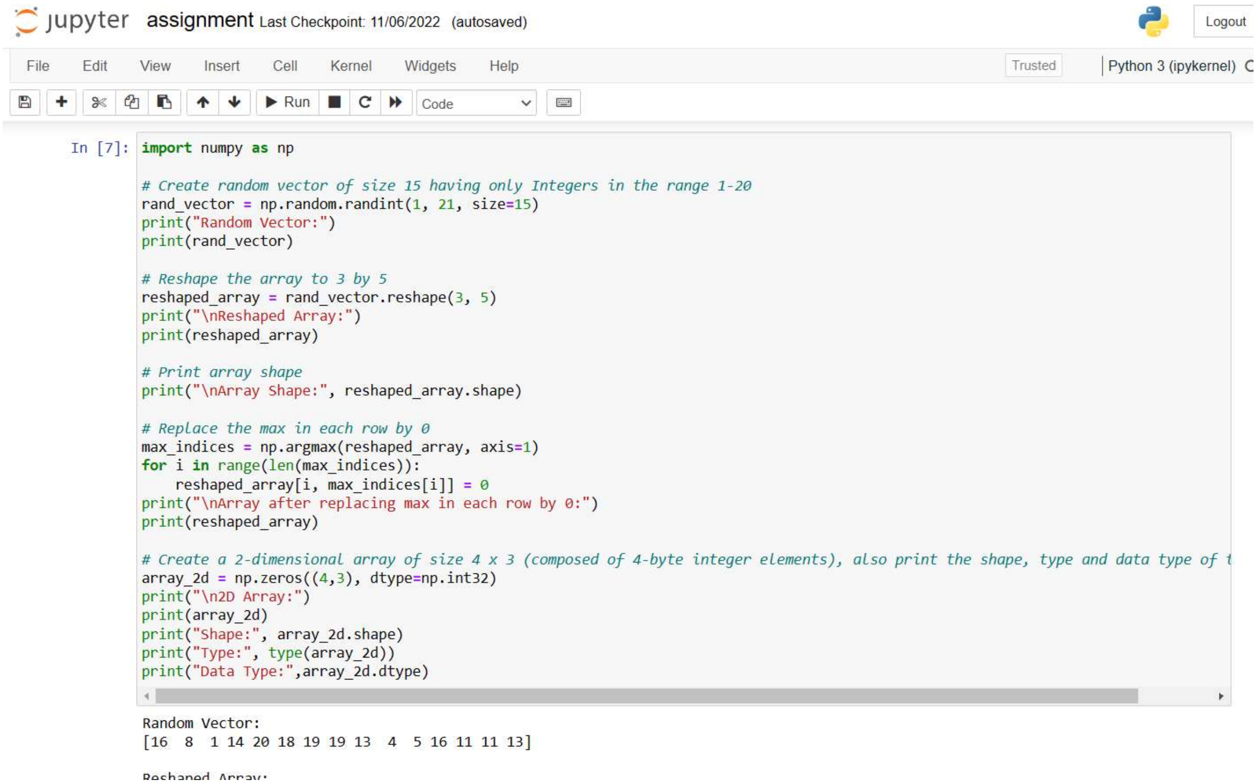
MACHINE LEARNING ASSIGNMENT - 3

Jagadeeswar Chimata
700731649

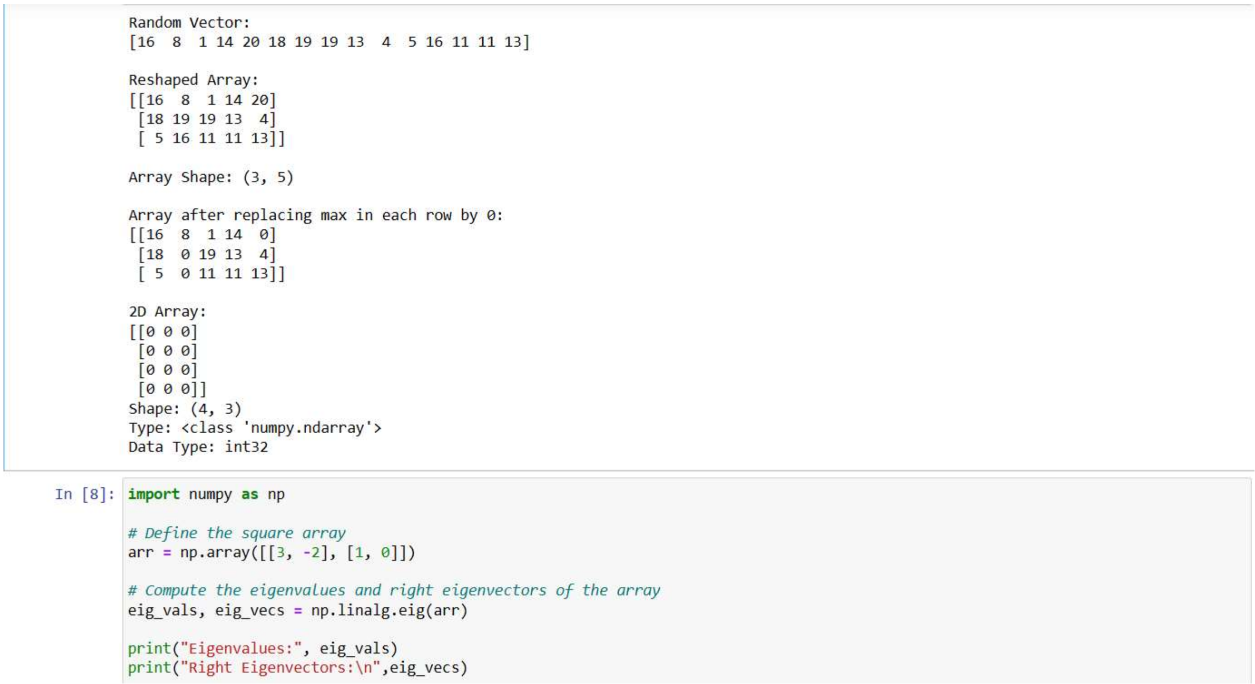
1 a. Using NumPy create random vector of size 15 having only Integers in the range 1-20. 1. Reshape the array to 3 by 5

2. Print array shape. 3. Replace the max in each row by 0 Create a 2-dimensional array of size 4 x 3 (composed of 4-byte integer elements), also print the shape, type and data type of the array.



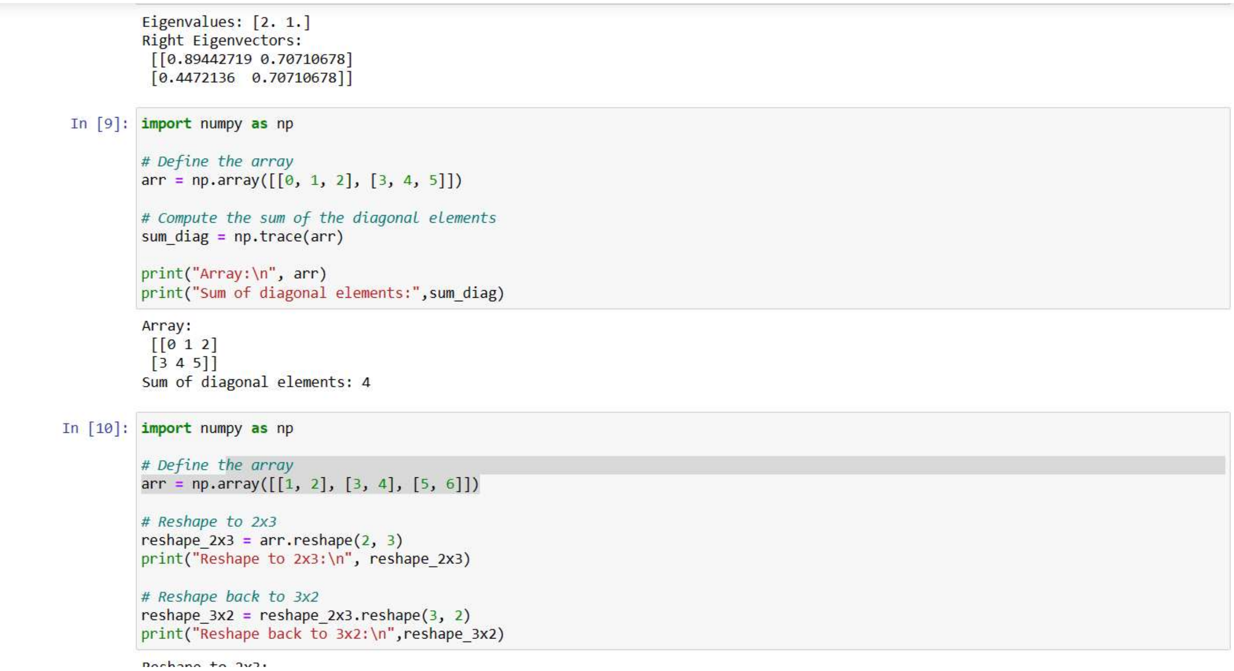
```
import numpy as np

# Create random vector of size 15 having only Integers in the range 1-20
rand_vector = np.random.randint(1, 21, size=15)
print("Random Vector:")
print(rand_vector)

# Reshape the array to 3 by 5
reshaped_array = rand_vector.reshape(3, 5)
print("\nReshaped Array:")
print(reshaped_array)

# Print array shape
print("\nArray Shape:", reshaped_array.shape)

# Replace the max in each row by 0
max_indices = np.argmax(reshaped_array, axis=1)
for i in range(len(max_indices)):
    reshaped_array[i, max_indices[i]] = 0
print("\nArray after replacing max in each row by 0:")
print(reshaped_array)

# Create a 2-dimensional array of size 4 x 3 (composed of 4-byte integer elements), also print the shape, type and data type of t
array_2d = np.zeros((4,3), dtype=np.int32)
print("\n2D Array:")
print(array_2d)
print("Shape:", array_2d.shape)
print("Type:", type(array_2d))
print("Data Type:",array_2d.dtype)
```

```
Random Vector:
[16  8  1 14 20 18 19 19 13  4  5 16 11 11 13]

Reshaped Array:
```



```
Random Vector:
[16  8  1 14 20 18 19 19 13  4  5 16 11 11 13]

Reshaped Array:
[[16  8  1 14 20]
 [18 19 19 13  4]
 [ 5 16 11 11 13]]

Array Shape: (3, 5)

Array after replacing max in each row by 0:
[[16  8  1 14  0]
 [18  0 19 13  4]
 [ 5  0 11 11 13]]

2D Array:
[[0 0 0]
 [0 0 0]
 [0 0 0]
 [0 0 0]]
Shape: (4, 3)
Type: <class 'numpy.ndarray'>
Data Type: int32
```

```
In [8]: import numpy as np

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

        # Compute the eigenvalues and right eigenvectors of the array
        eig_vals, eig_vecs = np.linalg.eig(arr)

        print("Eigenvalues:", eig_vals)
        print("Right Eigenvectors:\n",eig_vecs)
```



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

```
In [9]: import numpy as np

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

        # Compute the sum of the diagonal elements
        sum_diag = np.trace(arr)

        print("Array:\n", arr)
        print("Sum of diagonal elements:",sum_diag)
```

```
Array:
 [[0 1 2]
 [3 4 5]]
Sum of diagonal elements: 4
```

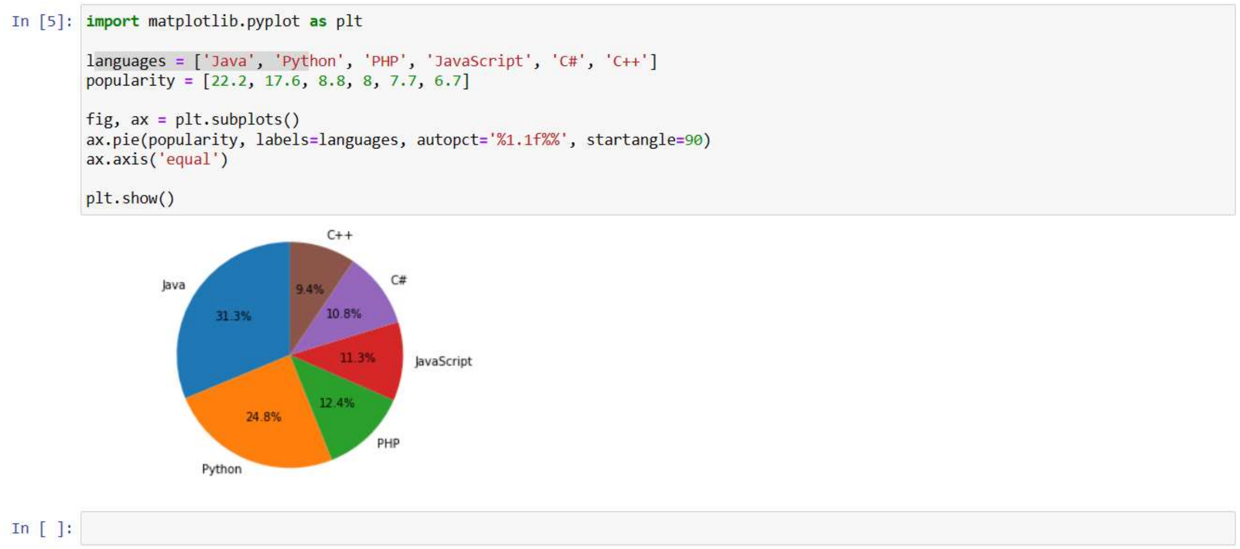
```
In [10]: import numpy as np

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

         # Reshape to 2x3
         reshape_2x3 = arr.reshape(2, 3)
         print("Reshape to 2x3:\n", reshape_2x3)

         # Reshape back to 3x2
         reshape_3x2 = reshape_2x3.reshape(3, 2)
         print("Reshape back to 3x2:\n",reshape_3x2)
```

```
Reshape to 2x3:
```

2)

1. Write a Python programming to create a below chart of the popularity of programming Languages.

2. Sample data: Programming languages: Java, Python, PHP, JavaScript, C#, C++ Popularity: 22.2, 17.6, 8.8, 8, 7.7, 6.7



```
In [5]: import matplotlib.pyplot as plt

        languages = ['Java', 'Python', 'PHP', 'JavaScript', 'C#', 'C++']
        popularity = [22.2, 17.6, 8.8, 8, 7.7, 6.7]

        fig, ax = plt.subplots()
        ax.pie(popularity, labels=languages, autopct='%1.1f%%', startangle=90)
        ax.axis('equal')

        plt.show()
```

```
In [ ]:
```

Git-Hub link : https://github.com/JagadeeswarChimata/Assignment-3.git