Name: Sakshi Ravindra Aher

Roll no: 602 (F-1)

PRN: 202201040089

Sub : EDS

import numpy as np

array1=np.array([[1,2,3],[4,5,6],[7,8,9]])

array1

Output :

array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])

array2=np.array([[11,12,13],[14,15,16],[17,18,19]])

array2

Output :

array([[11, 12, 13], [14, 15, 16], [17, 18, 19]])

#1.Matrix Operations

# 1.addition

resultarray=array1+array2

print("\nusing Operator:\n",resultarray)

resultarray=np.add(array1,array2)

print("\nUsing numpy function:\n",resultarray)

Output:

using Operator:

[[12 14 16]

[18 20 22]

[24 26 28]]

Using numpy function:

[[12 14 16]

[18 20 22]

[24 26 28]]

#1.2subtration:

resultarray=array1-array2

print("\nusing Operator:\n",resultarray)

resultarray=np.subtract(array1,array2)

print("\nUsing numpy function:\n",resultarray)

Output :

using Operator:

[[-10 -10 -10]

[-10 -10 -10]

[-10 -10 -10]]

Using numpy function:

[[-10 -10 -10]

[-10 -10 -10]

[-10 -10 -10]]

 # 1.3.Multiplicatin

resultarray=array1\*array2

print("\nusing Operator:\n",resultarray)

resultarray=np.multiply(array1,array2)

print("\nUsing numpy function:\n",resultarray)

Output :

using Operator:

[[ 21 44 69]

[ 96 125 156]

[189 224 261]]

Using numpy function:

[[ 21 44 69]

[ 96 125 156]

[189 224 261]]

#1.4.Division

resultarray=array1/array2

print("\nusing Operator:\n",resultarray)

resultarray=np.divide(array1,array2)

print("\nUsing numpy function:\n",resultarray)

Output:

using Operator:

[[0.09090909 0.16666667 0.23076923]

[0.28571429 0.33333333 0.375 ]

[0.41176471 0.44444444 0.47368421]]

Using numpy function:

[[0.09090909 0.16666667 0.23076923]

[0.28571429 0.33333333 0.375 ]

[0.41176471 0.44444444 0.47368421]]

#1.5.mod

resultarray=array1%array2

print("\nusing Operator:\n",resultarray)

resultarray=np.mod(array1,array2)

print("\nUsing numpy function:\n",resultarray)

Output:

using Operator:

[[1 2 3]

[4 5 6]

[7 8 9]]

Using numpy function:

[[1 2 3]

[4 5 6]

[7 8 9]]

#1.6.dot product

resultarray=np.dot(array1,array2)

print("",resultarray)

output:

[[ 90 96 102]

[216 231 246]

[342 366 390]]

#1.7.transpose

resultarray=np.transpose(array1)

print(resultarray)

#or

resultarray=array1.transpose

print("",resultarray)

output :

[[1 4 7]

[2 5 8]

#2.horizontal and vertical stacking of numpy Arrays

#2.1 verical stacking

resultarray=np.hstack((array1,array2))

resultarray

output:

array([[ 1, 2, 3, 11, 12, 13], [ 4, 5, 6, 14, 15, 16], [ 7, 8, 9, 17, 18, 19]])

#2.2 Vertical stacking

resultarray=np.vstack((array1,array2))

resultarray

output:

array([[ 1, 2, 3], [ 4, 5, 6], [ 7, 8, 9], [11, 12, 13], [14, 15, 16], [17, 18, 19]])

#3.Custom sequence generation

#3.1.Range

nparray=np.arange(0,12,1).reshape(3,4)

nparray

Output :

array([[ 0, 1, 2, 3],

[ 4, 5, 6, 7],

[ 8, 9, 10, 11]])

0s

#3.2.linearly Separable

nparray=np.linspace(start=0,stop=24,num=12).reshape(3,4)

nparray

Output:

array([[ 0. , 2.18181818, 4.36363636, 6.54545455],

[ 8.72727273, 10.90909091, 13.09090909, 15.27272727],

[17.45454545, 19.63636364, 21.81818182, 24. ]])

#3.3.Empty Array

nparray=np.empty((3,3),int)

nparray

Output:

array([[ 21, 44, 69], [ 96, 125, 156], [189, 224, 261]])

#3.4. Empty like Some other array

nparray=np.empty\_like(array1)

nparray

Output

array([[1, 2, 3],

[4, 5, 6],

[7, 8, 9]])

[ ]

#3.5.Identity Matrix

nparray=np.identity(3)

nparray

Output:

array([[1., 0., 0.],

[0., 1., 0.],

[0., 0., 1.]])

#4.Arithmatic Operation

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

array2=np.array([11,12,13,14,15])

print(array1)

print(array2)

output :

[1 2 3 4 5]

[11 12 13 14 15]

#Additon

print(np.add(array1,array2))

#Subtraction

print(np.subtract(array1,array2))

#Multiplication

print(np.multiply(array1,array2))

#Division

print(np.divide(array1,array2))

Output:

[[22 24 26]

[28 30 32]

[34 36 38]]

[[-20 -20 -20]

[-20 -20 -20]

[-20 -20 -20]]

[[ 21 44 69]

[ 96 125 156]

[189 224 261]]

[[0.04761905 0.09090909 0.13043478]

[0.16666667 0.2 0.23076923]

[0.25925926 0.28571429 0.31034483]]

#4.2.Statical and Mathematical Operation

#standard Deviation

print(np.std(array1))

#MInimum

print(np.min(array1))

#Summation

print(np.sum(array1))

#median

print(np.median(array1))

#mean

print(np.mean(array1))

#Mode

from scipy import stats

print("most frequent element=",stats.mode(array1)[0])

print("number of occarances=",stats.mode(array1)[1])

#Varience

print(np.var(array1))

Output:

1.4142135623730951

1

15

3.0

3.0

most frequent element= [1]

number of occarances= [1]

2.0

#4.2.Bitwise Operations

array1=np.array([1,2,3],dtype=np.uint8)

array2=np.array([4,5,6])

#AND

resultarray=np.bitwise\_and(array1,array2)

print(resultarray)

#OR

resultarray=np.bitwise\_or(array1,array2)

print(resultarray)

#Leftship

resultarray=np.left\_shift(array1,array2)

print(resultarray)

#RightShift

resultarray=np.right\_shift(array1,array2)

print(resultarray)

Output :

[0 0 2]

[5 7 7]

[ 16 64 192]

[0 0 0]

#You can get Binary Representation Of Number #####

print(np.binary\_repr(10,8))

resultarray=np.left\_shift(10,2)

print(resultarray)

print(np.binary\_repr(np.left\_shift(10,2),0))

Output:

00001010

40

101000

#5.Copying and viewing arrays.

#5.1.Coping

array1=np.arange(1,10)

print(array1)

newarray=array1.copy()

print(newarray)

##modification in original Array

array1[0]=100

print(array1)

print(newarray)

Output:

[1 2 3 4 5 6 7 8 9]

[1 2 3 4 5 6 7 8 9]

[100 2 3 4 5 6 7 8 9]

[1 2 3 4 5 6 7 8 9]

#5.2.View

array1=np.arange(1,10)

print(array1)

newarray=array1.view()

print(newarray)

##modification in original Array

array1[0]=100

print(array1)

print(newarray)

Output:

[1 2 3 4 5 6 7 8 9]

[1 2 3 4 5 6 7 8 9]

[100 2 3 4 5 6 7 8 9]

[100 2 3 4 5 6 7 8 9]

#6.Sorting

array1=np.array([[1,2,3,12,9,7],[94,5,6,7,89,44],[7,8,9,11,13,14]])

print(array1)

Output:

[[ 1 2 3 12 9 7]

[94 5 6 7 89 44]

[ 7 8 9 11 13 14]]

np.sort(array1,axis=0)#Horizontally sort

Output :

array([[ 1, 2, 3, 7, 9, 7], [ 7, 5, 6, 11, 13, 14], [94, 8, 9, 12, 89, 44]])

np.sort(array1,axis=1)#vertically sort

Output :

array([[ 1, 2, 3, 7, 9, 12], [ 5, 6, 7, 44, 89, 94], [ 7, 8, 9, 11, 13, 14]])

#7.Searching

array1=np.array([1,2,3,12,5,7])

np.searchsorted(array1,7,side="left") #perform search after sorting

Output :3

#8.Counting

array1=np.array([1,2,3,12,5,7,0])

print(np.count\_nonzero(array1))#return total non zero elements

print(np.nonzero(array1))#Return index

print(array1.size)#Total Elements

Output:

6

(array([0, 1, 2, 3, 4, 5]),)

7

#9.Data stacking

array1=np.array(np.arange(1,3).reshape(2,2))

print(array1)

array2=np.array(np.arange(11,15).reshape(2,2))

print(array2)

Output :

[[1 2]

[3 4]]

[[11 12

13 14]]

newarray=np.stack([array1,array2],axis=1)

print(newarray)

Output :

[[[1 2]

[3 4]

[11 12]

[13 14]]]

newarray=np.stack([array1,array2],axis=0)

print(newarray)

Output :

[[[1 2]

[11 12]]

[[3 4]

[13 14]]]

#10.Append

array1=np.arange(1,10).reshape(3,3)

print(array1)

array2=np.arange(21,30).reshape(3,3)

print(array2)

Output :

[[1 2 3]

[4 5 6]

[7 8 9]]

[[21 22 23]

[24 25 26]

[27 28 29]]