MACHINE LEARNING

ICP_3

Name: NISHANTH KURUGUNTLA

Student id:700766566

GITHUB: https://github.com/nishath0708/ML-ASS-3

VIDEO: https://drive.google.com/file/d/1ea3nqinOS1iCXt117Tbykcyd5hjkzMAp/view?usp=sharing

In [1]:
```
#1. Numpy:
# Using NumPy create random vector of size 15 having only Integers in the range 1-20.
import numpy as np
x = np.random.randint(1,20, size = 15) print
(x)
```

```
[ 6  2 17  5  3  6 13 13  5  8 17 19  7 17  9]
```

In [14]:
```
# 1. Reshape the array to 3 by 5
y=x.reshape(3,5) print(y)
```

```
[[ 6  2 17  5  3]
 [ 6 13 13  5  8]
 [17 19  7 17  9]]
```

In [17]:
```
# 2. Print array shape.
print("array is :",y)
print ("array shape is:",y.shape)
```

```
array is : [[ 6  2 17  5  3]
 [ 6 13 13  5  8]
 [17 19  7 17  9]]
array shape is: (3, 5)
```

In [5]:
```
# 3. Replace the max in each row by 0
new_a = np.where(y == [ [i]
    for i in np.amax(y, axis = 1)
], 0, y)

print(new_a)
```

```
[[ 6  2  0  5  3]
 [ 6  0  0  5  8]
```

```
[17  0  7 17  9]]
```

In [8]:
```python
# Create a 2-dimensional array of size 4 x 3 (composed of 4-byte integer elements), also print the
#of the array.
import numpy as np

# create a 2-dimensional array of size 4x3
arr = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9], [10, 11, 12]], dtype=np.int32)

# print the array shape
print("Array shape:", arr.shape)

# print the array type
print("Array type:", type(arr))

# print the array data type
print("Array data type:", arr.dtype)
```

```
Array shape: (4, 3)
Array type: <class 'numpy.ndarray'> Array
data type: int32
```

In [9]:
```python
#1(b) Write a program to compute the eigenvalues and right eigenvectors
import numpy as np

# define the square array
A = np.array([[3, -2], [1, 0]])

# compute the eigenvalues and right eigenvectors
eigenvalues, eigenvectors = np.linalg.eig(A)

# print the eigenvalues and right eigenvectors
print("Eigenvalues:", eigenvalues) print("Right
eigenvectors:") print(eigenvectors)
```

```
Eigenvalues: [2. 1.] Right
eigenvectors: [[0.89442719
0.70710678]
 [0.4472136  0.70710678]]
```

In [10]:
```python
#1(c)Compute the sum of the diagonal element of a given array.
import numpy as np

# define the array
A = np.array([[0, 1, 2], [3, 4, 5]])

# compute the sum of the diagonal elements
diagonal_sum = np.trace(A)

# print the sum of the diagonal elements
print("Sum of diagonal elements:", diagonal_sum)
```

```
Sum of diagonal elements: 4
```

In [11]:

```
#1(d)Write a NumPy program to create a new shape to an array without changing its data.
import numpy as np

# define the original array
arr = np.array([[1, 2], [3, 4], [5, 6]])

# reshape to 3x2
arr_3x2 = arr.reshape(3, 2)

# reshape to 2x3
arr_2x3 = arr.reshape(2, 3)

print("Reshaped to 3x2:¥n", arr_3x2) print("Reshaped
to 2x3:¥n", arr_2x3)
```

```
Reshaped to 3x2: [[1
 2]
 [3 4]
 [5 6]]
Reshaped to
2x3:
 [[1 2 3]
 [4 5 6]]
```