

Machine Learning

ICP-3

Name: Saideep Reddy V

700#: 700755895

Question a:

```
import numpy as np
random_vector = np.random.randint(1, 21, size=15)
print("Random vector:", random_vector)
reshaped_array = random_vector.reshape(3, 5)
print("Reshaped array:\n", reshaped_array)
print("Array shape:", reshaped_array.shape)
for i in range(reshaped_array.shape[0]):
    max_index = np.argmax(reshaped_array[i])
    reshaped_array[i, max_index] = 0
print("Array after replacing max in each row by 0:\n", reshaped_array)
array_2d = np.zeros((4, 3), dtype=np.int32)
print("2D array:\n", array_2d)
print("Shape of the array:", array_2d.shape)
print("Type of the array:", type(array_2d))
print("Data type of the array:", array_2d.dtype)
```

Random vector: [4 18 20 15 10 4 7 15 11 4 8 18 20 11 8]

Reshaped array:

```
[[ 4 18 20 15 10]
 [ 4  7 15 11  4]
 [ 8 18 20 11  8]]
```

Array shape: (3, 5)

Array after replacing max in each row by 0:

```
[[ 4 18  0 15 10]
 [ 4  7  0 11  4]
 [ 8 18  0 11  8]]
```

2D array:

```
[[0 0 0]
 [0 0 0]
 [0 0 0]
 [0 0 0]]
```

Shape of the array: (4, 3)

Type of the array: <class 'numpy.ndarray'>

Data type of the array: int32

Explanation:

The output will include the original random vector, reshaped array, array shape, modified array, and details of the 2D array for (shape, type, data type).

Question b:

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

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

Explanation:

First, The given square array is defined using `np.array`. and Then function `np.linalg.eig` computes the eigenvalues and right eigenvectors of the array. Printing the eigenvalues and eigenvectors are printed.

Question c:

```
[4] import numpy as np
    array = np.array([[0, 1, 2], [3, 4, 5]])
    diagonal_sum = np.trace(array)
    print("Sum of diagonal elements:", diagonal_sum)
```

```
➦ Sum of diagonal elements: 4
```

Explanation:

Using NumPy, we define a given array, compute the sum of its diagonal elements with `np.trace`, and print the result.

Question d:

```
import numpy as np
array1 = np.array([[1, 2], [3, 4], [5, 6]])
reshaped_array1 = array1.reshape(3, 2)
print("Reshape 3x2:")
print(reshaped_array1)
print()
array2 = np.array([[1, 2, 3],
                   [4, 5, 6]])
reshaped_array2 = array2.reshape(2, 3)
print("Reshape 2x3:")
print(reshaped_array2)
```

```
⇒ Reshape 3x2:
[[1 2]
 [3 4]
 [5 6]]

Reshape 2x3:
[[1 2 3]
 [4 5 6]]
```

Explanation:

We use the reshape method to reshape the array array1 from 3x2 to 3x2, which does not change the data. We use the reshape method to reshape the array array2 from 2x3 to 2x3, which does not change the data.

Github Link:

<https://github.com/saideep8/MachineLearning/tree/main>