import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from scipy import stats

from sklearn.datasets import load\_iris

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

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import accuracy\_score

import tensorflow as tf

# NumPy: Create a 1D array

array = np.array([1, 2, 3, 4, 5])

print("NumPy Array:", array)

# Perform element-wise addition

array = array + 10

print("Modified NumPy Array:", array)

# Pandas: Create a DataFrame

data = {'Name': ['Alice', 'Bob', 'Charlie'], 'Age': [25, 30, 35]}

df = pd.DataFrame(data)

print("\nPandas DataFrame:\n", df)

# Select a column

ages = df['Age']

print("Ages Column:", ages)

# Matplotlib: Create data and plot

x = [1, 2, 3, 4, 5]

y = [2, 3, 5, 7, 11]

plt.plot(x, y)

plt.xlabel('X-axis')

plt.ylabel('Y-axis')

plt.title('Simple Line Plot')

plt.show()

# SciPy: Generate random data and perform a statistical test

data = np.random.normal(0, 1, 1000)

t\_stat, p\_value = stats.ttest\_1samp(data, 0)

print(f"\nSciPy T-statistic: {t\_stat}, P-value: {p\_value}")

# scikit-learn: Load dataset, train a model, and calculate accuracy

iris = load\_iris()

X, y = iris.data, iris.target

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)

clf = RandomForestClassifier()

clf.fit(X\_train, y\_train)

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

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

print(f"\nscikit-learn Accuracy: {accuracy}")

# TensorFlow: Create and compile a simple model

model = tf.keras.Sequential([

tf.keras.layers.Dense(10, activation='relu', input\_shape=(4,)),

tf.keras.layers.Dense(3, activation='softmax')

])

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

model.summary()