Spring 2023: CS5710 – Machine Learning

In-Class Programming Assignment-3

GitHub Link - https://github.com/raimukul/MachineLearning_Assignments

Video link- https://drive.google.com/file/d/12AmiOel5rgY9Ox13Js-YWO1owgW-GTqZ/view?usp=sharing

1. Numpy:

a. Using NumPy, create a random vector of size 15 with 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.

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

[[3 -2]

[1 0]]

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

[[0 1 2]

[3 4 5]]

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]]
```

Reshape 2x3:

```
        [[1 2 3]

        [4 5 6]]
```

```python
import numpy as npy;

# 1.a
vector = npy.random.randint(1, 20, 15)
print ("1.a Vector: ", vector)
# 1.a.1 Reshape the array to 3 by 5
reshaped = vector.reshape(3, 5)
```

```
1.a Vector:  [ 1 11 15  3 15  7  1  5  7  9  6  4  6 12 19]
```

```python
# 1.a.2 Print array shape.
print ("1.a.2 Reshaped array shape: ", reshaped.shape)
```

```
1.a.2 Reshaped array shape:  (3, 5)
```

```python
# 1.a.3 Replace the max in each row by 0.
for i in range(reshaped.shape[0]):
    reshaped[i, npy.where(reshaped[i] == reshaped[i].max())] = 0
print ("1.a.3 Replaced max in each row by 0: \n", reshaped)
```

```
1.a.3 Replaced max in each row by 0:
 [[ 1 11  0  3  0]
 [ 7  1  5  7  0]
 [ 6  4  6 12  0]]
```

```
# 1.b compute the eigenvalues and right eigenvectors of a given square array
array = npy.random.randint(1, 20, (4, 3), dtype=npy.int32)
print ("1.b Array: \n", array)
print ("1.b Array shape: ", array.shape)
print ("1.b Array type: ", type(array))
print ("1.b Array data type: ", array.dtype)
```

```
1.b Array:
 [[13  6  3]
 [ 7 19  9]
 [17  3  5]
 [ 8 11  1]]
1.b Array shape:  (4, 3)
1.b Array type:  <class 'numpy.ndarray'>
1.b Array data type:  int32
```

```
# 1.b
newArray = npy.array([[3, -2], [1, 0]])
eigenvalues, eigenvectors = npy.linalg.eig(newArray)
print ("1.b Eigenvalues: \n", eigenvalues)
print ("1.b Eigenvectors: \n", eigenvectors)
```

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

```
# 1.c sum of the diagonal element of a given array:
oneC = npy.array([[0, 1, 2], [3, 4, 5]])
print ("1.c Array: \n", oneC)
print ("1.c Sum of diagonal elements: ", npy.trace(oneC))
```

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

```
# 1.d new shape to an array without changing its data. Reshape 3x2:
oneD = npy.arange(1, 7)
print ("1.d Array: ", oneD)
```

```
# reshape to 3x2
oneD = oneD.reshape(3, 2)
print ("1.d Reshaped array 3x2: \n", oneD)
# reshape to 2x3
oneD = oneD.reshape(2, 3)
print ("1.d Reshaped array 2x3: \n", oneD)
```

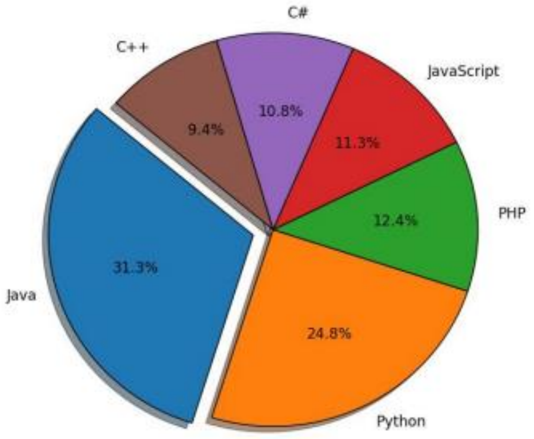
```
1.d Array:  [1 2 3 4 5 6]
1.d Reshaped array 3x2:
 [[1 2]
 [3 4]
 [5 6]]
1.d Reshaped array 2x3:
 [[1 2 3]
 [4 5 6]]
```

2. Matplotlib

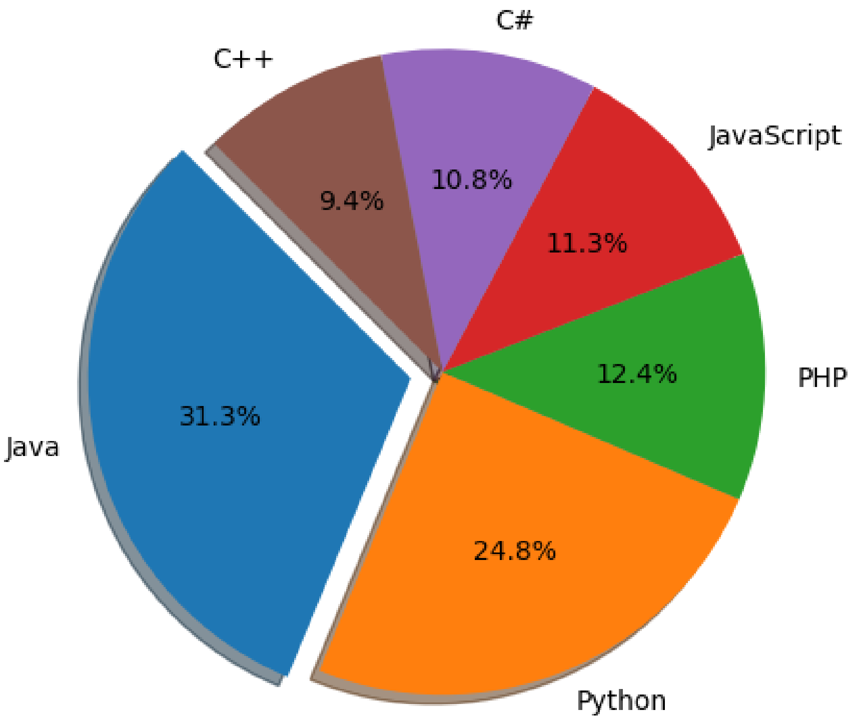
1. Write a Python programming to create a below chart of the popularity of programming Languages.
2. Sample data:

Programming languages: Java, Python, PHP, JavaScript, C#, C++

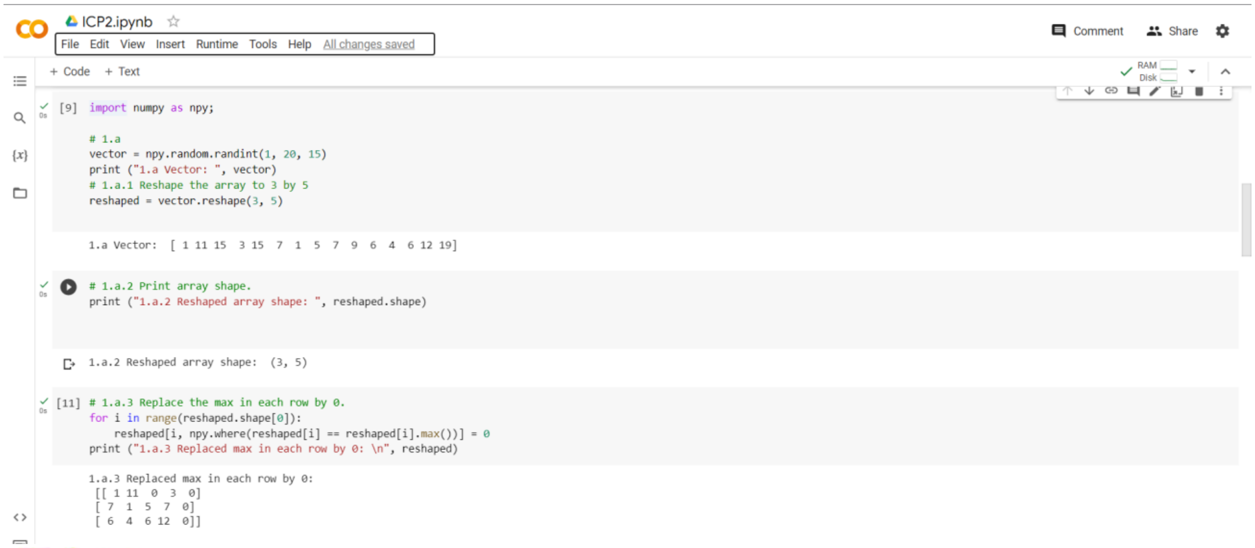
Popularity: 22.2, 17.6, 8.8, 8, 7.7, 6.7



```
#2
import matplotlib.pyplot as plt

programmingLanguages = ['Java', 'Python', 'PHP', 'JavaScript', 'C#', 'C++']
popularity = [22.2, 17.6, 8.8, 8, 7.7, 6.7]
dictionary = dict(zip(programmingLanguages, popularity))
dictionary = dict(sorted(dictionary.items(), key=lambda item: item[1], reverse=True))
explode = (0.1, 0, 0, 0, 0, 0)
plt.pie(dictionary.values(), labels=dictionary.keys(), explode=explode, autopct='%1.1f%%',
shadow=True, startangle=135)
plt.axis('equal')
plt.show()
```



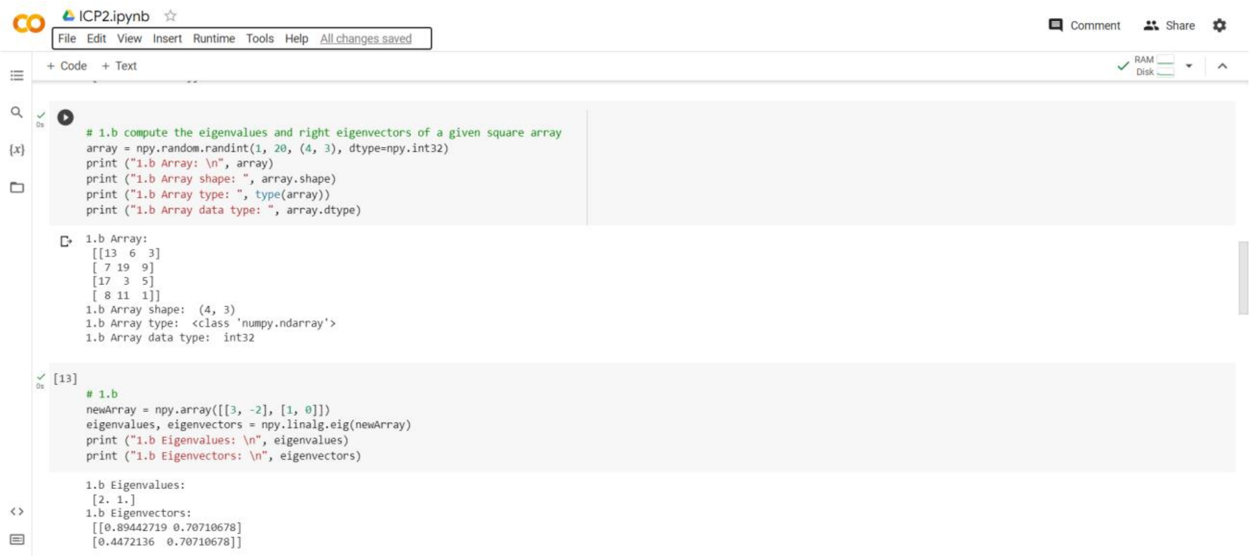
Screenshots:



```
[9] import numpy as npy;

    # 1.a
    vector = npy.random.randint(1, 20, 15)
    print ("1.a Vector: ", vector)
    # 1.a.1 Reshape the array to 3 by 5
    reshaped = vector.reshape(3, 5)

    1.a Vector:  [ 1 11 15  3 15  7  1  5  7  9  6  4  6 12 19]
```

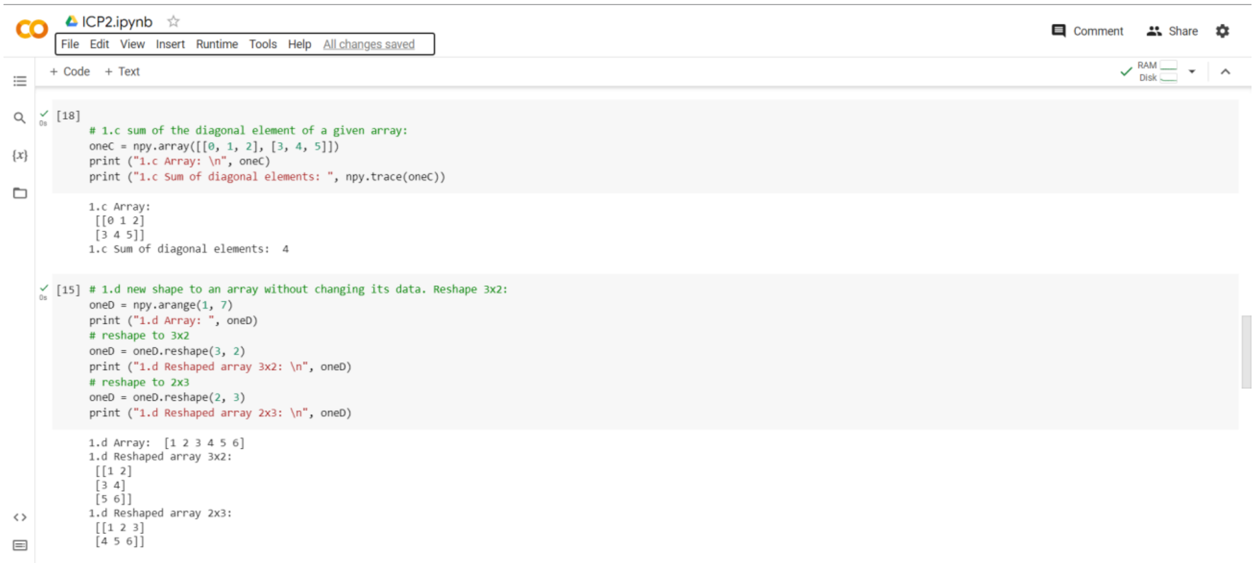
```
    # 1.a.2 Print array shape.
    print ("1.a.2 Reshaped array shape: ", reshaped.shape)

    1.a.2 Reshaped array shape:  (3, 5)
```

```
[11] # 1.a.3 Replace the max in each row by 0.
     for i in range(reshaped.shape[0]):
         reshaped[i, npy.where(reshaped[i] == reshaped[i].max())] = 0
     print ("1.a.3 Replaced max in each row by 0: \n", reshaped)

     1.a.3 Replaced max in each row by 0:
      [[ 1 11  0  3  0]
      [ 7  1  5  7  0]
      [ 6  4  6 12  0]]
```



```
    # 1.b compute the eigenvalues and right eigenvectors of a given square array
    array = npy.random.randint(1, 20, (4, 3), dtype=npy.int32)
    print ("1.b Array: \n", array)
    print ("1.b Array shape: ", array.shape)
    print ("1.b Array type: ", type(array))
    print ("1.b Array data type: ", array.dtype)

    1.b Array:
     [[13  6  3]
     [ 7 19  9]
     [17  3  5]
     [ 8 11  1]]
    1.b Array shape:  (4, 3)
    1.b Array type:  <class 'numpy.ndarray'>
    1.b Array data type:  int32
```

```
[13]
     # 1.b
     newArray = npy.array([[3, -2], [1, 0]])
     eigenvalues, eigenvectors = npy.linalg.eig(newArray)
     print ("1.b Eigenvalues: \n", eigenvalues)
     print ("1.b Eigenvectors: \n", eigenvectors)

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

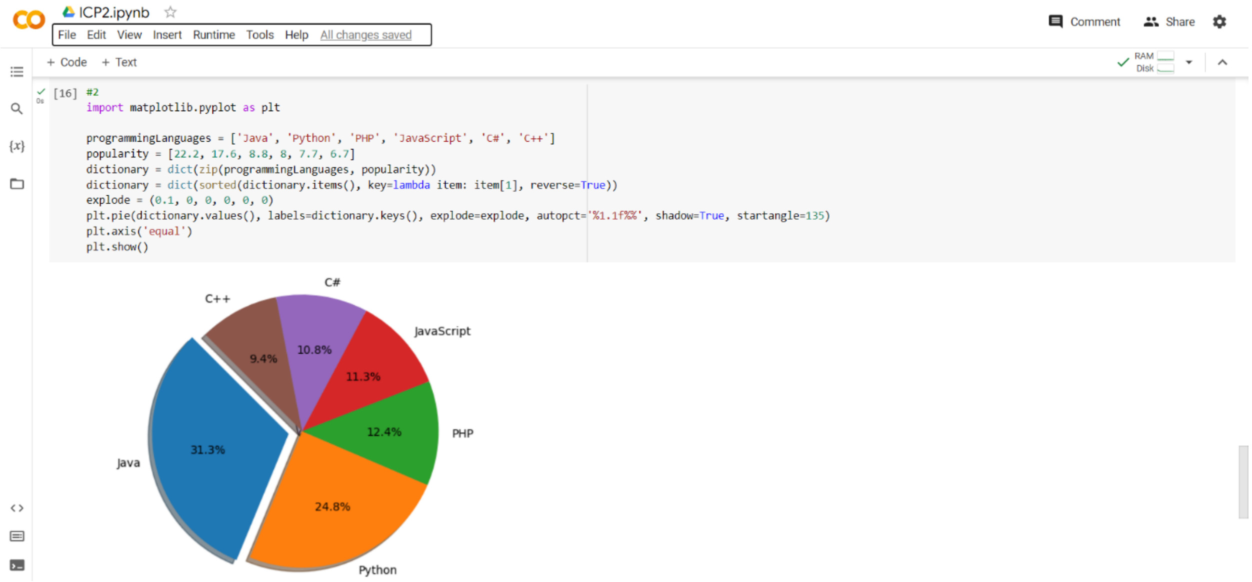


```
# 1.c sum of the diagonal element of a given array:
oneC = npy.array([[0, 1, 2], [3, 4, 5]])
print ("1.c Array: \n", oneC)
print ("1.c Sum of diagonal elements: ", npy.trace(oneC))
```

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

```
# 1.d new shape to an array without changing its data. Reshape 3x2:
oneD = npy.arange(1, 7)
print ("1.d Array: ", oneD)
# reshape to 3x2
oneD = oneD.reshape(3, 2)
print ("1.d Reshaped array 3x2: \n", oneD)
# reshape to 2x3
oneD = oneD.reshape(2, 3)
print ("1.d Reshaped array 2x3: \n", oneD)
```

```
1.d Array:  [1 2 3 4 5 6]
1.d Reshaped array 3x2:
 [[1 2]
 [3 4]
 [5 6]]
1.d Reshaped array 2x3:
 [[1 2 3]
 [4 5 6]]
```



```
#2
import matplotlib.pyplot as plt

programmingLanguages = ['Java', 'Python', 'PHP', 'JavaScript', 'C#', 'C++']
popularity = [22.2, 17.6, 8.8, 8, 7.7, 6.7]
dictionary = dict(zip(programmingLanguages, popularity))
dictionary = dict(sorted(dictionary.items(), key=lambda item: item[1], reverse=True))
explode = (0.1, 0, 0, 0, 0, 0)
plt.pie(dictionary.values(), labels=dictionary.keys(), explode=explode, autopct='%1.1f%%', shadow=True, startangle=135)
plt.axis('equal')
plt.show()
```