DAY - 1

Intrduction to Numpy

History

NumPy, short for Numerical Python, was created by Travis Olliphant in 2005 as an open-source library aimed at addressing the limitations of numeric computing in Python. Building on the foundation of Numeric and Numarray, two predecessor libraries, NumPy emerged as a powerful tool for handling large, multi-dimensional arrays and matrices, along with a comprehensive suite of mathematical functions. Its success was instrumental in establishing Python as a prominent language for scientific computing and data analysis. Over the years, NumPy has evolved, with contributions from a vibrant community, and has become a cornerstone in the Python ecosystem, forming the basis for numerous other scientific computing libraries and applications. Its widespread adoption in academia, research, and industry highlights its significance in enabling efficient numerical computations in the Python programming language.

Applications:

1. Scientific Computing
2. Data Analysis
3. Machine Learning
4. Signal Processing
5. Linear Algebra
6. Simulation and Modeling
7. Optimization
8. Visualization
9. Financial and Economic Modeling
10. Educational and Research

Concepts

- ➢ Arrays:
  - ● Description:

    An array is a data structure that stores a collection of elements of the same data type, accessible through indices or keys, with elements residing in contiguous memory locations.

  - ❖ To convert any data type to array:
    - ✓ Definition:

      In NumPy, an array is a multidimensional, homogeneous data structure that efficiently stores and manipulates numerical data for scientific computing in Python.

    - ✓ Syntax:

      ```
      array_name = numpy.array(value_name)
      ```

  - ❖ To append elements:
    - ✓ Definition:

      In NumPy, .append function is used to add elements to end of an array.

    - ✓ Syntax:

      ```
      array_name = numpy.append(array_name,value)
      //For single value
      Array_name = numpy.append(array_name,[values])
      //For multiple values
      ```

  - ❖ To insert elements:
    - ✓ Definition:

      In NumPy, .insert function is used to insert elements at particular index of an array.

- ✓ Syntax:

  array_name = numpy.append(array_name,index,value)
  //For single value
  array_name=numpy.append(array_name,[indices],[values]) //For multiple values

- ❖ To delete elements:
- ✓ Definition:

  In NumPy, delete function is used to delete elements at particular index or a particular value of an array.

- ✓ Syntax:

  array_name=numpy.delete(array_name,numpy.where(array_name==value))
  //For single value
  Array_name = numpy.setdif1d(array_name,[values])
  //For multiple values
  Array_name = numpy.delete(array_name,[values])
  //For multiple values at indices

- ❖ To filter elements:
- ✓ Definition:

  In NumPy, to filter elements we have to give condition.

- ✓ Syntax:

  array_name condition //Return boolean value
  Array_name[condition] //Return original values

- ❖ To modify elements:
- ✓ Definition:

  In NumPy, to modify elements we have to follw below syntax.

- ✓ Syntax:

  array_name[array_name==old_value]=new_vlaue
  //modifyold_value to new_value

- ❖ Zero matrix:
- ✓ Definition:

  In NumPy, to the following syntax will return matrix with elements zero.

- ✓ Syntax:

  numpy.zeros(columns) //Return 1d array
  numpy.zeros(rows,columns) //Return nd array

- ❖ Ones matrix:
- ✓ Definition:

  In NumPy, to the following syntax will return matrix with elements one.

- ✓ Syntax:

  numpy.ones(columns) //Return 1d array
  numpy.ones(rows,columns) //Return nd array

- ❖ Reshape matrix:
- ✓ Definition:

  In NumPy, to change 1d to nd array we use reshape function.

- ✓ Syntax:

  numpy.reshape(rows,columns) //Return nd array

- ❖ Slicing matrix:
- ✓ Definition:

  In NumPy, to get particular values in an array.

- ✓ Syntax:

  array_name[rows,start:end] //Return values under the limit given

- ❖ Mean:
  - ✓ Definition:

    In NumPy, to get mean of an array we use mean function.
  - ✓ Syntax:

    mean(array_name) //Return mean
- ❖ Median :
  - ✓ Definition:

    In NumPy, to get median of an array we use median function.
  - ✓ Syntax:

    numpy.median(array_name) //Return median
- ❖ Sum:
  - ✓ Definition:

    In NumPy, to get sumof an array we use sum function.
  - ✓ Syntax:

    numpy.sum(array_name) //Return sum
- ❖ Variance:
  - ✓ Definition:

    In NumPy, to get variance of an array we use var function.
  - ✓ Syntax:

    numpy.var(array_name) //Return variance
- ❖ Standard deviation:
  - ✓ Definition:

    In NumPy, to get standard deviation of an array we use std function.
- ✓ Syntax:

  numpy.std(array_name) //Return standard deviation

➢ dtype:

- ● Description:

  (data type) in the context of programming, especially in libraries like NumPy, refers to the specification of the type of data stored in an array, indicating whether the elements are integers, floating-point numbers, characters, or other types.
  - ✓ Definition:

    `dtype` in NumPy refers to the data type object, specifying the type of elements stored in a NumPy array, such as integers, floats, or custom data types.
  - ✓ Syntax:

    variable_name.dtype