

# **MACHINE LEARNING (CS-5710)**

## **ASSIGNMENT - 1**

**Name: Yukthi Sravani Tummalapalli**

**Student ID: 700746296**

**Github link:**

[https://github.com/yukthi16/Machine\\_Learning\\_Assignment1/tree/main](https://github.com/yukthi16/Machine_Learning_Assignment1/tree/main)

**Video Link:**

<https://drive.google.com/file/d/1W422BBy6xWkenJQ3DX9SYMaJhUdie9mj/view?usp=sharing>

## 1. Numpy:

a. Using NumPy creates 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.

```
#a.1, a.2
arr=np.random.randint(1,20,15)
#reshaping array
arr=arr.reshape(3,5)
#printing array shape
arr.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.

```
#a.3
print(arr)
#replacing maximum elements with zero
arr[np.arange(len(arr)), arr.argmax(1)] = 0
arr
#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.
arr1=np.random.randint(1,20,(4,3),dtype='int32')
print(arr1.shape)
print(type(arr1))
print(arr1.dtype)
```

b. Write a program to compute the eigenvalues and right eigenvectors of a given square array given below:  $\begin{bmatrix} 3 & -2 \\ 1 & 0 \end{bmatrix}$

```
#b
arr2=np.array([[3,-2],[1,0]])
#eigen values and eigen vectors
ev, evec = np.linalg.eig(arr2)
print(ev)
```

```
print (evec)
```

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

```
#c
arr3=np.array([[0,1,2],[3,4,5]])
#sum of diagonal elements
print(np.trace(arr3))
```

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]]`

```
#d
arr4=np.array([1,2,3,4,5,6])
#reshaping to 3*2
arr5=arr4.reshape(3,2)
print(arr5)
print()
#reshaping to 2*3
arr6=arr4.reshape(2,3)
print(arr6)
```

## 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

```
def func(pct, allvalues):
    absolute = int(pct / 100.*np.sum(allvalues))
    return "{:.1f}%\n({:d} g)".format(pct, absolute)

data=[22.2, 17.6, 8.8, 8, 7.7, 6.7]
languages=['Java', 'Python', 'PHP', 'JavaScript', 'C#', 'C++']
cols=['blue','orange','green','red','magenta','brown']
ex=(0.2, 0.0, 0.0, 0.0, 0.0, 0.0)
wp = { 'linewidth' : 1, 'edgecolor' : "black" }
fig, ax = plt.subplots(figsize =(8, 5))
w,t,at=ax.pie(data, explode=ex, labels=languages, colors=cols,
autopct=lambda pct: func(pct, data), shadow=True,startangle = 135,
            wedgeprops = wp,
            textprops = dict(color ="black"))
plt.show()
```

