NAME : Sakshi Satpute

DIV : G4

ROLL NO. : 781

PRN : 202201040159

Problem Statement : **Prepare/Take datasets for any real-life application. Read a dataset into an array. Perform the following operations on it:**

1. **Perform all matrix operations**
2. **Horizontal and vertical stacking of Numpy Arrays**
3. **Custom sequence generation**
4. **Arithmetic and Statistical Operations, Mathematical Operations, Bitwise Operators**
5. **Copying and viewing arrays**
6. **Data Stacking, Searching, Sorting, Counting, Broadcasting**

CODE :

```
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]5[14,15,16],[17,18,19]1)
```

CODE :

```
array2-np.array([[11,12,13]5[14,15,16],[17,18,19]])
array2
```

OUTPUT :

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

1. Matrix Operations

1.1 Addition

CODE :

```
resultarray=array1 + array2
print("Inusing Operator: \n",resultarray)
resultarray=np.add(array1,array2)
print("\nusing Numpy Function:\n",resultarray)
```

OUTPUT :

```
Using Operator:
[[12 14 16]
T18 20 22]
[2426 28]]
Using Numpy Function:
[[12 14 16]
[18 20 22]
[24 26 28]]
```

1.2. Subtraction

CODE :

```
resultarray=array1-array2
print("Inusing Operator:\n",resultarray)
resultarray-np.subtract(array1,array2)
print("Inusing Numpy Function:\n",resultarray)
```

OUTPUT :

```
using Operator:
[[-10-10 -10]
[-10 -10 -10]
[-10-10 -10]]
Using Numpy Function:
[[-19 -10-10]
[-10-10 -10]
[-10 -10 -10]]
```

CODE :

1.3. Multiplication

```
[] resultarray=array1*array2
print("Inusing Operator: \n",resultarray)
resultarray-np.multiply(array1,array2)
print("Inusing Numpy Function:\n",resultarray)
```

OUTPUT :

```
using Operator:
[[1124 39]
[56 75 96]
[119 144 171]]
using Numpy Function:
[[1124 39]
[56 75 96]

[119 144 171]]
```

1.4. Division

CODE :

```
resultarray=array1/array2

print("Inusing operator: \n",resultarray)

resultarray-np.divide(array1,array?)

print("InUsing Numpy Function:\n",resultarray)
```

OUTPUT :

```
using Operator:

[[0.09090909 0.16666667 .23076923]

[0.28571429 0.33333333 0.375

[0.41176471 0.44444444 0.47368421]]

Snipping Tool

Using Numpy Function:

[[0.09090909 0.16666667 0.23076923]

[0.28571429 0.33333333 0.375]
```

CODE :

1.5. Mod

CODE :

```
resultarray-array1%array?
```

```
print("Inusing Operator:\n",resultarray)

resultarray-np.mod(arrayl,array2)

print("InUsing Numpy Function: \n",resultarray)
```

OUTPUT :

```
using Operator:

[1 2 3]

[4 5 6]

[78 9]]

using Numpy Function:

[[12 3]

[45 6]

[7 8 9]
```

1.6. dot Product

CODE :

```
Resultarray=np.dot(array1,array2)

print(",resultarray)
```

OUTPUT :

```
[[98 9 102]

[216 231 246]

[342 366 390]]
```

1.7. Transpose

```
resultarray -np.transpose(array1)
```

print(resultarray)

#or

resultarray-array1.transpose()

print(resultarray)

Horizontal and vertical stacking of Numpy Arays

()

2.1.Horizontal Stacking

CODE :

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

Resultarray

OUTPUT :

Array

([[1, 2 3, 11, 12, 13]])

[6, 14、 15, 16],

[17, 18, 19]

2.2. Vertical Stacking

CODE :

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

resultarray

OUTPUT :

array([[1, 2, 3],

[11, 12, 13],

[14, 15,16],

[17, 18, 19]])

3.Custom sequence generation

CODE :

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

OUTPUT :

```
array([[ 0, 1, 29, 10, 11]])
```

3.2. Linearly Separable

CODE :

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

OUTPUT :

```
array([[0.2.18181818, .36363636, .54545455],
8.72727273, 10.90909091, 13.09090909, 15.27272727],
[17.45454545, 19.63636364, 21.81818182, 24. ]])
```

3.3. Empty Array

CODE :

```
1 nparray-np.empty((3,3),int)
Nparray
```

OUTPUT :

```
array([[ 11, 24, 39],
[56, 75, 96],
[119, 144, 171]])
```

3.4. Emply Like Some other array

CODE :

```
 nparray-np.empty_like(array1)
Nparray
```

OUTPUT :

```
array([[90, 96, 102],
[216, 231, 246],
[342, 366, 390]])
```

3.5. Identity Matrix

CODE :

```
[] nparray-np.identity(3)
Nparray
```

OPTPUT :

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

4. Arithmetic and Statistical Operations, Mathematical Operations, Bitwise Operators

4.1. Arithmetic Operation

CODE :

```
arrayl-np.array([1,2,3,4,5])
array2-np.array([11,12,13,14,15])
print(array])
print(array?)
```

OUTPUT :

```
[1 2345]
[11 12 13 14 15]
```

CODE :

```
#Addition
print(np.add(array1,ərray2))
# Subtraction
print(np.subtract(array1,array2))
# Multiplication
print(np.multiply(arrayl,array2))
# Division
print(np.divide(array1,array2))
```

Statistical and Mathematical Operations

CODE :

```
Array=np.array([1,2,3,4,5,9,6,7,8,9,9])
# standard Deviation
print(np.std(array1))
#Mininum
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("Host Frequent element=",stats.mode(array1)[0])
print("Number of occarances=",stats.mode (array1)[1])

# variance

print(np.var(array1))
```

OUTPUT :

```
2.7990553306073913

1

63

6.6

5.72777NTDTS

Most Frequent element- [9]

Number of Occarances- [3]
```

4.3. Bitwise Operations

CODE :

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

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

resultarray=np.bitwise_ and(array1,array2)

print(resultarray)

OR

#resultarray=np.bitwise_ or(arrayl,array2)

print(resultarray)

#Leftshift

resultarray-np.left_shift(arrayl,2)

print(resultarray)

#Rightshift

resultarray-np.right_shift(array1,2)

print(resultarray)
```

OUTPUT :

```
2]

[5 7 7 8 12]
```

```
# You can get Binary Representation of Number #asau#
print(np.binary_ repr(10,8))
resultarray=mp.left_ shift(10,2)
print(resultarray)
print(np.binary_repr(np.left_shift(10,2),8))
00001910
40
00101000
SR
```

5.Copying and viewing arrays

Cope

```
 array1=np.arange(1,10)
print(array1)
newarray=array1.copy()
print(newarray)
"wmodification in original Array
array1[0]-100
print (array1)
print(newarray)
output
[123456789]
[12345678
4
3
```

[100

S

(1234 6789]

78

9

5.2 View

arrayl-np.arange(1,10)

print(array1)

newarray-array1.view()

print(newarray)

#rmodification in Original Array

array1[0]-100

print(array1)

print(ncwarray)

sanneriitvera

Snipping Tool

[12345 67891

[12345 G 7

9]

[100

Screenshot copied to clipboard and saved

Select here to mark up and share the image