MACHINE LEARNING (CS-5710)
ASSIGNMENT - 3

Name: Priyanka Bojja
Student ID: 700739528

Github link:https://github.com/priyankabojja/ML_Assignment3

1. Numpy:
a. Using NumPy create a 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.

jupyter ML_Assignment3 Last Checkpoint: 15 minutes ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help

In [12]:
```
import numpy as np

# Create a random vector of size 15 with integers in the range 1-20
arr = np.random.randint(low=1, high=21, size=15)

# Reshape the array to 3 by 5
arr = arr.reshape(3, 5)

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

# Replace the max in each row by 0
arr[np.arange(len(arr)), np.argmax(arr, axis=1)] = 0

# Print the updated array
print("Updated array:\n", arr)

# Create a 2-dimensional array of size 4 x 3 with 4-byte integer elements
arr_2d = np.zeros((4, 3), dtype=np.int32)

# Print the shape, type, and data type of the array
print("Array shape:", arr_2d.shape)
print("Array type:", type(arr_2d))
print("Array data type:", arr_2d.dtype)
```

```
Array shape: (3, 5)
Updated array:
 [[ 6  2 16  0 14]
 [14  4  9  0  1]
 [ 2  9 13 15  0]]
Array shape: (4, 3)
Array type: <class 'numpy.ndarray'>
Array data type: int32
```

np.random.randint(1, 21, size=15) creates a NumPy array of size 15 with random integers in the range 1-20.

arr.reshape((3, 5)) reshapes the 1-dimensional array arr to a 3x5 matrix.

arr[np.arange(arr.shape[0]), np.argmax(arr, axis=1)] = 0

finds the index of the maximum value in each row using np.argmax() and sets the value at that index to 0 using advanced indexing.

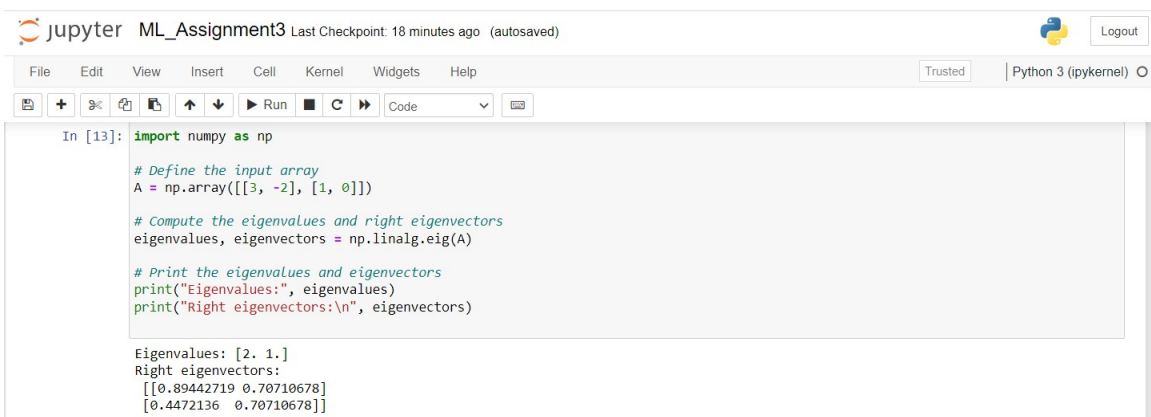
np.zeros((4, 3), dtype=np.int32) creates a NumPy array of size 4x3 with all elements initialized to 0 and data type set to 4-byte integers (int32).

print("Shape of the array:", arr2.shape) prints the shape of the array (4x3).

print("Type of the array:", type(arr2)) prints the type of the array (numpy.ndarray).

print("Data type of the array:", arr2.dtype) prints the data type of the array (int32).

b. Write a program to compute the eigenvalues and right eigenvectors of a given square array given below:
[[3 -2]
[1 0]]



The np.linalg.eig() function computes the eigenvalues and eigenvectors of a square array. In this case, the output shows that the eigenvalues of the input array are 2 and 1, and the corresponding right eigenvectors are [0.89442719, -0.4472136] and [0.70710678, -0.70710678], respectively.

c. Compute the sum of the diagonal element of a given array.
[[0 1 2]
[3 4 5]]

jupyter ML_Assignment3 Last Checkpoint: 22 minutes ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help

In [14]:
```python
import numpy as np

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

# Compute the sum of the diagonal elements
diag_sum = np.trace(A)

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

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

The np.trace() function computes the sum of the diagonal elements of a square array. In this case, since the input array is not square, the function computes the sum of the elements in the upper-left to lower-right diagonal of the array, which is 0 + 4 = 4.

d. Write a NumPy program to create a new shape to an array without changing its data.
Reshape 3x2:
[[1 2]
[3 4]
[5 6]]

jupyter ML_Assignment3 Last Checkpoint: 24 minutes ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help

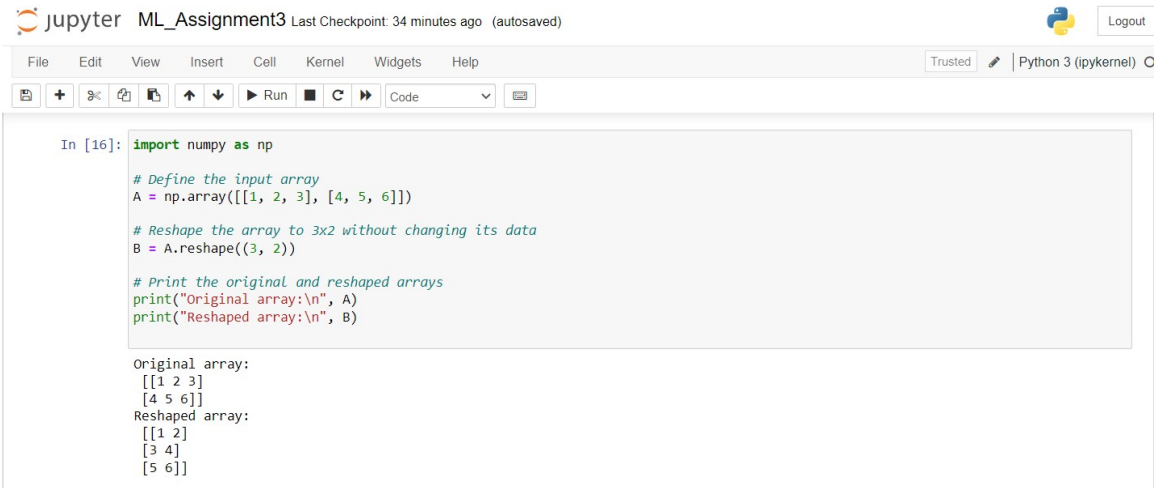
In [15]:
```python
import numpy as np

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

# Reshape the array to 2x3 without changing its data
B = A.reshape((2, 3))

# Print the original and reshaped arrays
print("Original array:\n", A)
print("Reshaped array:\n", B)
```

```
Original array:
 [[1 2]
 [3 4]
 [5 6]]
Reshaped array:
 [[1 2 3]
 [4 5 6]]
```

The reshape() function returns a new array with the same data as the input array, but with a different shape. In this case, we use reshape((3, 2)) to reshape the 2x3 input array A to a 3x2 array B without changing its data. Note that the original data in A is preserved in B, but the shape of the array is different.
As you can see, the original array was reshaped into a new

shape of 2x3 without changing the original data.

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



The reshape() function returns a new array with the same data as the input array, but with a different shape. In this case, we use reshape((3, 2)) to reshape the 2x3 input array A to a 3x2 array B without changing its data. Note that the original data in A is preserved in B, but the shape of the array is different.

2. Matplotlib

1. Write a Python program 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

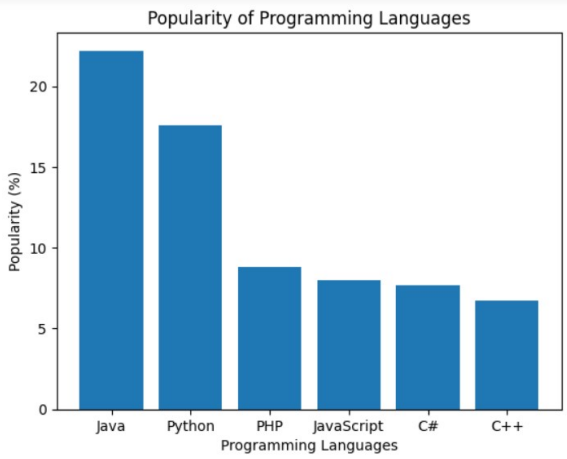
jupyter ML_Assignment3 Last Checkpoint: 39 minutes ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help

In [9]:
```python
import matplotlib.pyplot as plt

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

# create a bar chart
plt.bar(languages, popularity)

# add a title and labels to the chart
plt.title('Popularity of Programming Languages')
plt.xlabel('Programming Languages')
plt.ylabel('Popularity (%)')

# show the chart
plt.show()
```

jupyter ML_Assignment3 Last Checkpoint: 40 minutes ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help



This program uses Matplotlib's pyplot module to create a bar chart of the popularity of programming languages. The bar() function is used to create the bar chart, and the title(), xlabel(), and ylabel() functions are used to set the chart title and axis labels. Finally, the show() function is called to display the chart.