

MCA SEM-III
2023-2024

DATA SIENCE (Intermediate Level) –
INTRODUCTION TO NUMPY

By,
Dr.Anupriya Kamble,
Senior Faculty-IT
ADYPU, Pune.

INTRODUCTION TO NUMPY

NUMPY BASICS

`pip install numpy`

Numpy Array

Gives efficient storage

Ex. 1, 2, 3 is a list.

Its an object and will store in the memory.

Numpy community made an efficient storage for this python object so that it will save space.

INTRODUCTION TO NUMPY

Features of Numpy

Numpy saves time.

Easy to learn.

Takes relatively less memory to store.

No pointers needed.

INTRODUCTION TO NUMPY

Jupyter

Open source.

Live code.

Best for data analysis.

Supports nearly 40 languages.

Images, videos can be used.

INTRODUCTION TO NUMPY

Starting with Jupyter

Create a folder folder – Shift+Right click – Open Partial Window here.

It opens directory.

Write – jupyter notebook

Starts jupyter notebook

INTRODUCTION TO NUMPY

Starting with Jupyter

In folder create a new text file.

This file will be seen in jupyter notebook.

To create a new notebook using jupyter, click on new on top right of the jupyter notebook and click Python3.

Rename can be done on clicking on the Untitled option.

A cell is given to type the code if more cells are needed + option can be clicked.

INTRODUCTION TO NUMPY

Starting with Numpy

In folder create a new text file.

This file will be seen in jupyter notebook.

To create a new notebook using jupyter, click on new on top right of the jupyter notebook and click Python3.

Rename can be done on clicking on the Untitled option.

A cell is given to type the code if more cells are needed + option can be clicked.

Always run in sequence

To run in sequence in options go to :

Kernel – Restart and Run all

INTRODUCTION TO NUMPY

Starting with Numpy

Options are there to save, add the cell, cut the cell, copy and paste the cell, move the cell up and down.

Kernel can be run, stopped, restart and play from the beginning.

Next option to write a code select “Code”, to write a heading select “Markdown” and write by adding two stars write any text as headline.

Ex.: **Starting with numpy
And press Shift+enter

More effects in Markdown can be found in markdown tutorial.

INTRODUCTION TO NUMPY

Starting with Numpy

In the cell type:

```
import numpy as np  
and press Shift+Enter
```

To print a statement type in the cell:

```
Print("hello world")  
and press shift+enter
```

INTRODUCTION TO NUMPY

Starting with Numpy

In the cell type:
import numpy as np
and press Shift+Enter

To print a statement type in the cell:
Print(“hello world”)
and press shift+enter

INTRODUCTION TO NUMPY

Creating an Array

In any variable store a list to create an array as follows:

```
Ex.: array = np.array([2, 12, 24, 35])
```

In next cell type
array

And press shift + enter.

The entered array will be printed as follows

```
Array([2, 12, 24, 35])
```

INTRODUCTION TO NUMPY

Creating an Array with datatype int8

In any variable store a list to create an array as follows:

Ex.: `array = np.array([2, 12, 24, 35], np.int8)`

In next cell type

`array`

And press shift + enter.

The entered array will be printed as follows

`Array([2, 12, 24, 35], dtype=int8)`

That means 8 bit integer is used. Similarly if big numbers are there then int32, int64. Which manage the memory.

INTRODUCTION TO NUMPY

To access the element in the array

Type the following:

```
array[0]
```

In output we will get the element present in the array at the zeroth position:

Output:

2

INTRODUCTION TO NUMPY

To add two dimensional Array

Simply add two square brackets which will indicate that the array is a two dimensional array:

Ex.: `array = np.array([[2, 12, 24, 35]], np.int8)`

In next cell type “`array[0]`” to print the array at zeroth position

Output:

`array([2, 12, 24, 35], np.int8)`

To print the second number in the list type in the cell:

`array[0,1]`

INTRODUCTION TO NUMPY

To know the shape of the Array

Type in the cell:

`array.shape`

And press shift+enter

Which will indicate how many rows and columns are present in the array.

Output: (1,4)

Which means that the array contains one row and four columns.

INTRODUCTION TO NUMPY

To know the data type of the Array

Type in the cell:

`array.dtype`

And press shift+enter

Which will show which data type has been used.

Output: `dtype('int8')`

Numpy types reference will give you information about the types of the numpy to be used.

INTRODUCTION TO NUMPY

To change the existing element in the Array

Type in the cell:

Array[0, 1] = 30

And press shift+enter

Which will show which data type has been used.

Output: array([2, 30, 24, 35], np.int8)

Numpy types reference will give you information about the types of the numpy to be used.

INTRODUCTION TO NUMPY

Numpy Array Creation

There are 6 general mechanisms for creating arrays:

1. Conversion from other Python structures (i.e. lists and tuples)
2. Intrinsic NumPy array creation functions (e.g. `arange`, `ones`, `zeros`, etc.)
3. Replicating, joining, or mutating existing arrays
4. Reading arrays from disk, either from standard or custom formats
5. Creating arrays from raw bytes through the use of strings or buffers
6. Use of special library functions (e.g., `random`)

Array creation methods in numpy

INTRODUCTION TO NUMPY

Numpy Array Creation

There are 6 general mechanisms for creating arrays:

1. Conversion from other Python structures (i.e. lists and tuples)
2. Intrinsic NumPy array creation functions (e.g. arange, ones, zeros, etc.)
3. Replicating, joining, or mutating existing arrays
4. Reading arrays from disk, either from standard or custom formats
5. Creating arrays from raw bytes through the use of strings or buffers
6. Use of special library functions (e.g., random)

Array creation methods in numpy

INTRODUCTION TO NUMPY

Numpy Array Creation

1. Conversion from other Python structures (i.e. lists and tuples)

listarray = np.array([[1,12,23],[34, 45, 56]])

For output: ***listarray***

O/P : `array([[1,12,23],[34, 45, 56]])`

For output datatype: ***listarray.dtype***

O/P : `dtype('int8')`

For output as shape: ***listarray.shape***

O/P : `(2,3)`

For output for size of array: ***listarray.size***

O/P : `6`

INTRODUCTION TO NUMPY

Numpy Array Creation

1. Conversion from other Python structures (i.e. Dictionaries)

```
dictarray = np.array({7, 8, 9})
```

For output: *dictarray*

O/P : *array*({7,8,9}, dtype=object)

Its always not efficient to make a data type as a object. So generally the array is made up of integer or float. If dealing with efficient calculation then don't deal with objects and deal with integers or floating point architecture for making numpy array.

INTRODUCTION TO NUMPY

Numpy Array Creation

2. Intrinsic NumPy array creation functions (e.g. arange, ones, zeros, etc.)

Numpy provides functions to create arrays.

Numpy array creation function 'zeros'

zerosfunc = np.zeros((1,2))

For output: ***zerosfunc***

O/P : array([0,0]) // It will create a 1x2 array of zeros

For output datatype: ***zerosfunc.dtype***

O/P : dtype('float32')

For output as shape: ***zerosfunc.shape***

O/P : (1,2)

For output for size of array: ***listarray.size***

O/P : 2

INTRODUCTION TO NUMPY

Numpy Array Creation

2. Intrinsic NumPy array creation functions (e.g. arange, ones, zeros, etc.)

Numpy provides functions to create arrays.

Numpy array creation function 'arange'

rngfunc = np.arange(13)

For output: *rngfunc*

O/P : `array([0,1,2,3,4,5,6,7,8,9,10,11,12,13])`

// It will create an array from 0 to n-1.

For output datatype: *zerosfunc.dtype*

O/P : `dtype('float32')`

For output as shape: *zerosfunc.shape*

O/P : `(1,2)`

For output for size of array: *listarray.size*

INTRODUCTION TO NUMPY

Numpy Array Creation

2. Intrinsic NumPy array creation functions (e.g. `arange`, `ones`, `zeros`, etc.)

Numpy provides functions to create arrays.

Numpy array creation function 'linspace'

linspacefunc = np.linspace(1,4,10)

For output: ***linspacefunc***

O/P :

// It will create an array from 1 to 4 equally linearly spaced 10 elements.

For output datatype: ***zerosfunc.dtype***

O/P : `dtype('float32')`

For output as shape: ***zerosfunc.shape***

O/P :

For output for size of array: ***listarray.size***

INTRODUCTION TO NUMPY

Numpy Array Creation

2. Intrinsic NumPy array creation functions (e.g. arange, ones, zeros, etc.)

Numpy provides functions to create arrays.

Numpy array creation function 'empty'

```
empfunc = np.empty((4,6))
```

For output: ***empfunc***

```
O/P :array([0,1,2,3,4,5,6,7,8,9,10,11,12,13])
```

// It will create an empty array of 4,6 and in which all elements will be random and then we can insert values that are needed.

INTRODUCTION TO NUMPY

Numpy Array Creation

2. Intrinsic NumPy array creation functions (e.g. `arange`, `ones`, `zeros`, etc.)

NumPy provides functions to create arrays.

NumPy array creation function 'empty'

`emp_likefunc = np.empty_like(lspace)` // give any array

For output: ***`emp_likefunc`***

O/P :

// It will take the size of the old array and gives an empty array.

// It is for efficiency.

INTRODUCTION TO NUMPY

Numpy Array Creation

2. Intrinsic NumPy array creation functions (e.g. `arange`, `ones`, `zeros`, etc.)

Numpy provides functions to create arrays.

Numpy array creation function 'identity'

identfunc = np.identity(34)//give any array

For output:***identfunc***

O/P :

// It will give an identity matrix whose size is 34x34

// Identity matrix is the one in which all principal elements are 1 rest all elements are zero.

INTRODUCTION TO NUMPY

Numpy Array Creation

2. Intrinsic NumPy array creation functions (e.g. arange, ones, zeros, etc.)

Numpy provides functions to create arrays.

Numpy array creation function 'arange'

arangereshape = np.arange(33)//give any array

For output: *arangereshape*

O/P :

arangereshape.reshape(5, 32)

// It will give an identity matrix whose size is 34x34

// Identity matrix is the one in which all principal elements are 1 rest all elements are zero.

INTRODUCTION TO NUMPY

Numpy Axis

Numpy Axis

For one D [1,2,3] – 1 Axis[0]

For Two D - 2 Axis [1]

Row scanning – Axis 0

Column scanning – Axis 1

```
a=[[1,2,3],[1,0,1],[0,1,2]]
```

```
b=np.array(x)
```

```
b
```

INTRODUCTION TO NUMPY

Numpy Array Creation

```
b.sum(axis=0)
```

```
b
```

```
b.sum(axis=1)
```

Array Transpose:

```
b.T
```

```
Numpy.flatiter
```

```
b.Flat
```

```
For item in b.flat:  
    print(item)
```

INTRODUCTION TO NUMPY

Numpy Array Creation

For Number of dimensions:

b.Ndim //Run as attribute

To know elements in array:

b.Size //Run as attribute

To know Array is taking how much space:

b.Nbytes //Run as attribute

INTRODUCTION TO NUMPY

Numpy Array Creation

Create a one d array:

```
C=[3,2,1]
```

Method:

To find maximum element in the array:

```
c.Argmax()//Run as a function
```

To find the minimum element in the array:

```
c.argmin() //Run as a function
```

Indices of array in sorted manner:

```
c.argsort()
```


INTRODUCTION TO NUMPY

Numpy Array Creation

Apply the above functions for 3D array created.



INTRODUCTION TO NUMPY

Numpy Array Creation

To return contiguous flattened array:

```
b.ravel()
```

//To reshape in the given size:

```
b.reshape((9,1))
```

```
b = np.array
```

//To add two arrays:

```
a + b
```

//To Multiply two arrays:

```
a * b
```

INTRODUCTION TO NUMPY

Numpy Array Creation

To take square root elementwise:

```
np.sqrt(b)
```

//Sum of all elemnts

```
b.sum()
```

//To find maximum of all elements

```
b.max()
```

//To find minimum of all elements

```
b.min()
```

INTRODUCTION TO NUMPY

Numpy Array Creation

//To find element in an array:

```
b.where(b>4)
```

//To check type

```
Type(np.where(b>4))
```

//To know the non-zero elements. It will sum up. It will give a tuple of each axis.

```
np.count_nonzero(b)
```

INTRODUCTION TO NUMPY

Numpy Array Creation

//How numpy uses less space:

Import sys

pythonarray= [1, 2, 3]

nparray= np.array(pythonarray)

//To know size of element in python. Size of an element

sys.getsizeof(1) * len(pythonarray)

//To know the size of element in numpy array.

nparray.itemsize * nparray.size

INTRODUCTION TO NUMPY

Numpy Array Creation

In search Engine type “Numpy Array Methods and Attributes”.

- Official website : <https://docs.scipy.org/doc//numpy-1.10.4/reference/arrays.html>

INTRODUCTION TO NUMPY

ndArray

```
//To create array using ndarray function  
/*Specify:1.shape in the form of tuple 2. data type to be used.  
import numpy as np
```

```
a=np.ndarray(shape=(7), dtype=int)
```

The predefined dynamic variables:

```
print("size:", a.size)  
print("shape:", a.shape)  
print("dimension:", a.ndim)  
Print("Datatype:", a.dtype)
```

INTRODUCTION TO NUMPY

Numpy Array Creation

```
//Read elements in array
```

```
b= arr.size
```

```
print("Enter %d elemets:" %n) // Using the format specifiers concept of  
string, n value will store in %d element.
```

```
for i in range(n):
```

```
    b[i]=int(input())
```

```
Print("Elements:", b)
```


INTRODUCTION TO NUMPY

Data Types for ndarray

//List, Number, String



INTRODUCTION TO NUMPY

Basic Indexing and Slicing

//One dimensional Array

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

Indexing=0,1,2,3,4

Negative indexing=-5,-4,-3,-2,-1

//Two Dimensional Array

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

Indexing=Row Indexing is Zero, Next Row Indexing is One.

B=[[0,1,2,3,4],[0,1,2,3,4]]

INTRODUCTION TO NUMPY

Basic Indexing and Slicing

//Three dimensional Array

```
c=([[1,2,3],[5,4,3],[3,2,1]])
```

0 1 2

Indexing=Row Indexing is Zero, Next Row Indexing is One.

```
B=([[0,1,2,3,4],[0,1,2,3,4]])
```

INTRODUCTION TO NUMPY

Basic Indexing and Slicing

//One dimensional Array

```
import numpy as np
```

```
a=np.array([1,2,3])
```

```
print(a[0])
```

//Get first and second element from the array and add it.

```
print(a[1] + a[2])
```

INTRODUCTION TO NUMPY

Slicing

//Taking elements from one given index to another given index.

Slice is passed as: [start:end]

Step can be defined as: [start:end:step].

If start is not passed then it is considered as 0.

If step is not passed it is considered as 1.

```
import numpy as np
arr = np.array([1, 2, 3, 4])
print(arr[1:3])
```

INTRODUCTION TO NUMPY

Basic Indexing and Slicing

//Three dimensional Array

Access element of second row and first column:

```
import numpy as np
```

```
b=np.array([1,2,3],[3,2,1])
```

```
print(b[2,1])
```

INTRODUCTION TO NUMPY

Reference

1. https://www.w3schools.com/python/numpy/numpy_intro.asp
2. https://www.w3schools.com/python/numpy/numpy_getting_started.asp
3. https://www.w3schools.com/python/numpy/numpy_creating_arrays.asp
4. https://www.w3schools.com/python/numpy/numpy_array_indexing.asp
5. https://www.w3schools.com/python/numpy/numpy_array_slicing.asp
6. https://www.w3schools.com/python/numpy/numpy_data_types.asp
7. https://www.w3schools.com/python/numpy/numpy_array_shape.asp
8. https://www.w3schools.com/python/numpy/numpy_array_reshape.asp

THANK YOU

iNURTURE
Education Solutions
— TOMORROW'S HERE —