

```
os [31] model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
[32] X_train.shape
   → (380, 10)
os [33]
       y_train.shape

→ (380, 1)
[34] import numpy as np
        import pandas as pd
        from tensorflow.keras.models import Sequential
        from tensorflow.keras.layers import Dense, Dropout, Flatten
        from tensorflow.keras.optimizers import Adam
        from tensorflow.keras.callbacks import EarlyStopping

  [35] y_train = np.where(y_train == -1, 0, y_train)
        y_test = np.where(y_test == -1, 0, y_test)
os [36] from sklearn.utils.class_weight import compute_class_weight
        import numpy as np
        y_train_flat = y_train.ravel() # แปลงเป็น 1 มิติ
        classes = np.unique(y_train_flat) # ดึงค่าคลาสที่มีอยู่
        # คำนวณค่า class weight
        class_weights = compute_class_weight(class_weight='balanced', classes=classes, y=y_train
        # แปลงเป็น dictionary โดยใช้ enumerate เพื่อจับคู่ index กับ class
        class_weight_dict = {classes[i]: class_weights[i] for i in range(len(classes))}
        print(class_weight_dict) # ดูค่า class weight
   → {0.0: 1.0, 1.0: 1.0}
(50) # สร้างโมเดล
        model = Sequential([
            Flatten(),
            Dense(128, input_dim=10, activation='relu'),
            Dropout(0.35),
            Dense(64, activation='relu'),
            Dropout(0.35),
            Dense(32, activation='relu'),
            Dropout(0.35),
            Dense(16, activation='relu'),
            Dropout(0.3).
            Dense(1, activation='sigmoid')
        optimizer = Adam(learning_rate=0.0001)
        model.compile(optimizer='adam', loss='binary crossentropy', metrics=['accuracy'])
        early_stopping = EarlyStopping(monitor='val_loss', patience=20, restore_best_weights=T
        history = model.fit(X_train, y_train, epochs=100, validation_data=(X_test, y_test), batch_size=64, callbacks=[early_stopping], class_weight=class_weight=
        model.evaluate(X_test, y_test)
        model.save('my_model.keras')
        loss, accuracy = model.evaluate(X_test, y_test)
   <del>_____</del> 6/6 -
                              -- 0s 15ms/step - accuracy: 0.8079 - loss: 0.4168 - val_accuracy ▲
        Epoch 74/100
                               — 0s 14ms/step - accuracy: 0.7844 - loss: 0.4907 - val_accuracy
        Epoch 75/100
        6/6 -
                                - 0s 12ms/step - accuracy: 0.8055 - loss: 0.4428 - val_accuracy
        Epoch 76/100
                                - Os 14ms/step - accuracy: 0.7962 - loss: 0.4341 - val accuracy
        6/6 -
        Epoch 77/100
                                - 0s 18ms/step - accuracy: 0.7921 - loss: 0.4338 - val accuracy
        6/6 -
        Epoch 78/100

    Os 16ms/step - accuracy: 0.7551 - loss: 0.4435 - val accuracy

        6/6 -
        Epoch 79/100
                                - 0s 18ms/step - accuracy: 0.7882 - loss: 0.4293 - val accuracy
        6/6 -
        Epoch 80/100
                                — 0s 18ms/step - accuracy: 0.7927 - loss: 0.4324 - val_accuracy
        Enoch 81/100
                                 0s 15ms/step - accuracy: 0.8034 - loss: 0.4549 - val_accuracy
        Epoch 82/100
                                — 0s 15ms/step - accuracy: 0.8054 - loss: 0.4218 - val_accuracy
        Epoch 83/100
                                - 0s 15ms/step - accuracy: 0.8051 - loss: 0.4334 - val_accuracy
        Epoch 84/100
        6/6
                                - 0s 14ms/step - accuracy: 0.8320 - loss: 0.4086 - val_accuracy
        Epoch 85/100
        6/6 -
                                — 0s 14ms/step - accuracy: 0.8192 - loss: 0.4348 - val_accuracy
        Epoch 86/100
        6/6 -

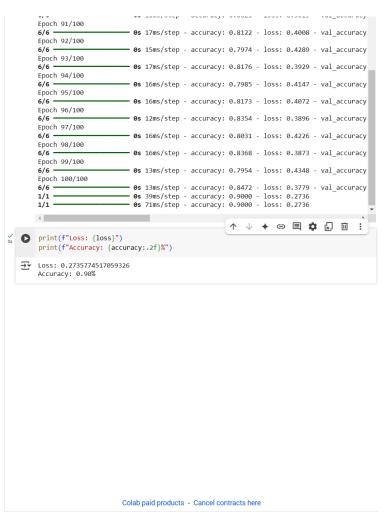
    Os 16ms/step - accuracy: 0.8084 - loss: 0.4266 - val accuracy

        Epoch 87/100

    Os 16ms/step - accuracy: 0.8320 - loss: 0.4501 - val accuracy

        6/6 -
        Epoch 88/100
                                - 0s 18ms/step - accuracy: 0.8303 - loss: 0.3951 - val_accuracy
        6/6 -
        Epoch 89/100
                                 0s 14ms/step - accuracy: 0.8084 - loss: 0.4208 - val accuracy
        Epoch 90/100
```

As 16ms/sten - accuracy: 0.8623 - loss: 0.3619 - val accuracy



✓ 0s completed at 10:44 PM