Lab1

July 24, 2025

Machine Learning Lab #1

Working with numpy, scify, and CSV file format

[2]:

```
#generic imports for all programs

import numpy as np
import pandas as pd
import sklearn as sk
import matplotlib
```

[17]:

```
'''
1. Convert a 1-D array into a 2-D array with 3 rows.
'''

Assign1 = np.array([0, 1, 2, 3, 4, 5, 6, 7, 8]).reshape([3,3])
print(Assign1)
```

```
[[0 1 2]
 [3 4 5]
 [6 7 8]]
```

[26]:

```
'''
2. Replace all odd numbers in the given array with -1
'''

Assign2 = np.array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])

# for num in range(len(Assign2)-1):
#     if num % 2 != 0:
#         num == -1
# print(Assign2)

cond = Assign2 % 2 != 0
Assign2[cond] = -1
print(Assign2)
```

```
[ 0 -1  2 -1  4 -1  6 -1  8 -1]
```

[ ]:

```
'''
3. Find the positions of elements in x where its value is more than its␣
 ↪corresponding element in y, and elements in x where its value is
equals to its corresponding element in y.
'''

x = np.array([21, 64, 86, 22, 74, 55, 81, 79, 90, 89])
y = np.array([21, 7, 3, 45, 10, 29, 55, 4, 37, 18])

greater = np.where(x > y)[0]
equal = np.where(x == y)[0]
print(greater)
print("\n",equal)
```

```
[1 2 4 5 6 7 8 9]

 [0]
```

[67]:

```
'''
4. Extract the first four columns of this 2-D array
'''

Assign4= np.arange(100).reshape(5,-1)
a = np.arange(100).reshape(5,-1)
arr = a[:,:4]
print(arr)
```

```
[[ 0  1  2  3]
 [20 21 22 23]
 [40 41 42 43]
 [60 61 62 63]
 [80 81 82 83]]
```

Additional Questions:

[11]:

```
#Code for subsequent questions

a = np.array([1, 2, 3, 4, 5, 6, 7, 8,10]).reshape([3,3])
print(a)

b = np.array([7,8,10, 4,5,6,1,2,3]).reshape([3,3])
print("\n",b)
```

```
[[ 1  2  3]
 [ 4  5  6]
 [ 7  8 10]]

 [[ 7  8 10]
 [ 4  5  6]
 [ 1  2  3]]
```

[ ]:
```
'''
1. Generate a 1-D array of 10 random integers. Each integer should be a number␣
 ↪between 30 and 40 (inclusive).
'''
lmao = np.random.randint(low= 30, high=40, size=10)
print(lmao)
```

```
[34 31 37 35 37 35 31 32 36 37]
```

[ ]:
```
'''
i) Add and Subtract of the Matrix A and B, print the resultant matrix C for add␣
 ↪and E for subtract.
'''


c = a + b
print("Sum Matrix: \n",c)

e = a - b
print("\n Difference Matrix: \n",e)
```

```
Sum Matrix:
 [[ 8 10 13]
 [ 8 10 12]
 [ 8 10 13]]

 Difference Matrix:
 [[-6 -6 -7]
 [ 0  0  0]
 [ 6  6  7]]
```

[ ]:
```
'''
ii) Compute the sum of all elements of Matrix A, sum of each column of Matrix B␣
 ↪and sum of each row of Matrix C
'''


sumA = np.sum(a)
sumcolB = np.sum(b, axis=0)
c_sum = np.sum(c, axis=1)

print(sumA, sumcolB, c_sum)
```

```
4950 [12 15 19] [31 30 31]
```

[ ]:
```
'''
iii) Product of two matrices A and B, and print the resultant matrix D
'''
```

```
d = a*b
print("Product Matix: \n", d)
```

```
Product Matix:
 [[ 7 16 30]
 [16 25 36]
 [ 7 16 30]]
```

[ ]:

```
'''
iv) Sort the elements of resultant matrix C and print the resultant Matrix E.
'''

C_sort = np.sort(c.flatten()).reshape(c.shape)
print(C_sort)
```

```
[[ 8  8  8]
 [10 10 10]
 [12 13 13]]
```

[ ]:

```
'''
v) Transpose the Matrix E and print the result
'''
C_trans = C_sort.T
print(C_trans)
```

```
[[ 8 10 12]
 [ 8 10 13]
 [ 8 10 13]]
```