Python Practical 5

Name : Gaurav singh

Roll.no : 70

Section : c

Batch : C4

python_numpython practical 5

[2]:
```
import numpy
```

1 1. Write a python code for addition, subtraction and multipli- cation of two 4x4 matrices.

[17]:
```
import numpy as np

A = np.array([[1, 2,3,4 ], [3, 4,2,1],[2,1,4,3] ,[4,3,2,1]])
B = np.array([[4,1,2,3],[6,1,2,7],[5,4,2,1],[5,3,1,4]])

print("Printing elements of first matrix") print(A)
print("Printing elements of second matrix") print(B)

print("Addition of two matrix:") print(np.add(A,
B)) print("Subtraction of two matrix")
print(np.subtract(A, B)) print("Multiplication:",
np.dot(A,B))
```

```
 [2 1 4 3]
 [4 3 2 1]]
Printing elements of second matrix [[4 1 2
3]
 [6 1 2 7]
 [5 4 2 1]
 [5 3 1 4]]
Addition of two matrix: [[5 3
5 7]
 [9 5 4 8]
 [7 5 6 4]
 [9 6 3 5]]
Subtraction of two matrix
```

```
 [[-3   1   1   1]
  [-3   3   0  -6]
  [-3  -3   2   2]
  [-1   0   1  -3]]
Multiplication: [[51 27 16 36]
 [51 18 19 43]
 [49 28 17 29]
 [49 18 19 39]]
```

2 2. Create a 5 by 2 integer array from a range between 100 to 200 such that the difference between each element is 10. Print the same.

[8]:

```
import numpy

sampleArray = numpy.arange(100, 200, 10)
sampleArray = sampleArray.reshape(5,2) print
(sampleArray)
```

```
 [120 130]
 [140 150]
 [160 170]
 [180 190]]
```

3 3. Consider two matrices M1=([[2,3,4], [6,5,2], [6,7,3]]) M2=([[1,4,2], [4,3,6],[5,9,8]]) Calculate manually as well as de- velop the python program for the following: (1) matrix multi- plication (dot product) (2) inner product (3) cross product (4) outer product.

[23]:

```
import numpy as np

M1=([[2,3,4], [6,5,2], [6,7,3]])
M2=([[1,4,2], [4,3,6],[5,9,8]])


res = np.dot(M1, M2) print("1.Dot
Product is:") print(res)
print("2.Inner Product is:")
print(np.inner(M1, M2)) print("3.Cross
Product is:") print(np.cross(M1, M2))
print("4.Outer Product is:")
```

```
print(np.outer(M1, M2))
```

```
1. Dot Product is:
[[34 53 54]
 [36 57 58]
 [49 72 78]]
2. Inner Product is:
[[ 22  41  69]
 [ 30  51  91]
 [ 40  63 117]]
3. Cross Product is:
[[-10   0   5]
 [ 24 -28  -2]
 [ 29 -33  19]]
4. Outer Product is:
[[ 2  8  4  8  6 12 10 18 16]
 [ 3 12  6 12  9 18 15 27 24]
 [ 4 16  8 16 12 24 20 36 32]
 [ 6 24 12 24 18 36 30 54 48]
 [ 5 20 10 20 15 30 25 45 40]
 [ 2  8  4  8  6 12 10 18 16]
 [ 6 24 12 24 18 36 30 54 48]
 [ 7 28 14 28 21 42 35 63 56]
 [ 3 12  6 12  9 18 15 27 24]]
```

4 4. Randomly generate the marks of the 80 students in the range of 40 to 95. Write a NumPy program to compute the 70 per- centile for all elements in a given array.

(Hint:use np.random. randint(start,stop,no_of_items) for list generation) (Hint: Use np.percentile)

[24]:
```
import numpy as np

s= np.random.randint(40,95,80)
print("\nOriginal array:") print(s)

t = np.percentile(s, 70)
print("\nThe 70 percentile for allelements:") print(t)
```

```
Original array:
[66 61 68 81 84 58 64 51 56 41 89 49 46 78 78 90 57 80 44 65 69 81 6567
 82 40 58 63 61 89 60 69 94 60 72 53 64 62 60 62 57 85 79 40 82 85 7042
 74 81 78 81 65 60 90 43 59 69 88 83 49 70 61 80 51 92 72 49 88 53 7475
```

60 54 75 69 84 52 86 64]

The 70 percentile for all elements:
78.0

5 5. Write a NumPy program to compute the eigenvalues and right eigenvectors of a given square array. Arr =([[1,4,8], [8,9,2],[9,7,8]]) HINT: Use linear algebra package under numpy. Add these statements i.e. for eigenvalues, import numpy as np from numpy import linalg np.linalg.eig(a)

[25]:
```
from numpy import linalg
Arr =([[1,4,8], [8,9,2],[9,7,8]])
val,vec = np.linalg.eig(Arr)
print("The eigenvalues of the given array are\n",val) print("The right
eigenvectors of the given arrayare\n",vec)
```

```
The eigenvalues of the given arrayare
 [18.61713304 -4.50451282  3.88737977]
The right eigenvectors of the given arrayare [[-
 0.44798294 -0.82768979  0.36471887]
 [-0.52339057  0.43844259 -0.77347804]
 [-0.7248266   0.35028233  0.51837426]]
```

[ ]: