

ML ASSIGNMENT 1

NAME : SAI SNUSHA NAKKA

ID : 700746287

GitHub Link- https://github.com/NSnusha/ML_Assignment1

Video Link -

https://drive.google.com/file/d/1IfyjO_ypEAqJsl2iuOYJbd4Z3y_zKTj0/view?usp=sharing

```
In [26]: #Question 1a
import numpy as np
integer_vec = np.random.randint(low=1, high=21, size=15)
arTest = integer_vec.reshape((3, 5))
print("Array shape:", arTest.shape)
arTest[np.arange(arTest.shape[0]), np.argmax(arTest, axis=1)] = 0
print("Modified array:\n", arTest)

print("Array Shape:", arTest.shape)
print("Type of the array:", type(arTest))
print("Data type of the array:", arTest.dtype)
#We use random.randint present in numpy to create the random values Then we use the argmax, arrange functions

Array shape: (3, 5)
Modified array:
[[ 3  0  9  4  1]
 [10  5  0  1  7]
 [ 6 11 18  0  5]]
Array Shape: (3, 5)
Type of the array: <class 'numpy.ndarray'>
Data type of the array: int32
```

```
In [19]: #Question 1.b
import numpy as np

# Define the square array
A = np.array([[3, -2], [1, 0]])

# Compute the eigenvalues and right eigenvectors
eigenvalues, eigenvectors = np.linalg.eig(A)

# Print the results
print("Eigenvalues:", eigenvalues)
print("Right eigenvectors:\n", eigenvectors)
#We use linalg.eig to compute the eigen value and right eigen vector

Eigenvalues: [2. 1.]
Right eigenvectors:
[[0.89442719 0.70710678]
 [0.4472136  0.70710678]]
```

```
In [20]: #Question 1.c
import numpy as np

# Define the array
input = np.array([[0, 1, 2], [3, 4, 5]])

# Compute the sum of the diagonal elements
diagonalSum = np.trace(input)

# Print the result
print("Sum of diagonal elements:", diagonalSum)
#In this we use the trace function to find the sum of diagonal element

Sum of diagonal elements: 4
```

```
In [11]: #Question 1.d
import numpy as np

# Create a NumPy array
arr = np.array([[1, 2], [3, 4], [5, 6]])

# Reshape the array to 2x3 shape without changing data
new_arr = arr.reshape(2, 3)

# Print the original and new array
print("Original array:")
print(arr)
print("\nNew array:")
print(new_arr)
#Reshape function from the numpy module is used to reshape the array

Original array:
[[1 2]
 [3 4]
 [5 6]]

New array:
[[1 2 3]
 [4 5 6]]
```

```
In [12]: #Question 2
import matplotlib.pyplot as plt
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"]
# explode 1st slice
explode = (0.1, 0, 0, 0, 0, 0)
# Plot
plt.pie(popularity, explode=explode, labels=languages, colors=colors,
autopct='%1.1f%%', shadow=True, startangle=140)

plt.axis('equal')
plt.show()
#Here we use matplotlib.pyplot and use to pie function to generate the pie chart of given data.
```

