# Import necessary libraries

import numpy as np

import pandas as pd

import tensorflow as tf

from tensorflow import keras

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

from sklearn.preprocessing import StandardScaler

from sklearn.metrics import accuracy\_score, precision\_score, recall\_score, f1\_score, confusion\_matrix

# Load and preprocess healthcare data (replace with your dataset)

data = pd.read\_csv('healthcare\_data.csv')

X = data.drop(columns=['target\_column'])

y = data['target\_column']

# Split the data into training and testing sets

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

# Normalize/Standardize the data

scaler = StandardScaler()

X\_train = scaler.fit\_transform(X\_train)

X\_test = scaler.transform(X\_test)

# Define a deep learning model

model = keras.Sequential([

keras.layers.Input(shape=(X\_train.shape[1],)),

keras.layers.Dense(64, activation='relu'),

keras.layers.Dropout(0.2),

keras.layers.Dense(32, activation='relu'),

keras.layers.Dense(1, activation='sigmoid') # For binary classification

])

# Compile the model

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

# Train the model

model.fit(X\_train, y\_train, epochs=10, batch\_size=32, validation\_split=0.2)

# Evaluate the model on the test data

y\_pred = model.predict(X\_test)

y\_pred\_binary = (y\_pred > 0.5).astype(int)

# Calculate evaluation metrics

accuracy = accuracy\_score(y\_test, y\_pred\_binary)

precision = precision\_score(y\_test, y\_pred\_binary)

recall = recall\_score(y\_test, y\_pred\_binary)

f1 = f1\_score(y\_test, y\_pred\_binary)

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

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

print(f"Precision: {precision:.2f}")

print(f"Recall: {recall:.2f}")

print(f"F1 Score: {f1:.2f}")

print("Confusion Matrix:")

print(confusion)