# Import required libraries

import numpy as np

import matplotlib.pyplot as plt

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import confusion\_matrix, ConfusionMatrixDisplay, classification\_report

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense

# 1. Generate sample binary classification data

from sklearn.datasets import make\_classification

X, y = make\_classification(n\_samples=500, n\_features=10, n\_classes=2, random\_state=42)

# 2. Split the data into train and test

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# 3. Build a simple neural network

model = Sequential([

Dense(16, activation='relu', input\_shape=(X.shape[1],)),

Dense(1, activation='sigmoid') # Binary output

])

# 4. Compile the model

model.compile(optimizer='adam', loss='binary\_crossentropy', metrics=['accuracy'])

# 5. Train the model

history = model.fit(X\_train, y\_train, epochs=20, batch\_size=8, validation\_data=(X\_test, y\_test))

# 6. Evaluate the model

loss, accuracy = model.evaluate(X\_test, y\_test)

print(f"Test Accuracy: {accuracy:.2f}")

# 7. Predict and Confusion Matrix

y\_pred = (model.predict(X\_test) > 0.5).astype(int)

cm = confusion\_matrix(y\_test, y\_pred)

# Display Confusion Matrix

disp = ConfusionMatrixDisplay(confusion\_matrix=cm, display\_labels=["Normal", "Sick"])

disp.plot(cmap=plt.cm.Blues)

plt.title("Confusion Matrix - Binary Classification")

plt.show()

# 8. Print Classification Report

print("Classification Report:\n", classification\_report(y\_test, y\_pred))