**MODULE 4A ASSIGNMENT**

import numpy as np

import tensorflow as tf

from sklearn import datasets

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

from sklearn.preprocessing import StandardScaler

from sklearn.metrics import accuracy\_score

# Step 1: Load the Iris dataset

iris = datasets.load\_iris()

X = iris.data # Features: sepal length, sepal width, petal length, petal width

y = iris.target # Target: species labels (0, 1, 2)

# Step 2: Select only the first two species (Iris Setosa and Iris Versicolor)

mask = (y == 0) | (y == 1) # Filter for species 0 and 1

X = X[mask]

y = y[mask]

# Step 3: 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)

# Step 4: Normalize the features

scaler = StandardScaler()

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

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

# Step 5: Build the TensorFlow model

model = tf.keras.Sequential([

tf.keras.layers.Dense(8, activation='relu', input\_shape=(4,)), # Input layer with 4 features

tf.keras.layers.Dense(4, activation='relu'), # Hidden layer

tf.keras.layers.Dense(1, activation='sigmoid') # Output layer for binary classification

])

# Compile the model

model.compile(optimizer='adam',

loss='binary\_crossentropy',

metrics=['accuracy'])

# Step 6: Train the model

history = model.fit(X\_train, y\_train, epochs=50, batch\_size=16, validation\_split=0.2, verbose=1)

# Step 7: Evaluate the model on the test set

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

y\_pred\_binary = (y\_pred > 0.5).astype(int).flatten() # Convert probabilities to binary predictions

# Calculate accuracy

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

print(f"Test Accuracy: {accuracy:.4f}")

# Display the model summary

model.summary()

**MODULE 4B ASSIGNMENT**

from sklearn.datasets import load\_breast\_cancer

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

from sklearn.preprocessing import StandardScaler

from sklearn.neural\_network import MLPClassifier

from sklearn.metrics import classification\_report, confusion\_matrix, accuracy\_score

# Step 1: Load the Breast Cancer dataset

data = load\_breast\_cancer()

X = data.data # Features

y = data.target # Target labels

# Step 2: 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)

# Step 3: Standardize the features

scaler = StandardScaler()

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

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

# Step 4: Build the MLP Classifier

# Create an instance of MLPClassifier

mlp\_classifier = MLPClassifier(

hidden\_layer\_sizes=(100, 50), # Number of neurons in each hidden layer

activation='relu', # Activation function

solver='adam', # Optimization algorithm

max\_iter=500, # Maximum number of iterations

random\_state=42 # Random seed for reproducibility

)

# Step 5: Train the model

mlp\_classifier.fit(X\_train, y\_train)

# Step 6: Make predictions on the test set

y\_pred = mlp\_classifier.predict(X\_test)

# Step 7: Evaluate the model

print("Confusion Matrix:")

print(confusion\_matrix(y\_test, y\_pred))

print("\nClassification Report:")

print(classification\_report(y\_test, y\_pred))

print("\nAccuracy Score:")

print(accuracy\_score(y\_test, y\_pred))