

```

from keras.models import Sequential
from keras.layers import Dense
from sklearn.datasets import make_blobs
from sklearn.preprocessing import MinMaxScaler
import numpy as np

# Generate a dataset
X, Y = make_blobs(n_samples=100, centers=2, n_features=2, random_state=1)

# Scale features
scalar = MinMaxScaler()
scalar.fit(X)
X = scalar.transform(X)

# Define the model
model = Sequential()
model.add(Dense(4, input_dim=2, activation='relu')) # First hidden layer
model.add(Dense(4, activation='relu')) # Second hidden layer
model.add(Dense(1, activation='sigmoid')) # Output layer for binary classification

# Compile the model
model.compile(loss='binary_crossentropy', optimizer='adam')

# Train the model
model.fit(X, Y, epochs=500, verbose=0) # Set verbose to 0 to suppress training output

# Generate new data for prediction
Xnew, _ = make_blobs(n_samples=3, centers=2, n_features=2, random_state=1)
Xnew = scalar.transform(Xnew)

# Predict probabilities and classes
Y_pred_prob = model.predict(Xnew)
Yclass = (Y_pred_prob > 0.5).astype(int) # Classify based on probability threshold

# Print results
for i in range(len(Xnew)):
    print("X=%s, Predicted Probability=%s, Predicted Class=%s" % (Xnew[i], Y_pred_prob[i],
Yclass[i]))

```