```
import numpy as np
```

```
import pandas as pd
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
```

```
data = {
    'feature1': [0.1, 0.2, 0.3, 0.4, 0.5],
    'feature2': [0.5, 0.4, 0.3, 0.2, 0.1],
    'label': [0, 0, 1, 1, 1]
}

df = pd.DataFrame(data)
X = df[['feature1', 'feature2']].values
y = df['label'].values
```

```
model=Sequential()
model.add(Dense(8,input_dim=2,activation="relu"))
model.add(Dense(1,activation="sigmoid"))
```

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

```
model.fit(X,y,epochs=100,batch_size=1,verbose=1)
```

```
Epoch 1/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 9ms/step - accuracy: 0.3556 - loss: 0.6900
Epoch 2/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.6611 - loss: 0.6651
Epoch 3/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.4778 - loss: 0.6906
Epoch 4/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.4778 - loss: 0.6851
Epoch 5/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.4222 - loss: 0.6775
Epoch 6/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.8361 - loss: 0.6738
Epoch 7/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 1.0000 - loss: 0.6622
Epoch 8/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 1.0000 - loss: 0.6745
Epoch 9/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - accuracy: 1.0000 - loss: 0.6691
Epoch 10/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - accuracy: 1.0000 - loss: 0.6647
Epoch 11/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 1.0000 - loss: 0.6689
Epoch 12/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 1.0000 - loss: 0.6584
Epoch 13/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 1.0000 - loss: 0.6531
Epoch 14/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 1.0000 - loss: 0.6567
Epoch 15/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - accuracy: 1.0000 - loss: 0.6510
Epoch 16/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 1.0000 - loss: 0.6577
Epoch 17/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 1.0000 - loss: 0.6466
Epoch 18/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - accuracy: 1.0000 - loss: 0.6567
Epoch 19/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 1.0000 - loss: 0.6514
Epoch 20/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - accuracy: 1.0000 - loss: 0.6451
Epoch 21/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 1.0000 - loss: 0.6307
Epoch 22/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 1.0000 - loss: 0.6261
Epoch 23/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 1.0000 - loss: 0.6371
Epoch 24/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 1.0000 - loss: 0.6342
Epoch 25/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 1.0000 - loss: 0.6284
Epoch 26/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 1.0000 - loss: 0.6474
Epoch 27/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 1.0000 - loss: 0.6408
Epoch 28/100
```

```
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 1.0000 - loss: 0.6419
Epoch 29/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 1.0000 - loss: 0.6171
```

```
test_data = np.array([[0.2, 0.4]])
prediction = model.predict(test_data)
prediction
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 61ms/step
array([[0.49663034]], dtype=float32)
```

```
prediction
```

```
array([[0.49663034]], dtype=float32)
```

```
prediction_label=(prediction>0.5).astype(int)
```

```
prediction_label
```

```
array([[0]])
```

```
# implementation of Single perceptron
import tensorflow as tf
from sklearn.datasets import make_classification
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
```

```
X,y=make_classification(
    n_samples=500,
    n_informative=2,
    n_features=2,
    n_redundant=0,
    n_classes=2,
    random_state=42
)
```

```
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,random_state=42)
```

```
scaler=StandardScaler()
```

```
X_train=scaler.fit_transform(X_train)
X_test=scaler.transform(X_test)
```

```
model=tf.keras.Sequential([tf.keras.layers.Dense(1,activation="sigmoid",input_shape=(2,))])
```

```
/usr/local/lib/python3.12/dist-packages/keras/src/layers/core/dense.py:93: UserWarning: Do not pass an `input_shape`/`input_dim`
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
```

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

```
history=model.fit(X_train,y_train,epochs=100,validation_split=0.1,verbose=0)
```

```
loss, accuracy = model.evaluate(X_test, y_test, verbose=0)
print(f"Test Accuracy: {accuracy:.2f}")
```

```
Test Accuracy: 0.88
```

```
loss
```

```
0.288163959980011
```

Start coding or generate with AI.