**Predictive Analytics**

**NAME - MANAS ARORA**

**B.TECH CSE AIML BATCH 5**

**SAP ID – 500109627**

**ROLL.NO - R2142221204**

**Lab 3:** **Python Library Numpy**

**Q1. Array Creation and Manipulation:**

* **Create different types of arrays (1D, 2D, 3D) using various methods (np.array, np.arange, np.linspace, etc.).**

import numpy as np

# 1D Arrays

array1 = np.array([1, 2, 3, 4, 5])

array2 = np.arange(0, 10, 2) # [0, 2, 4, 6, 8]

array3 = np.linspace(0, 1, 5) # [0. , 0.25, 0.5 , 0.75, 1. ]

# 2D Arrays

array2D\_1 = np.array([[1, 2, 3], [4, 5, 6]])

array2D\_2 = np.zeros((3, 3)) # 3x3 array of zeros

array2D\_3 = np.ones((2, 4)) # 2x4 array of ones

# 3D Arrays

array3D\_1 = np.array([[[1, 2], [3, 4]], [[5, 6], [7, 8]]])

array3D\_2 = np.random.random((2, 2, 3)) # 2x2x3 array with random values

* **Perform basic operations on arrays (indexing, slicing, reshaping, concatenation).**

# Indexing

element = array1[2] # Accesses the 3rd element of 1D array (index starts at 0)

# Slicing

slice\_1D = array1[1:4] # [2, 3, 4]

slice\_2D = array2D\_1[:, 1:3] # Extracts the 2nd and 3rd column of all rows

reshaped = array1.reshape(5, 1) # Converts 1D array to 2D array with 5 rows and 1 column

resized = np.resize(array1, (3, 2)) # Resizes to 3x2 array

# Concatenating 1D arrays

concatenated\_1D = np.concatenate((array1, array2))

# Concatenating 2D arrays along axis 0 (rows) and axis 1 (columns)

concatenated\_2D = np.concatenate((array2D\_1, array2D\_2), axis=0)

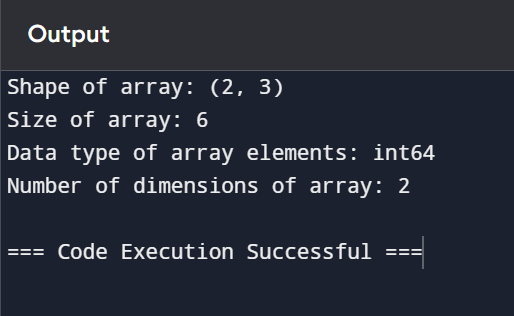
* **Investigate array attributes like shape, size, dtype, ndim.**

print("Shape of array:", array2D\_1.shape)

print("Size of array:", array2D\_1.size)

print("Data type of array elements:", array2D\_1.dtype)

print("Number of dimensions of array:", array2D\_1.ndim)

****

* **Use methods like reshape, resize, flatten,**

flattened = array2D\_1.flatten() # Flattens the 2D array to 1D

reshaped\_array = array2D\_1.reshape(3, 2) # Changes shape to 3x2

Q2. **Data Loading and Preprocessing:**

* **Load a dataset (e.g., CSV, Excel) using NumPy.**

import numpy as np

file\_name = 'machine-readable-business-employment-data-Jun-2024-quarter.csv'

data = np.genfromtxt(file\_name, delimiter=',', skip\_header=1, missing\_values='')

* **Clean and preprocess the data (handling missing values, normalization, standardization).**

# Check the first few rows of the loaded data

print("Loaded Data (first 5 rows):")

print(data[:5])

# Replace missing values (NaN) with the column mean

means = np.nanmean(data, axis=0)

for i in range(data.shape[1]): # Iterate over each column

data[np.isnan(data[:, i]), i] = means[i]

# Normalize (Min-Max Scaling)

data\_min = np.min(data, axis=0)

data\_max = np.max(data, axis=0)

normalized\_data = (data - data\_min) / (data\_max - data\_min)

* **Calculate mean, median, standard deviation, variance, and other statistical measures**.

# Calculate mean, median, standard deviation, and variance

mean = np.mean(data, axis=0)

median = np.median(data, axis=0)

std\_dev = np.std(data, axis=0)

variance = np.var(data, axis=0)

# Output statistical measures

print("Mean:", mean)

print("Median:", median)

print("Standard Deviation:", std\_dev)

print("Variance:", variance)