Summer 2023: CS5710 – Machine Learning

**In-Class Programming Assignment-1**

**Youtubelink:** **https://youtu.be/OpWFxmk3LYE**

**GITHUB LNK:**

**https://github.com/john672000/ML\_Assignmet700742724\_-30460.git**

**700742724(JOHN EMMANUEL)**

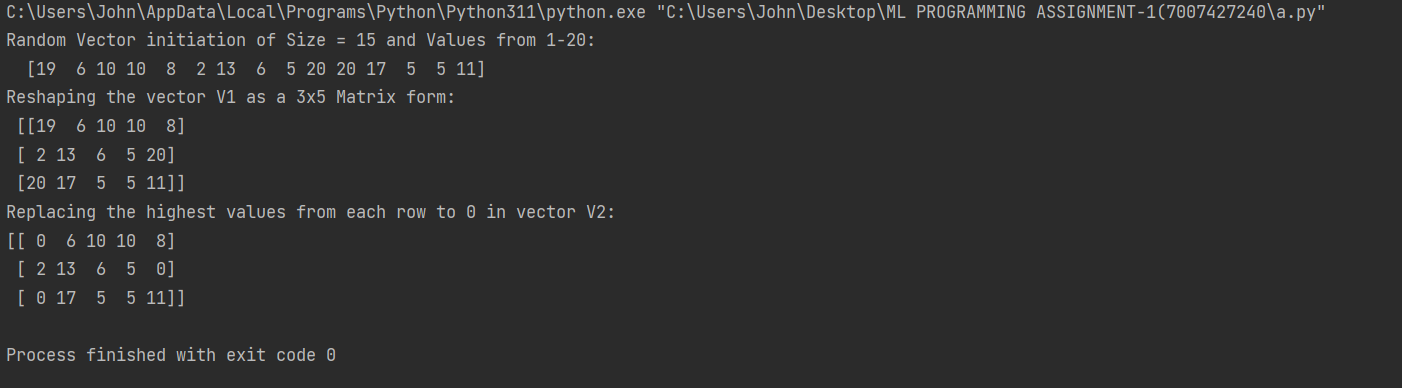
# Numpy:

* 1. 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

**Code:**

import numpy as np  
v1 = np.random.randint(1, 21, size=15)  
print("Random Vector initiation of Size = 15 and Values from 1-20:\n ", v1)  
  
v2 = v1.reshape(3, 5)  
print("Reshaping the vector V1 as a 3x5 Matrix form: \n", v2)  
  
  
print("Replacing the highest values from each row to 0 in vector V2: ")  
v2[np.arange(3), np.argmax(v2, axis=1)] = 0  
print(v2)

**Output:**

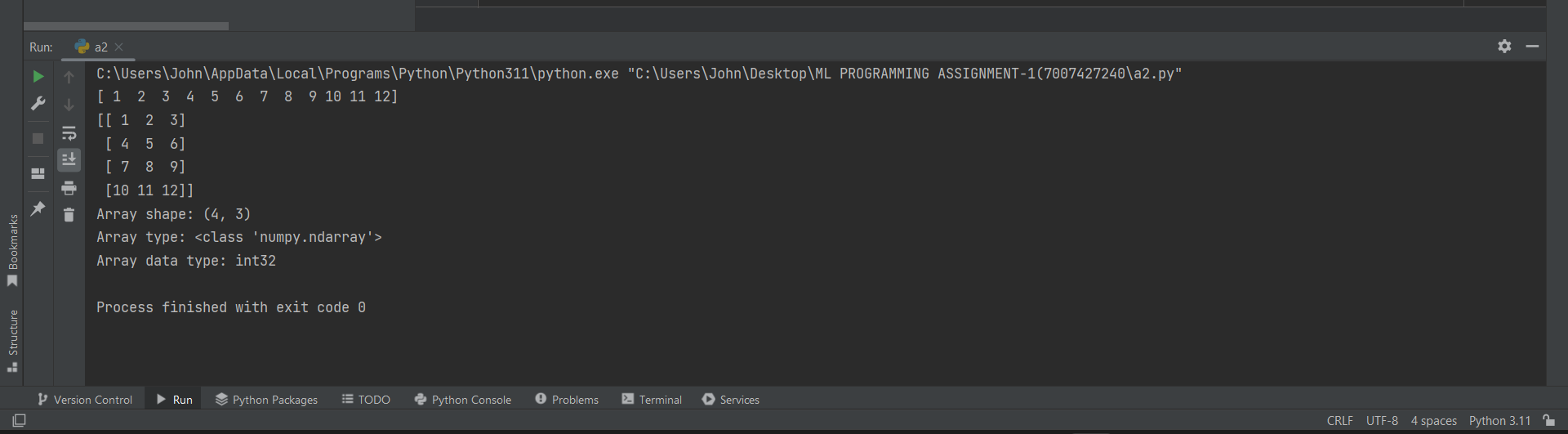
****

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.

**Code:**

*"""  
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.  
b. Write a program to compute the eigenvalues and right eigenvectors of a given square array given below:  
[[ 3 -2]  
[ 1 0]]  
  
"""*import numpy as np  
l1 = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]  
v1 = np.array(l1, dtype=np.int32) # (4 bytes = 4x8 = 32bit)  
print(v1)  
v2 = v1.reshape(4, 3)  
print(v2)  
  
# Print the shape of the array  
print("Array shape:", v2.shape)  
  
# Print the type of the array  
print("Array type:", type(v2))  
  
# Print the data type of the array  
print("Array data type:", v2.dtype)

**Output:**



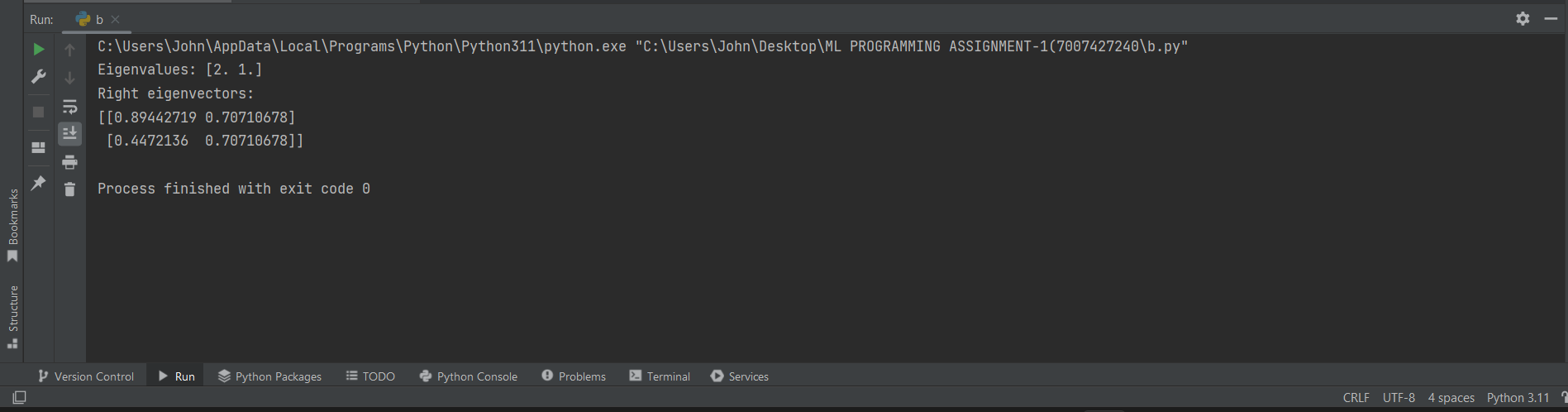
* 1. Write a program to compute the eigenvalues and right eigenvectors of a given square array given below: [[ 3 -2]

[ 1 0]]

**Code:**

import numpy as np  
  
# Given square array  
square\_array = np.array([[3, -2],  
 [1, 0]])  
  
# Compute eigenvalues and right eigenvectors  
eigenvalues, eigenvectors = np.linalg.eig(square\_array)  
  
"""  
Print the eigenvalues   
We can manually calculate eigen values as the following "(3-x)(0-x) - (-2x1)" where x gives the values  
"""  
print("Eigenvalues:", eigenvalues)  
  
# Print the right eigenvectors  
print("Right eigenvectors:")  
print(eigenvectors)

**Output:**

****

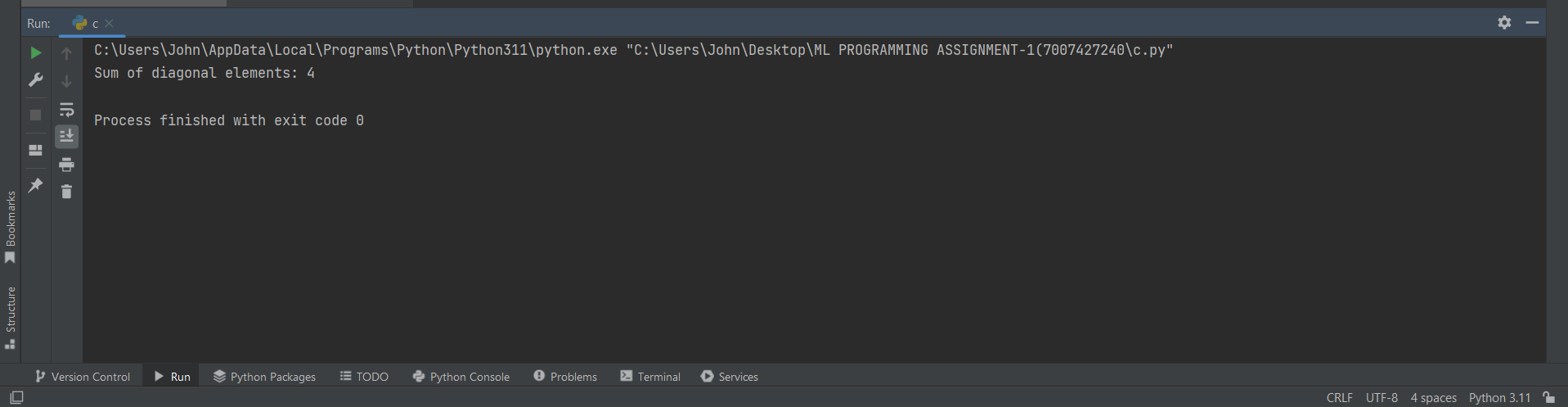
* 1. Compute the sum of the diagonal element of a given array. [[0 1 2]

[3 4 5]]

**Code:**

import numpy as np  
# Given array  
given\_array = np.array([[0, 1, 2],  
 [3, 4, 5]])  
  
# Compute the sum of diagonal elements  
diagonal\_sum = np.trace(given\_array)  
"""  
This above is a rectangular matrix and the diagonal as MP00 and MP11 as it has only 2 rows   
"""  
# Print the sum of diagonal elements  
print("Sum of diagonal elements:", diagonal\_sum)

**Output:**

****

* 1. Write a NumPy program to create a new shape to an array without changing its data. Reshape 3x2:

[[1 2]

[3 4]

[5 6]]

Reshape 2x3:

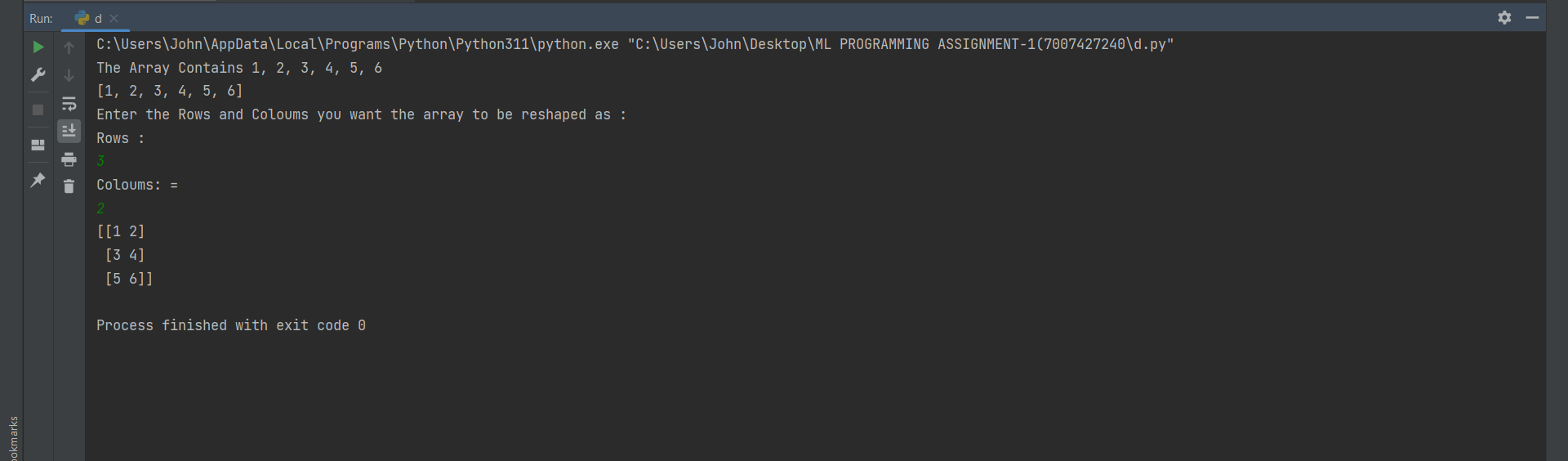
[[1 2 3]

[4 5 6]]

**Code:**

import numpy as np  
def reshape(a,r,c):  
 l = np.reshape(a, (r, c))  
 return l  
  
array = []  
print("The Array Contains 1, 2, 3, 4, 5, 6 ")  
for x in range (1,7):  
 array.append(x)  
  
print(array)  
print("Enter the Rows and Coloums you want the array to be reshaped as : ")  
print("Rows : ")  
r = int(input())  
print("Coloums: = ")  
c = int(input())  
new\_array = reshape(array,r,c)  
print(new\_array)

**Output:**

****

# Matplotlib

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

**Code:**

import matplotlib.pyplot as plt  
  
# Programming languages  
languages = ['Java', 'Python', 'PHP', 'JavaScript', 'C#', 'C++']  
  
# Popularity percentages  
popularity = [22.2, 17.6, 8.8, 8, 7.7, 6.7]  
  
# Create pie chart  
plt.pie(popularity, labels=languages, autopct='%1.1f%%')  
  
# Set aspect ratio to be equal so that pie is drawn as a circle  
plt.axis('equal')  
  
# Add a title  
plt.title('Popularity of Programming Languages')  
  
# Display the chart  
plt.show()

**Output:**

