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

```
df = pd.read_csv('/content/Social_Network_Ads.csv')
```

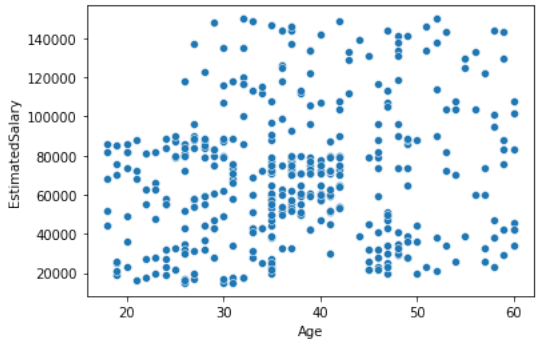
```
df = df.iloc[:,2:]
df.head()
```

	Age	EstimatedSalary	Purchased
0	19	19000	0
1	35	20000	0
2	26	43000	0
3	27	57000	0
4	19	76000	0

```
import seaborn as sns
```

```
sns.scatterplot(df.iloc[:,0],df.iloc[:,1])
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variables as keyword arg
  FutureWarning
<matplotlib.axes._subplots.AxesSubplot at 0x7f5eda7c43d0>
```




```
X = df.iloc[:,0:2]
y = df.iloc[:,-1]
```

```
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=2)
```

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

```
model = Sequential()

model.add(Dense(128,activation='relu',input_dim=2))
model.add(Dense(1,activation='sigmoid'))
```

```
model.summary()
```

```
Model: "sequential_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_1 (Dense)             (None, 128)               384

 dense_2 (Dense)             (None, 1)                 129

=================================================================
Total params: 513
Trainable params: 513
```

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

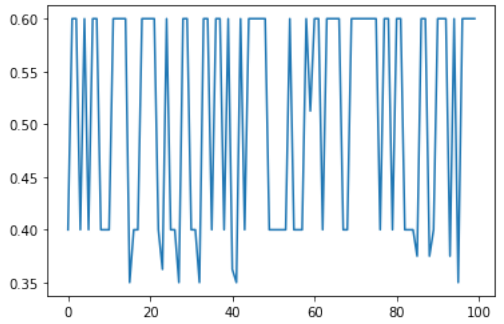
```
model.compile(optimizer='adam',loss='binary_crossentropy',metrics=['accuracy'])
```

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

```
10/10 [==============================] - 0s 9ms/step - loss: 105.6520 - accuracy: 0.5188 - val_loss: 5.7623 - val_accuracy
Epoch 6/100
10/10 [==============================] - 0s 8ms/step - loss: 19.1263 - accuracy: 0.5406 - val_loss: 17.4829 - val_accuracy
Epoch 7/100
10/10 [==============================] - 0s 8ms/step - loss: 20.7027 - accuracy: 0.5312 - val_loss: 33.1314 - val_accuracy
Epoch 8/100
10/10 [==============================] - 0s 8ms/step - loss: 43.7668 - accuracy: 0.4406 - val_loss: 26.5325 - val_accuracy
Epoch 9/100
10/10 [==============================] - 0s 9ms/step - loss: 53.5641 - accuracy: 0.5719 - val_loss: 91.2861 - val_accuracy
Epoch 10/100
10/10 [==============================] - 0s 8ms/step - loss: 87.6180 - accuracy: 0.4969 - val_loss: 126.8609 - val_accurac
Epoch 11/100
10/10 [==============================] - 0s 8ms/step - loss: 83.3905 - accuracy: 0.5094 - val_loss: 112.9638 - val_accurac
Epoch 12/100
10/10 [==============================] - 0s 9ms/step - loss: 54.7899 - accuracy: 0.5094 - val_loss: 10.9042 - val_accuracy
Epoch 13/100
10/10 [==============================] - 0s 9ms/step - loss: 40.3692 - accuracy: 0.5281 - val_loss: 37.2375 - val_accuracy
Epoch 14/100
10/10 [==============================] - 0s 9ms/step - loss: 67.3726 - accuracy: 0.5281 - val_loss: 176.2580 - val_accurac
Epoch 15/100
10/10 [==============================] - 0s 8ms/step - loss: 96.3750 - accuracy: 0.5312 - val_loss: 206.0263 - val_accurac
Epoch 16/100
10/10 [==============================] - 0s 9ms/step - loss: 145.4039 - accuracy: 0.5469 - val_loss: 9.8404 - val_accuracy
Epoch 17/100
10/10 [==============================] - 0s 8ms/step - loss: 121.6079 - accuracy: 0.5125 - val_loss: 114.8439 - val_accura
Epoch 18/100
10/10 [==============================] - 0s 9ms/step - loss: 144.6243 - accuracy: 0.6062 - val_loss: 116.7214 - val_accura
Epoch 19/100
10/10 [==============================] - 0s 8ms/step - loss: 152.1929 - accuracy: 0.4969 - val_loss: 225.4776 - val_accura
Epoch 20/100
10/10 [==============================] - 0s 8ms/step - loss: 131.7005 - accuracy: 0.4844 - val_loss: 193.4258 - val_accura
Epoch 21/100
10/10 [==============================] - 0s 9ms/step - loss: 110.8412 - accuracy: 0.5469 - val_loss: 25.9139 - val_accurac
Epoch 22/100
10/10 [==============================] - 0s 9ms/step - loss: 59.6000 - accuracy: 0.5219 - val_loss: 49.9986 - val_accuracy
Epoch 23/100
10/10 [==============================] - 0s 11ms/step - loss: 58.9580 - accuracy: 0.4688 - val_loss: 15.9721 - val_accurac
Epoch 24/100
10/10 [==============================] - 0s 9ms/step - loss: 44.8860 - accuracy: 0.5188 - val_loss: 7.2842 - val_accuracy:
Epoch 25/100
10/10 [==============================] - 0s 11ms/step - loss: 41.7748 - accuracy: 0.5000 - val_loss: 77.9447 - val_accurac
Epoch 26/100
10/10 [==============================] - 0s 9ms/step - loss: 60.8529 - accuracy: 0.5219 - val_loss: 32.6075 - val_accuracy
Epoch 27/100
10/10 [==============================] - 0s 8ms/step - loss: 23.4272 - accuracy: 0.5469 - val_loss: 20.3319 - val_accuracy
Epoch 28/100
10/10 [==============================] - 0s 8ms/step - loss: 28.0190 - accuracy: 0.5125 - val_loss: 11.2252 - val_accuracy
Epoch 29/100
10/10 [==============================] - 0s 9ms/step - loss: 35.0810 - accuracy: 0.5688 - val_loss: 8.0743 - val_accuracy:
Epoch 30/100
10/10 [==============================] - 0s 9ms/step - loss: 99.1157 - accuracy: 0.4969 - val_loss: 46.1971 - val_accuracy
Epoch 31/100
10/10 [==============================] - 0s 10ms/step - loss: 73.2487 - accuracy: 0.5344 - val_loss: 76.7246 - val_accurac
Epoch 32/100
10/10 [==============================] - 0s 8ms/step - loss: 92.3608 - accuracy: 0.4594 - val_loss: 63.9129 - val_accuracy
Epoch 33/100
10/10 [==============================] - 0s 8ms/step - loss: 33.7578 - accuracy: 0.4656 - val_loss: 10.4733 - val_accuracy
Epoch 34/100
```

```
import matplotlib.pyplot as plt
plt.plot(history.history['val_accuracy'])
```

```
[<matplotlib.lines.Line2D at 0x7f5e6dbec310>]
```



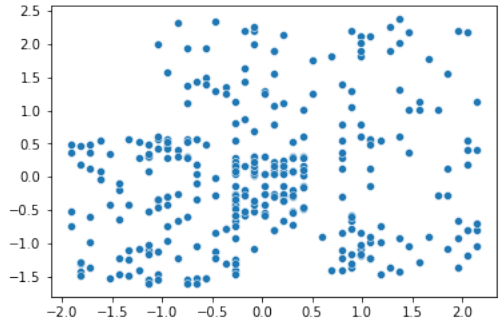

```
# Applying scaling


from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()

X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)


X_train_scaled
```

```
array([[ 0.88928823, -0.65924556],
       [-0.17254846,  0.87392651],
       [-1.04132394, -0.36440478],
       [ 0.98581884,  0.6085698 ],
       [-0.94479333,  0.57908572],
       [ 0.40663519,  0.01888824],
       [ 0.98581884,  2.11225779],
       [ 0.31010458, -0.30543662],
       [ 1.7580637 , -0.27595254],
       [-0.17254846,  2.20071003],
       [ 1.7580637 ,  1.0213469 ],
       [-1.33091576, -1.48479975],
       [ 2.04765553,  0.54960165],
       [ 1.27541066,  1.90586924],
       [-1.13785454,  0.31372902],
       [-0.36560968, -0.77718187],
       [-1.71703819,  0.49063349],
       [-0.5586709 , -1.51428383],
       [ 0.31010458, -0.71821372],
       [ 0.02051275, -0.57079333],
       [ 0.02051275,  0.04837232],
       [-0.07601785, -0.51182517],
       [-0.6552015 , -1.51428383],
       [ 0.02051275,  0.31372902],
       [ 0.31010458,  0.07785639],
       [-0.46214029, -1.13099081],
       [-0.75173211, -1.54376791],
       [-0.26907907, -0.65924556],
       [-1.13785454,  0.49063349],
       [-0.07601785,  2.20071003],
       [ 0.02051275,  0.04837232],
       [-1.13785454, -1.57325199],
       [ 1.08234944,  0.54960165],
       [-0.26907907, -1.24892713],
       [ 1.37194127, -0.92460227],
       [-1.42744637, -1.21944305],
       [-0.94479333, -0.95408634],
       [ 1.95112492, -0.65924556],
       [ 0.88928823, -0.57079333],
       [-1.13785454,  0.31372902],
       [ 0.02051275, -0.24646847],
       [ 0.79275762, -1.39634752],
       [-0.26907907, -0.36440478],
       [ 0.88928823,  1.2867036 ],
       [ 0.31010458, -0.18750031],
       [-0.26907907, -0.57079333],
       [-0.26907907, -1.39634752],
       [ 1.46847188, -1.04253858],
       [-0.07601785,  0.13682455],
       [-0.84826272, -0.65924556],
       [-0.07601785,  0.01888824],
       [-0.26907907,  0.10734047],
       [ 0.21357397, -0.30543662],
       [-0.26907907,  0.28424494],
```

```
[ 0.11704336,  0.04837232],
[ 1.95112492,  2.20071003],
[-1.04132394, -1.45531567],
```

```
sns.scatterplot(X_train_scaled[:,0],X_train_scaled[:,1])
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning:
  FutureWarning
<matplotlib.axes._subplots.AxesSubplot at 0x7f5e6d91d950>
```

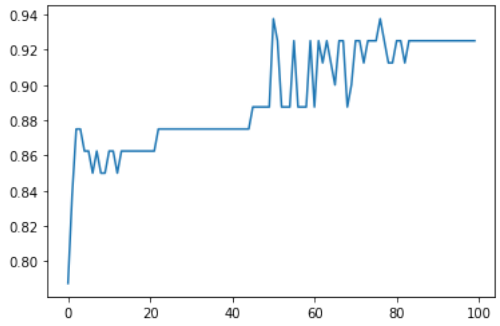



```
model = Sequential()

model.add(Dense(128,activation='relu',input_dim=2))
model.add(Dense(1,activation='sigmoid'))

model.compile(optimizer='adam',loss='binary_crossentropy',metrics=['accuracy'])

history = model.fit(X_train_scaled,y_train,validation_data=(X_test_scaled,y_test),epochs=100)
```

```
Epoch 52/100
10/10 [==============================] - 0s 8ms/step - loss: 0.2392 - accuracy: 0.9062 - val_loss: 0.2631 - val_accuracy:
Epoch 53/100
10/10 [==============================] - 0s 8ms/step - loss: 0.2390 - accuracy: 0.9062 - val_loss: 0.2657 - val_accuracy:
Epoch 54/100
10/10 [==============================] - 0s 19ms/step - loss: 0.2390 - accuracy: 0.9031 - val_loss: 0.2660 - val_accuracy:
Epoch 55/100
10/10 [==============================] - 0s 16ms/step - loss: 0.2384 - accuracy: 0.9062 - val_loss: 0.2635 - val_accuracy:
Epoch 56/100
```

```
import matplotlib.pyplot as plt
plt.plot(history.history['val_accuracy'])
```

```
[<matplotlib.lines.Line2D at 0x7f5e6d9dc610>]
```



Start coding or generate with AI.