=kt.RandomSearch(build_model,
                      objective='val_accuracy',
                      max_trials=3,
                      directory='mydir',
                      project_name='final_correct3' )
```

```
tuner.search(X_train,y_train,epochs=5,
             validation_data=(X_test,y_test))
```

```
Trial 3 Complete [00h 00m 02s]
val_accuracy: 0.7207792401313782

Best val_accuracy So Far: 0.7922077775001526
Total elapsed time: 00h 00m 10s
```

```
tuner.get_best_hyperparameters()[0].values
```

```
{'num_layers': 8,
 'units0': 128,
 'activation0': 'tanh',
 'optimizer': 'rmsprop',
 'units1': 112,
 'activation1': 'sigmoid',
 'units2': 80,
 'activation2': 'tanh',
 'units3': 56,
 'activation3': 'tanh',
 'units4': 64,
 'activation4': 'relu',
 'units5': 56,
 'activation5': 'relu',
 'units6': 40,
 'activation6': 'relu',
 'units7': 40,
 'activation7': 'relu',
 'units8': 120,
 'activation8': 'sigmoid',
 'units9': 128,
 'activation9': 'tanh'}
```

```
model=tuner.get_best_models(num_models=1)[0]
```

```
model.fit(X_train,y_train,epochs=150,initial_epoch=6,
          validation_data=(X_test,y_test))
```

```
    Epoch 7/150
    20/20 [==============================] - 1s 16ms/step - loss: 0.4907 - accuracy: 0.7590 - val_loss: 0.4818 - val_accuracy: 0
    Epoch 8/150
    20/20 [==============================] - 0s 5ms/step - loss: 0.4855 - accuracy: 0.7704 - val_loss: 0.5059 - val_accuracy: 0.
    Epoch 9/150
    20/20 [==============================] - 0s 5ms/step - loss: 0.4889 - accuracy: 0.7638 - val_loss: 0.4880 - val_accuracy: 0.
    Epoch 10/150
    20/20 [==============================] - 0s 4ms/step - loss: 0.4862 - accuracy: 0.7622 - val_loss: 0.4831 - val_accuracy: 0.
    Epoch 11/150
    20/20 [==============================] - 0s 5ms/step - loss: 0.4806 - accuracy: 0.7704 - val_loss: 0.4898 - val_accuracy: 0.
    Epoch 12/150
    20/20 [==============================] - 0s 5ms/step - loss: 0.4792 - accuracy: 0.7704 - val_loss: 0.4875 - val_accuracy: 0.
    Epoch 13/150
    20/20 [==============================] - 0s 6ms/step - loss: 0.4788 - accuracy: 0.7671 - val_loss: 0.4861 - val_accuracy: 0.
    Epoch 14/150
    20/20 [==============================] - 0s 5ms/step - loss: 0.4718 - accuracy: 0.7785 - val_loss: 0.5002 - val_accuracy: 0.
    Epoch 15/150
    20/20 [==============================] - 0s 4ms/step - loss: 0.4730 - accuracy: 0.7638 - val_loss: 0.4759 - val_accuracy: 0.
    Epoch 16/150
    20/20 [==============================] - 0s 4ms/step - loss: 0.4712 - accuracy: 0.7704 - val_loss: 0.4778 - val_accuracy: 0.
    Epoch 17/150
    20/20 [==============================] - 0s 5ms/step - loss: 0.4682 - accuracy: 0.7720 - val_loss: 0.4938 - val_accuracy: 0.
    Epoch 18/150
    20/20 [==============================] - 0s 5ms/step - loss: 0.4649 - accuracy: 0.7720 - val_loss: 0.4881 - val_accuracy: 0.
    Epoch 19/150
    20/20 [==============================] - 0s 6ms/step - loss: 0.4619 - accuracy: 0.7801 - val_loss: 0.4694 - val_accuracy: 0.
    Epoch 20/150
    20/20 [==============================] - 0s 4ms/step - loss: 0.4713 - accuracy: 0.7785 - val_loss: 0.4710 - val_accuracy: 0.
    Epoch 21/150
    20/20 [==============================] - 0s 4ms/step - loss: 0.4641 - accuracy: 0.7818 - val_loss: 0.4749 - val_accuracy: 0.
    Epoch 22/150
    20/20 [==============================] - 0s 5ms/step - loss: 0.4529 - accuracy: 0.7834 - val_loss: 0.4972 - val_accuracy: 0.
    Epoch 23/150
    20/20 [==============================] - 0s 5ms/step - loss: 0.4600 - accuracy: 0.7720 - val_loss: 0.4972 - val_accuracy: 0.
    Epoch 24/150
    20/20 [==============================] - 0s 5ms/step - loss: 0.4633 - accuracy: 0.7671 - val_loss: 0.5022 - val_accuracy: 0.
    Epoch 25/150
    20/20 [==============================] - 0s 5ms/step - loss: 0.4616 - accuracy: 0.7736 - val_loss: 0.4638 - val_accuracy: 0.
    Epoch 26/150
    20/20 [==============================] - 0s 5ms/step - loss: 0.4493 - accuracy: 0.7752 - val_loss: 0.4893 - val_accuracy: 0.
    Epoch 27/150
    20/20 [==============================] - 0s 6ms/step - loss: 0.4601 - accuracy: 0.7720 - val_loss: 0.4989 - val_accuracy: 0.
    Epoch 28/150
    20/20 [==============================] - 0s 5ms/step - loss: 0.4557 - accuracy: 0.7769 - val_loss: 0.4832 - val_accuracy: 0.
    Epoch 29/150
    20/20 [==============================] - 0s 4ms/step - loss: 0.4583 - accuracy: 0.7818 - val_loss: 0.4729 - val_accuracy: 0.
    Epoch 30/150
    20/20 [==============================] - 0s 4ms/step - loss: 0.4529 - accuracy: 0.7785 - val_loss: 0.4810 - val_accuracy: 0.
    Epoch 31/150
    20/20 [==============================] - 0s 4ms/step - loss: 0.4555 - accuracy: 0.7834 - val_loss: 0.4929 - val_accuracy: 0.
    Epoch 32/150
    20/20 [==============================] - 0s 4ms/step - loss: 0.4517 - accuracy: 0.7704 - val_loss: 0.4972 - val_accuracy: 0.
    Epoch 33/150
    20/20 [==============================] - 0s 5ms/step - loss: 0.4585 - accuracy: 0.7818 - val_loss: 0.4797 - val_accuracy: 0.
    Epoch 34/150
    20/20 [==============================] - 0s 4ms/step - loss: 0.4506 - accuracy: 0.7720 - val_loss: 0.4932 - val_accuracy: 0.
    Epoch 35/150
    20/20 [==============================] - 0s 4ms/step - loss: 0.4534 - accuracy: 0.7834 - val_loss: 0.4880 - val_accuracy: 0.
    Epoch 36/150
```