Machine Learning (ICP#3)

Name: Preksha Reddy

700#: 700765245

https://colab.research.google.com/drive/1tE5D8fe3pBGQ49tWjfFiSuxBej8H4Q8T#scrollTo=HgTdczRxIZKS

Question 1:

```python
import numpy as np

# Create a random vector of size 15 with integers in the range 1-20
random_vector = np.random.randint(1, 20, size=15)
print(random_vector)

# Reshape the array to 3 by 5
reshaped_array = random_vector.reshape(3, 5)

# Print array shape
print("Array shape before replacing max values:")
print(reshaped_array.shape)

# Replace the max in each row by 0
max_indices = np.argmax(reshaped_array, axis=1)
for i in range(len(reshaped_array)):
    reshaped_array[i, max_indices[i]] = 0

# Print array shape after replacing max values
print("\nArray shape after replacing max values:")
print(reshaped_array.shape)

# Create a 2-dimensional array of size 4 x 3 (composed of 4-byte integer elements)
#array_2d = np.zeros((4, 3), dtype=np.int32)
array_2d = np.random.randint(low=-100, high=100, size=(4, 3),dtype=np.int32)
print(array_2d)

# Print shape, type, and data type of the array
print("\nArray shape:", array_2d.shape)
print("Array type:", type(array_2d))
print("Array data type:", array_2d.dtype)
```

Output:

```
[ 4 10 17 16  8 10  4  2  3  7 19 16  3  6  9]
Array shape before replacing max values:
(3, 5)

Array shape after replacing max values:
(3, 5)
[[ 74  74  -6]
 [-53  99  53]
 [-42  99 -65]
 [ 95 -87  27]]

Array shape: (4, 3)
Array type: <class 'numpy.ndarray'>
Array data type: int32
```

Question 2:

```python
#Question 2

import numpy as np

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

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

# Print the eigenvalues and right eigenvectors
print("Eigenvalues:")
print(eigenvalues)
print("\nRight Eigenvectors:")
print(eigenvectors)
```

Output:

```
Eigenvalues:
[2. 1.]

Right Eigenvectors:
[[0.89442719 0.70710678]
 [0.4472136  0.70710678]]
```

Question 3:

```python
[7] #Question 3
import numpy as np

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


# Compute the sum of the diagonal elements
diagonal_sum = np.trace(array)

# Print the sum
print("Sum of diagonal elements:", diagonal_sum)
```

Output:

```
Sum of diagonal elements: 4
```

Question 4:

```python
#Question 4

import numpy as np

# Define the arrays
array1 = np.array([[1, 2], [3, 4], [5, 6]])  # 3x2 array
array2 = np.array([[1, 2, 3], [4, 5, 6]])    # 2x3 array

# Reshape array1 to 2x3 without changing its data
reshaped_array1 = np.reshape(array1, (2, 3))

# Reshape array2 to 3x2 without changing its data
reshaped_array2 = np.reshape(array2, (3, 2))

# Print the reshaped arrays
print("Reshaped array1 (2x3):")
print(reshaped_array1)
print("\nReshaped array2 (3x2):")
print(reshaped_array2)
```

Output:

```
Reshaped array1 (2x3):
[[1 2 3]
 [4 5 6]]

Reshaped array2 (3x2):
[[1 2]
 [3 4]
 [5 6]]
```