MACHINE LEARNING (CS-5710)

ASSIGNMENT - 3

Name: Neeraj Kumar Kajuluri

Student ID: 700742091

Git hub Link: -

Video Link: -

1. NumPy:

```
In [35]: ▶ import numpy as np
           y=np.random.randint(1,20,15)
           len(y)

  Out[35]: 15
```

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 [38]: ▶ reshaped=y.reshape((3, 5))
           reshaped

  Out[38]: array([[ 9, 10, 10,  3,  2],
                  [19, 18, 12,  7, 15],
                  [ 1,  1, 14,  1, 11]])

In [39]: ▶ reshaped.shape

  Out[39]: (3, 5)

In [55]: ▶ reshaped

  Out[55]: array([[ 9,  0,  0,  3,  2],
                  [ 0,  0, 12,  7,  0],
                  [ 1,  1,  0,  1,  0]])

In [54]: ▶ max_ind = np.argmax(reshaped, axis=1)
           row_ind = np.arange(reshaped.shape[0])
           multi_ind = np.array([row_ind, max_ind])
           linear_ind = np.ravel_multi_index(multi_ind, reshaped.shape)
           reshaped.reshape((-1))[linear_ind] = 0
           reshaped

  Out[54]: array([[ 9,  0,  0,  3,  2],
                  [ 0,  0, 12,  7,  0],
                  [ 1,  1,  0,  1,  0]])
```

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

[[3 -2]

[1 0]]

```
In [67]:  ▶ m = np.array([[3, -2],
                          [1, 0]])

In [68]:  ▶ w, v = np.linalg.eig(m)

In [69]:  ▶ print("Eigen values of the given square array:\n",w)

             Eigen values of the given square array:
              [2. 1.]

In [70]:  ▶ print("Right eigenvectors of the given square array:\n",v)

             Right eigenvectors of the given square array:
              [[0.89442719 0.70710678]
               [0.4472136  0.70710678]]
```

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

[[0 1 2]

[3 4 5]]

```
In [73]:  ▶ n_array = np.array([[0, 1, 2],
                                [3, 4, 5]])

             trace = np.trace(n_array)
             print("\nTrace of given matrix:")
             print(trace)

             Trace of given matrix:
             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]]

Reshape 2x3:

[[1 2 3]

[4 5 6]]

```
In [73]:  ▶ n_array = np.array([[0, 1, 2],
                                [3, 4, 5]])

            trace = np.trace(n_array)
            print("\nTrace of given matrix:")
            print(trace)
```

```
Trace of given matrix:
4
```

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

```
In [78]:  ▶ import matplotlib.pyplot as plt

            programming_languages = 'Java', 'Python', 'PHP', 'JavaScript', 'C#', 'C++'
            popularity = [22.2, 17.6, 8.8, 8, 7.7, 6.7]
            colors = ["#1f77b4", "#ff7f0e", "#2ca02c", "#d62728", "#9467bd", "#8c564b"]
            # exploding the first slice
            explode = (0.1, 0, 0, 0,0,0)

            plt.pie(popularity, explode=explode, labels=programming_languages, colors=colors,
            autopct='%1.1f%%', shadow=True, startangle=140)

            plt.axis('equal')
            plt.show()
```