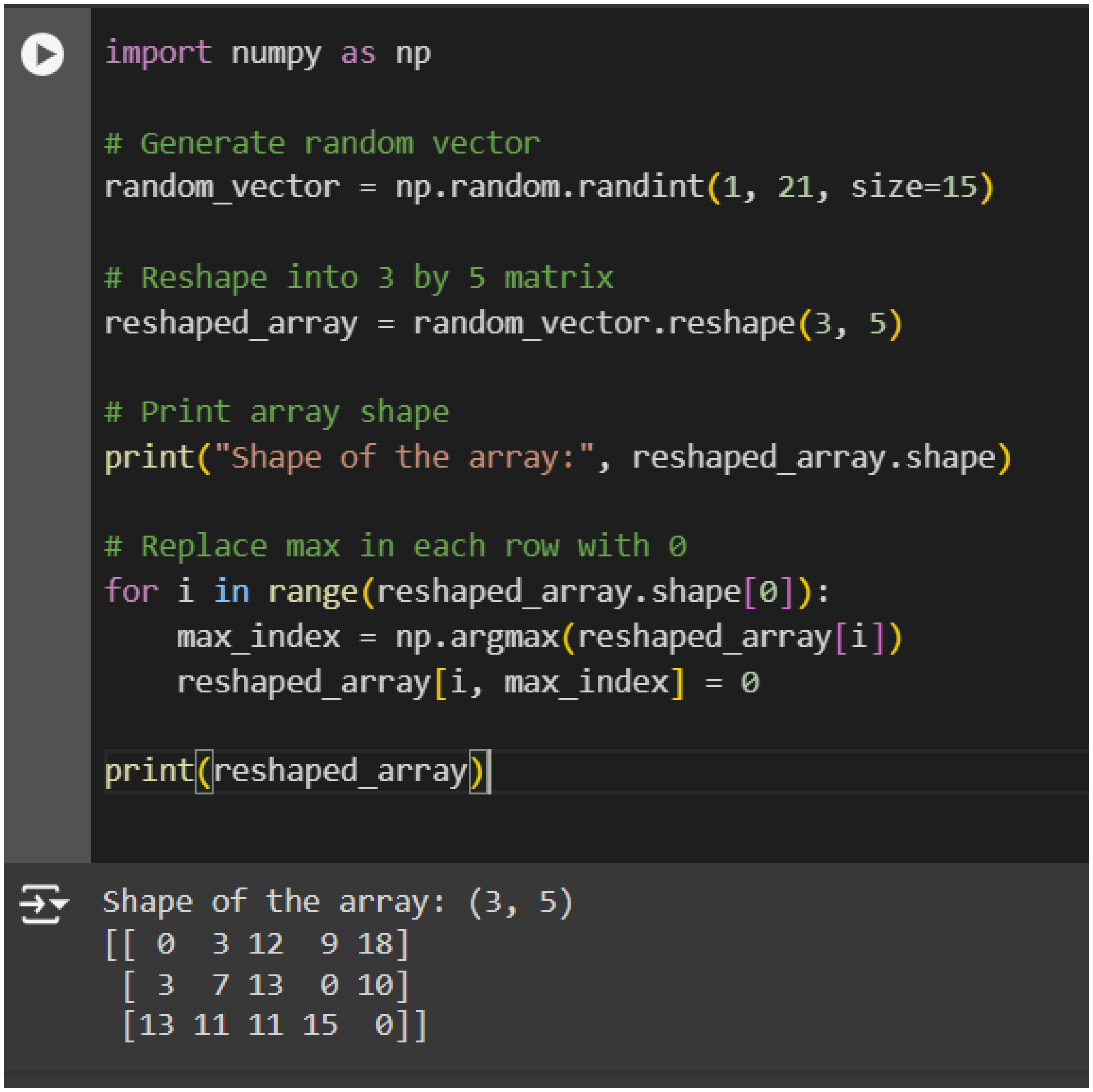
MACHINE LEARNING

Prem Kumar Kamma

700756204

Github Link: https://github.com/PremKumarKamma/ML_Assignment3

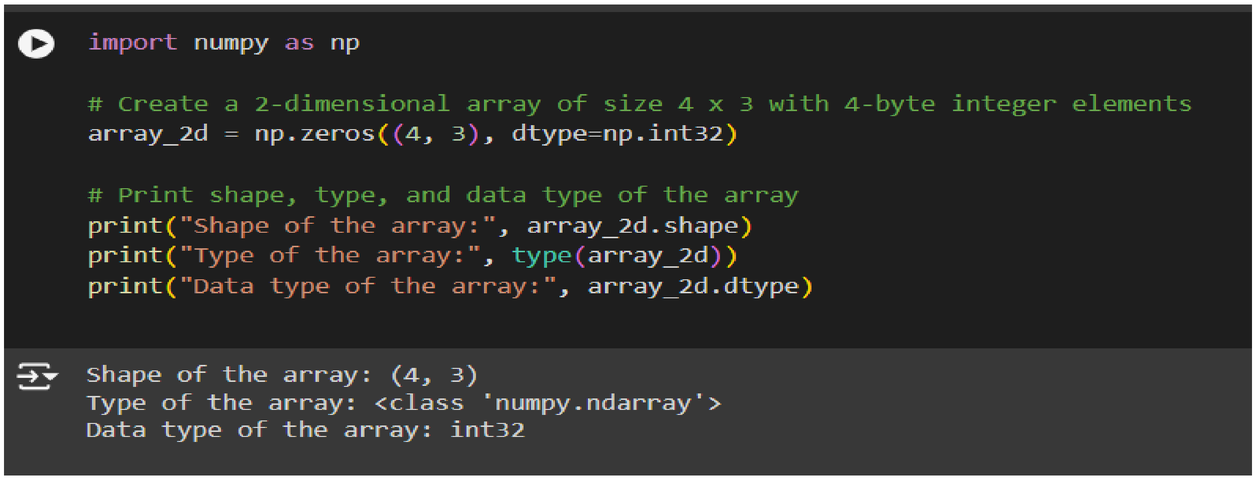
1)



```python
import numpy as np

# Generate random vector
random_vector = np.random.randint(1, 21, size=15)

# Reshape into 3 by 5 matrix
reshaped_array = random_vector.reshape(3, 5)

# Print array shape
print("Shape of the array:", reshaped_array.shape)

# Replace max in each row with 0
for i in range(reshaped_array.shape[0]):
    max_index = np.argmax(reshaped_array[i])
    reshaped_array[i, max_index] = 0

print(reshaped_array)
```

```
Shape of the array: (3, 5)
[[ 0  3 12  9 18]
 [ 3  7 13  0 10]
 [13 11 11 15  0]]
```

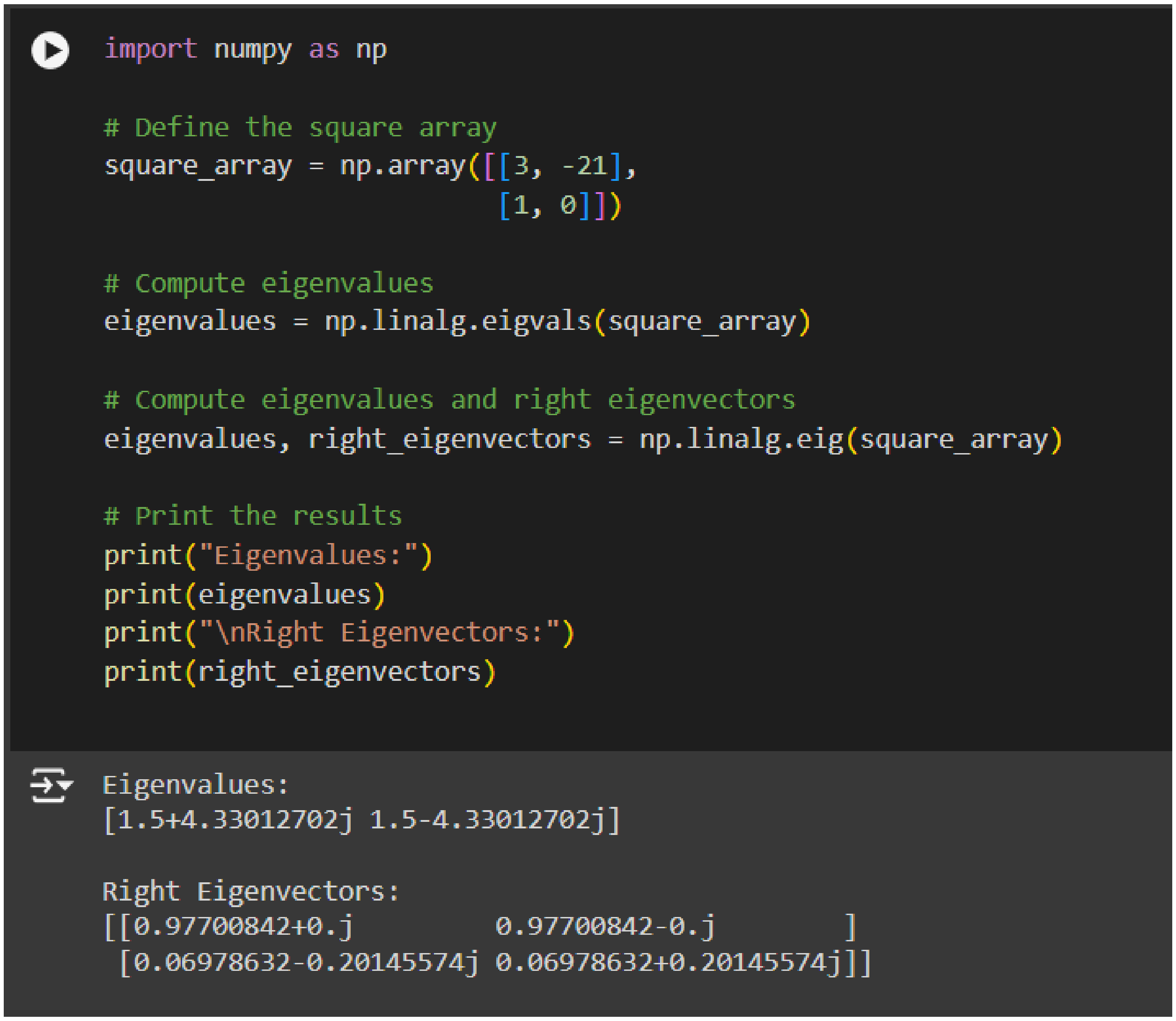
2)



```
import numpy as np

# Create a 2-dimensional array of size 4 x 3 with 4-byte integer elements
array_2d = np.zeros((4, 3), dtype=np.int32)

# Print shape, type, and data type of the array
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)
```

```
Shape of the array: (4, 3)
Type of the array: <class 'numpy.ndarray'>
Data type of the array: int32
```

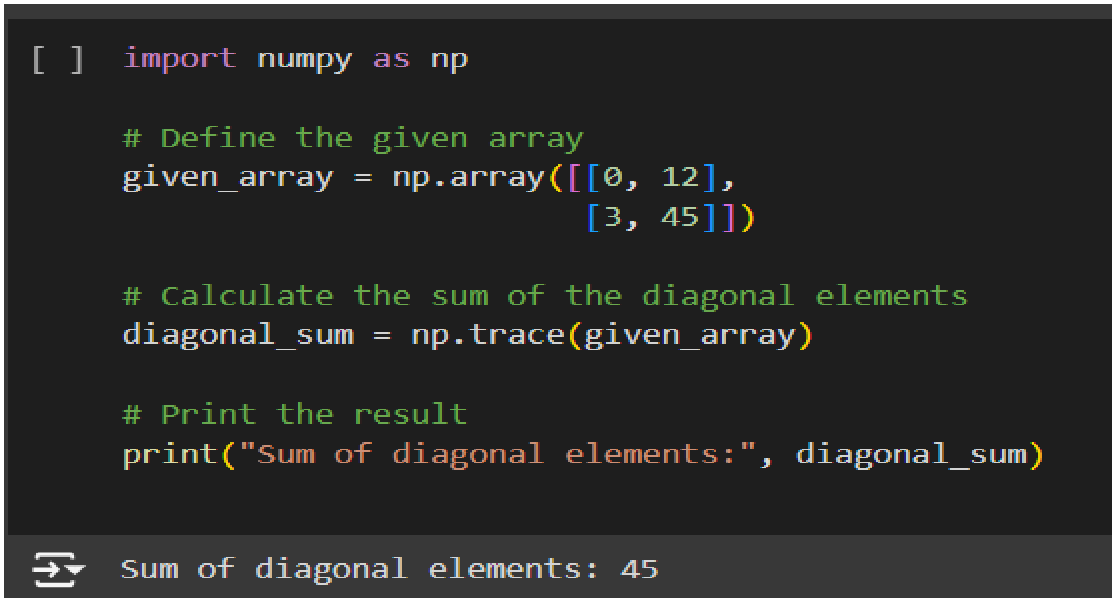
3)



```
import numpy as np

# Define the square array
square_array = np.array([[3, -21],
                         [1, 0]])

# Compute eigenvalues
eigenvalues = np.linalg.eigvals(square_array)

# Compute eigenvalues and right eigenvectors
eigenvalues, right_eigenvectors = np.linalg.eig(square_array)

# Print the results
print("Eigenvalues:")
print(eigenvalues)
print("\nRight Eigenvectors:")
print(right_eigenvectors)
```

```
Eigenvalues:
[1.5+4.33012702j 1.5-4.33012702j]

Right Eigenvectors:
[[0.97700842+0.j         0.97700842-0.j        ]
 [0.06978632-0.20145574j 0.06978632+0.20145574j]]
```

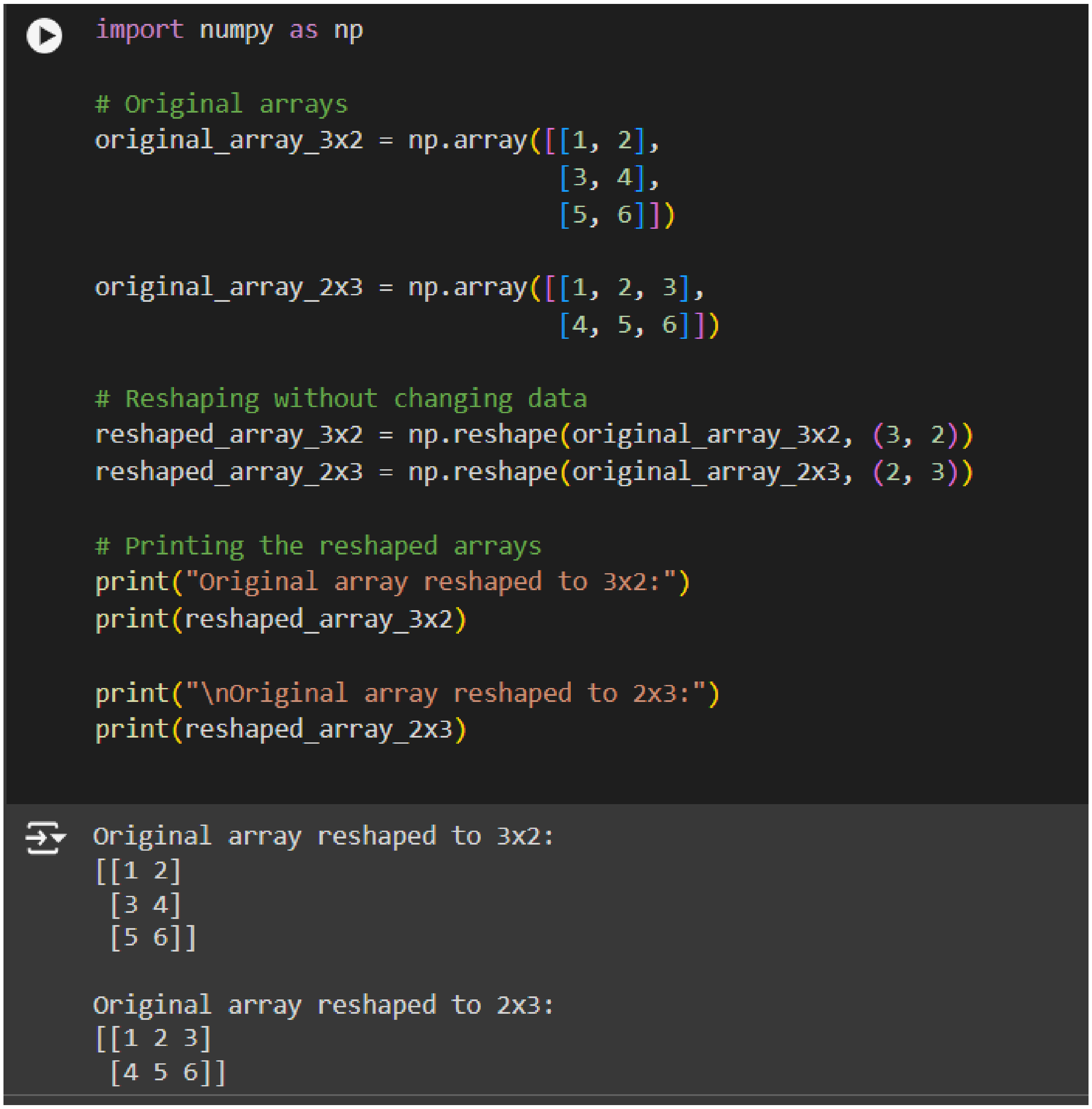
4)



```
[ ]  import numpy as np

     # Define the given array
     given_array = np.array([[0, 12],
                             [3, 45]])

     # Calculate the sum of the diagonal elements
     diagonal_sum = np.trace(given_array)

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

```
Sum of diagonal elements: 45
```

5)



```python
import numpy as np

# Original arrays
original_array_3x2 = np.array([[1, 2],
                               [3, 4],
                               [5, 6]])

original_array_2x3 = np.array([[1, 2, 3],
                               [4, 5, 6]])

# Reshaping without changing data
reshaped_array_3x2 = np.reshape(original_array_3x2, (3, 2))
reshaped_array_2x3 = np.reshape(original_array_2x3, (2, 3))

# Printing the reshaped arrays
print("Original array reshaped to 3x2:")
print(reshaped_array_3x2)

print("\nOriginal array reshaped to 2x3:")
print(reshaped_array_2x3)
```

```
Original array reshaped to 3x2:
[[1 2]
 [3 4]
 [5 6]]

Original array reshaped to 2x3:
[[1 2 3]
 [4 5 6]]
```