MACHINE LEARNING (IP#3)

STUDENT ID: 700766162

NAME: SRAVANI MANNURU

Github link: https://github.com/sravs2031/Machine-learning-Assignment-3.git

Video link: https://drive.google.com/file/d/1jm-uy-ms5lpfCCR5eA85LKQf_qsb70jC/view?usp=drive_link

Screenshots:

a)

In [3]:
```
import numpy as np
vector = np.random.randint(1, 21, size=15)
matrix = vector.reshape(3, 5)
print("Array shape:", matrix.shape)
for i in range(matrix.shape[0]):
    max_index = np.argmax(matrix[i])
    matrix[i, max_index] = 0
print("Final matrix:\n", matrix)
print("\nArray shape:", matrix.shape)
print("Array type:", type(matrix))
print("Array data type:", matrix.dtype)
```

```
Array shape: (3, 5)
Final matrix:
 [[ 0 13  6  2  4]
 [ 0  5  5 10 13]
 [ 8  4  6  0  7]]

Array shape: (3, 5)
Array type: <class 'numpy.ndarray'>
Array data type: int32
```

b)

In [4]:
```
import numpy as np

array_2d = np.zeros((4, 3), dtype=np.int32)

square_array = np.array([[3, -2], [1, 0]])
eigenvalues, eigenvectors = np.linalg.eig(square_array)
print("Eigenvalues:", eigenvalues)
print("Right eigenvectors:\n", eigenvectors)
```

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

c)

In [5]:
```
import numpy as np

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


# Compute the sum of the diagonal elements
diagonal_sum = np.trace(array)

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

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

d)

In [6]:
```python
import numpy as np

# Define the arrays
array1 = np.array([[1, 2], [3, 4], [5, 6]])  # 3x2 array
array2 = np.array([[1, 2, 3], [4, 5, 6]])    # 2x3 array
# Reshape array1 to 2x3 without changing its data
reshaped_array1 = np.reshape(array1, (2, 3))

# Reshape array2 to 3x2 without changing its data
reshaped_array2 = np.reshape(array2, (3, 2))

# Print the reshaped arrays
print("Reshaped array1 (2x3):")
print(reshaped_array1)
print("\nReshaped array2 (3x2):")
print(reshaped_array2)
```

```
Reshaped array1 (2x3):
[[1 2 3]
 [4 5 6]]


Reshaped array2 (3x2):
[[1 2]
 [3 4]
 [5 6]]
```