

ML_Assignment_3

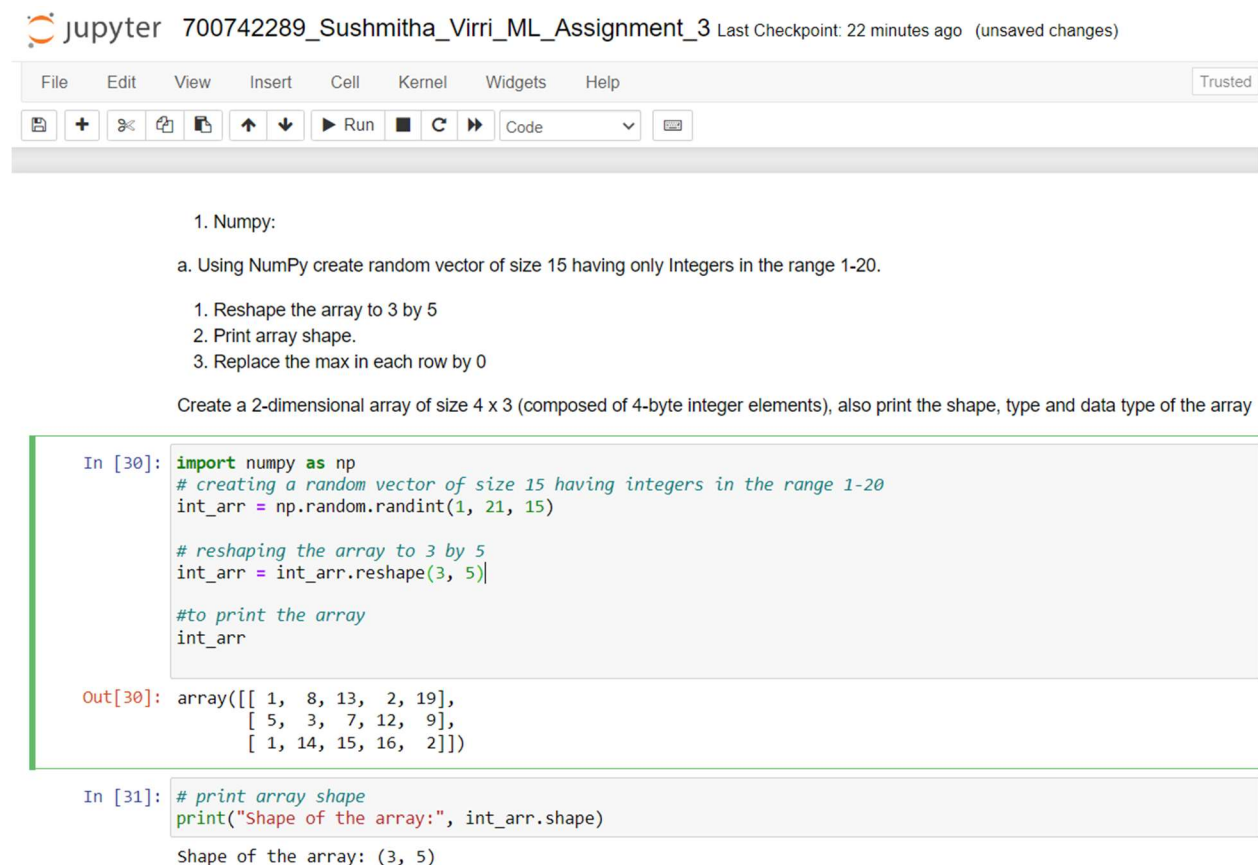
#700742289

Sushmitha Virri

GitHub Link: <https://github.com/Sushmitha-Virri/MLAssignments21627>

Video Link: https://drive.google.com/file/d/1Z3GyD6d-eRTt86gKA7Uqe48vgqOH59vf/view?usp=share_link

1.Numpy



Jupyter 700742289_Sushmitha_Virri_ML_Assignment_3 Last Checkpoint: 22 minutes ago (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help Trusted

1. Numpy:

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

```
In [30]: import numpy as np
# creating a random vector of size 15 having integers in the range 1-20
int_arr = np.random.randint(1, 21, 15)

# reshaping the array to 3 by 5
int_arr = int_arr.reshape(3, 5)

#to print the array
int_arr

Out[30]: array([[ 1,  8, 13,  2, 19],
 [ 5,  3,  7, 12,  9],
 [ 1, 14, 15, 16,  2]])

In [31]: # print array shape
print("Shape of the array:", int_arr.shape)

Shape of the array: (3, 5)
```

In this code I have imported NumPy library from python. With the help of `random.randint()` function I have created a One-Dimensional array with random integers in the range of 1-20

`reshape()` is the function to reshape the one-dimensional array into a two-dimensional array of size 3 by 5

```
File Edit View Insert Cell Kernel Widgets Help
[+] [-] [x] [y] [z] [a] [b] [c] [d] [e] [f] [g] [h] [i] [j] [k] [l] [m] [n] [o] [p] [q] [r] [s] [t] [u] [v] [w] [x] [y] [z] [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [Enter] [Shift] [Ctrl] [Alt] [Cmd] [F1] [F2] [F3] [F4] [F5] [F6] [F7] [F8] [F9] [F10] [F11] [F12] [Print] [Help]
Code [v] [x]
```

```
In [32]: #Replacing the max in each row by 0
int_arr[np.where(int_arr==np.max(int_arr))] = 0

#to print the array after replacing
int_arr

Out[32]: array([[ 1,  8, 13,  2,  0],
                [ 5,  3,  7, 12,  9],
                [ 1, 14, 15, 16,  2]])

In [34]: # to create a 2-dimensional array of size 4 x 3 (composed of 4-byte integer elements)
array1 = np.array([[10, 20, 22], [20, 42, 61], [1, 0, 9], [12, 8, 53]], dtype = np.int32)

# to print the shape,type and data type of the array
print("Shape of the array:", array1.shape)
print("Type of the array:", type(array1))
print("Data type of the array:", array1.dtype)

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

With the help of NumPy's where() function I have found the positions of the maximum values in the array using max() function in python, replaced them with '0'. Printed the modified array.

I have created the 2-dimensional array using the array() function and have set the datatype of the array to be 32-bit integer.

Then print() function

to print shape of the array using shape property,

to print the type of the array using type() function,

to print the data type of the array using dtype attribute of the array.

b. Write a program to compute the eigenvalues and right eigenvectors of a given square array given below:

[[3 -2]]

 $[1\ 0]$

```
In [36]: import numpy as np
#to define the square array using array() function of the numpy library
array2 = np.array([[3, -2], [1, 0]])
#to calculate the eigenvalues and right eigenvectors of the given square array using linalg.eig() function
eigen_values, eigen_vectors = np.linalg.eig(array2)

#to print the computed values
print("eigenvalues of the square array : ", eigen_values)
print("eigenvectors of the square array : ", eigen_vectors)

eigenvalues of the square array : [2. 1.]
eigenvectors of the square array : [[0.89442719 0.70710678]
 [0.4472136  0.70710678]]
```

For this question I have created an array called 'array2' with the given values.

`linalg.eig()` function of NumPy is used to calculate the eigenvalues and eigenvectors of the array. The `print()` function is used to print the labels along with the `eigen_values` and `eigen_vectors`.

c. Compute the sum of the diagonal element of a given array.

[[0 1 2]]

[3 4 5]

```
In [41]: import numpy as np
array3 = np.array([[0, 1, 2], [3, 4, 5]])

#to compute the sum of diagonal elements using trace() function
diag_elements_sum = np.trace(array3)

# to print the sum of the diagonal elements
print("sum of the diagonal elements of given array : ", diag_elements_sum)

sum of the diagonal elements of given array : 4
```

In this code I have imported NumPy library from python. Then created an array called 'array3' using the NumPy's array() function.

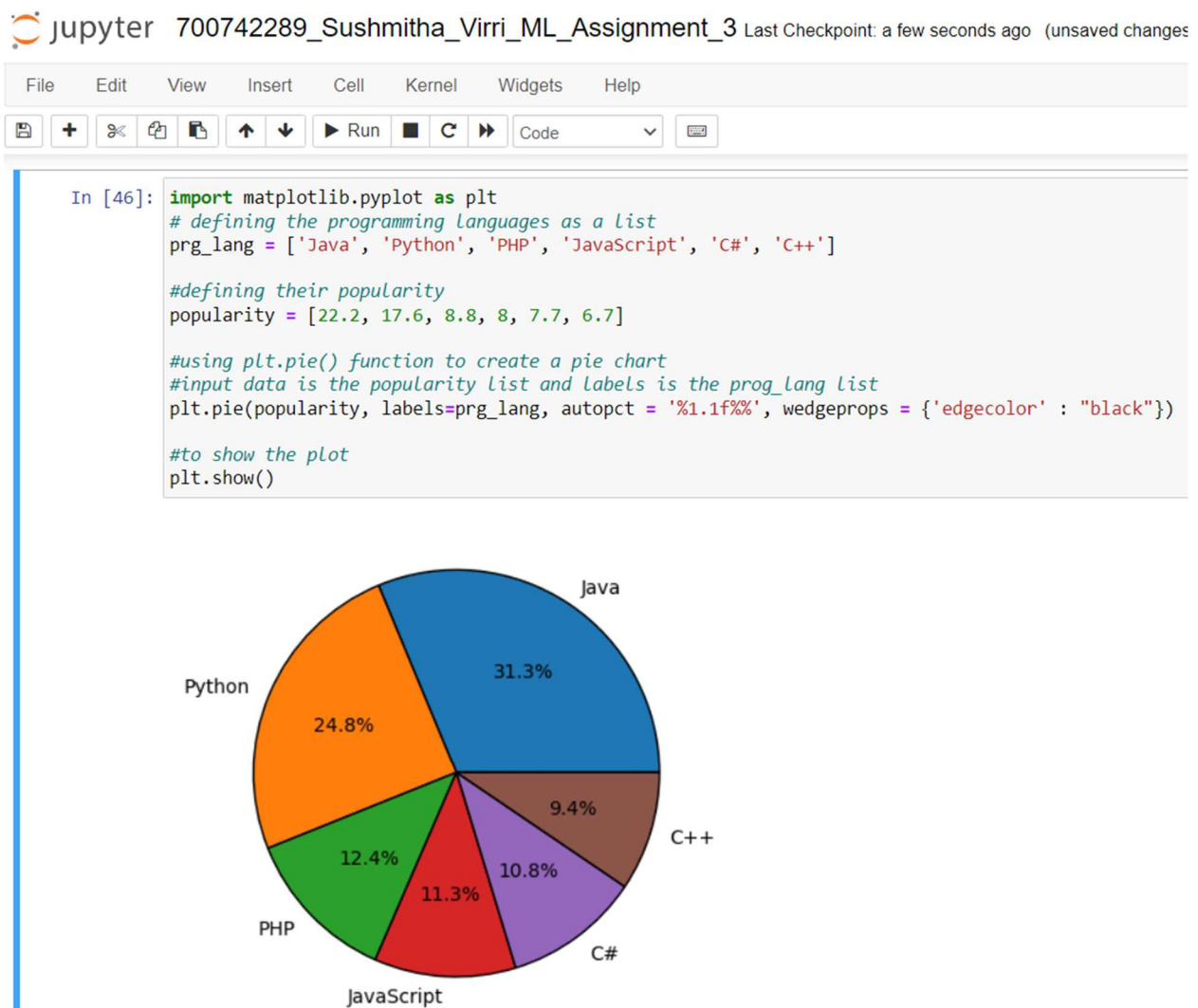
The trace() function is used to calculate the sum of the diagonal elements and then print the sum of the diagonal elements using the print function.

2. 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



For this Matplotlib question I have imported Matplotlib library from python and pyplot module from Matplotlib to create pie chart.

Next, I have defined the list of programming languages and their popularity percentages.

Then with the help of `pie()` function of the Matplotlib created a pie chart with

Input data: 'popularity' list

Labels for each slice of pie chart is the programming language list.

`autopct` to calculate percentages and display on each slice with the specified format.

`wedgeprops`: To specify the appearance of wedges in the pie chart with edge color set to black.