Assignment_1_BeetaSamad_181210016

January 18, 2021

1 CSB352: Data Mining

Instructor : [Dr. Chandra Prakash]

- For more information visit the class website.

2 LAB_1_Assignment : Introduction to Python

Due Date: 09-Jan-2021

Student Name: Beeta Samad

3 Assignment Instructions

You must save your as Assignment_NO_Yourname

Assignment 1 : To complete assignment one, first you have to add your name and roll no in the Google Colab Instructions section below and print it. Perform the 10 task given in the assignment and submit it over Microsoft Team.

Your source file will most likely end in **.pynb** if you are using a Jupyter notebook; however, it might also end in **.py** if you are using a Python script.

4 Google CoLab Instructions

The following code ensures that Google CoLab is running the correct version of TensorFlow.

[2]:

```
from datetime import datetime

try:
    from google.colab import drive
    %tensorflow_version 2.x
    COLAB = True
    print("Hello World")
    print("Note: using Google CoLab")
except:
    print("Hello NITD")
    print("Note: not using Google CoLab")
    COLAB = False
```

```python

# Print your name and Roll No.
print("My name is Beeta Samad")
print("Roll number is: 181210016")

# Print the curent time
now = datetime.now().time()
print("The current time is: ", now)
```

```
Hello NITD
Note: not using Google CoLab
My name is Beeta Samad
Roll number is: 181210016
The current time is:  14:50:22.598158
```

Exercise 1. Create following list:

data = [2,34,23,16,56,45,34,26,78,56,1,16]

Write a function to count number of elements in a list . Now delete the duplicate values and print the list in ascending order.

[5]:
```python
#Write your code here

data = list([2,34,23,16,56,45,34,26,78,56,1,16])

def countElements(data):
    print("The number of elements in the list are: ", len(data))

def deleteDuplicates(data):
    new_data = list(set(data))
    new_data.sort()
    print("The unique elements sorted are: ", new_data)

countElements(data)
deleteDuplicates(data)
```

```
The number of elements in the list are:  12
The unique elements sorted are:  [1, 2, 16, 23, 26, 34, 45, 56, 78]
```

Exercise 2: Take five numbers as input from the user and save into a list. Find the maximum of the list and sort the data in descending order.

[6]:
```python
#Write your code here
data = list([])

print("Enter 5 numbers: ")
for x in range(5):
    try:
        inputted_int = int(input())
    except:
        print("It is not a number!")
    data.append(inputted_int)

data.sort(reverse=True);
print("The maximum number is: ", data[0]);

print("The sorted (descending) form of the list is: ", data)
```

```
Enter 5 numbers:
7
3
1
5
9
The maximum number is:  9
The sorted (descending) form of the list is:  [9, 7, 5, 3, 1]
```

Exercise 4:

(i) Generate two arrays A1 and A2 of size 5 X 4 and 3 X 4 respectively using np.random()

(ii) Join them and make an array A3 of 8 X 4. Now append random numbers ranging between from 0 to 5 to make the fourth array A4 of size 10 X 10.

(iii) Print all the arrays and their transpose (Transpose of ‘A’ can be obtained by ‘A.T’)

[16]:

```
#Write your code here

import numpy as np

A1 = np.random.random((5, 4))
A2 = np.random.random((3, 4))

print('A1 : ', A1)
print('\nA2 : ', A2)

A3 = np.concatenate((A1, A2), axis = 0)
print('\nA3 : ', A3)

A4 = np.append(A3, np.random.randint(0, 5, size = (8, 6)), axis = 1)
A4 = np.append(A4, np.random.randint(0, 5, size = (2, 10)), axis = 0)
print('\nA4 : ', A4)

print('\n Transpose of A1: ', A1.transpose())
print('\n Transpose of A2: ', A2.transpose())
print('\n Transpose of A3: ', A3.transpose())
print('\n Transpose of A4: ', A4.transpose())
```

```
A1 :  [[0.8251468  0.73584723 0.3401246  0.26113363]
```

```
 [0.67198466 0.18697015 0.33555648 0.86768553]
 [0.92531312 0.19533315 0.39591113 0.25036974]
 [0.53921811 0.66601695 0.43869129 0.34139659]
 [0.12205406 0.02245908 0.61679198 0.60892582]]

A2 :  [[0.70875066 0.38407994 0.68938539 0.14182264]
 [0.82263397 0.73532933 0.53119858 0.12568477]
 [0.25888442 0.69163086 0.3147132  0.63863194]]

A3 :  [[0.8251468  0.73584723 0.3401246  0.26113363]
 [0.67198466 0.18697015 0.33555648 0.86768553]
 [0.92531312 0.19533315 0.39591113 0.25036974]
 [0.53921811 0.66601695 0.43869129 0.34139659]
 [0.12205406 0.02245908 0.61679198 0.60892582]
 [0.70875066 0.38407994 0.68938539 0.14182264]
 [0.82263397 0.73532933 0.53119858 0.12568477]
 [0.25888442 0.69163086 0.3147132  0.63863194]]

A4 :  [[0.8251468  0.73584723 0.3401246  0.26113363 2.         0.
  1.         3.         1.         3.        ]
 [0.67198466 0.18697015 0.33555648 0.86768553 1.         1.
  3.         0.         3.         1.        ]
 [0.92531312 0.19533315 0.39591113 0.25036974 4.         3.
  2.         4.         4.         3.        ]
 [0.53921811 0.66601695 0.43869129 0.34139659 1.         3.
  1.         3.         4.         1.        ]
 [0.12205406 0.02245908 0.61679198 0.60892582 0.         0.
  4.         3.         4.         4.        ]
 [0.70875066 0.38407994 0.68938539 0.14182264 2.         3.
  3.         4.         0.         0.        ]
 [0.82263397 0.73532933 0.53119858 0.12568477 4.         2.
  2.         0.         3.         4.        ]
 [0.25888442 0.69163086 0.3147132  0.63863194 0.         3.
  4.         2.         2.         1.        ]
 [2.         2.         1.         4.         4.         1.
  0.         4.         0.         3.        ]
 [1.         1.         3.         3.         2.         4.
  1.         2.         1.         2.        ]]

 Transpose of A1:  [[0.8251468  0.67198466 0.92531312 0.53921811 0.12205406]
 [0.73584723 0.18697015 0.19533315 0.66601695 0.02245908]
 [0.3401246  0.33555648 0.39591113 0.43869129 0.61679198]
 [0.26113363 0.86768553 0.25036974 0.34139659 0.60892582]]

 Transpose of A2:  [[0.70875066 0.82263397 0.25888442]
 [0.38407994 0.73532933 0.69163086]
 [0.68938539 0.53119858 0.3147132 ]
 [0.14182264 0.12568477 0.63863194]]
 Transpose of A3:  [[0.8251468  0.67198466 0.92531312 0.53921811 0.12205406
0.70875066
  0.82263397 0.25888442]
 [0.73584723 0.18697015 0.19533315 0.66601695 0.02245908 0.38407994
  0.73532933 0.69163086]
 [0.3401246  0.33555648 0.39591113 0.43869129 0.61679198 0.68938539
  0.53119858 0.3147132 ]
 [0.26113363 0.86768553 0.25036974 0.34139659 0.60892582 0.14182264
  0.12568477 0.63863194]]

 Transpose of A4:  [[0.8251468  0.67198466 0.92531312 0.53921811 0.12205406
0.70875066
  0.82263397 0.25888442 2.         1.        ]
 [0.73584723 0.18697015 0.19533315 0.66601695 0.02245908 0.38407994
  0.73532933 0.69163086 2.         1.        ]
 [0.3401246  0.33555648 0.39591113 0.43869129 0.61679198 0.68938539
  0.53119858 0.3147132  1.         3.        ]
 [0.26113363 0.86768553 0.25036974 0.34139659 0.60892582 0.14182264
  0.12568477 0.63863194 4.         3.        ]
 [2.         1.         4.         1.         0.         2.
  4.         0.         4.         2.        ]
 [0.         1.         3.         3.         0.         3.
  2.         3.         1.         4.        ]
 [1.         3.         2.         1.         4.         3.
  2.         4.         0.         1.        ]
 [3.         0.         4.         3.         3.         4.
  0.         2.         4.         2.        ]
 [1.         3.         4.         4.         4.         0.
  3.         2.         0.         1.        ]
 [3.         1.         3.         1.         4.         0.
  4.         1.         3.         2.        ]]
```

Exercise 5: Create two dictionaries.

The first dictionary ‘name’ will contain first name(key) of a person and its hash value(value). The Second will contain hash value(key) and mobile no(value).

i) Add 5 entries.

ii) Delete two entries by taking the input from user as the first name.

iii) Add two entries by taking the input as the first name and mobile no.

Hint: You can use remainder (%) to obtain hash value.

[8]:

```
#Write your code here
name = {'Beeta' : 1, 'Samyak' : 2, 'Ankit' : 3, 'Yash' : 4, 'Ritika' : 5}
mobile = {1 : 7303077410, 2 : 72211113113, 3 : 2131312231, 4 : 1234567890, 5 :␣
 ↪5131522341}
```

```python
print('\'Name\' dictionary: ')
for item, value in name.items():
  print(item, value)

print('\'Mobile\' dictionary: ')
for item, value in mobile.items():
  print(item, value)

item_del = input('\nEnter the first name to delete:  ')
value_del = name[item_del]
del name[item_del]
del mobile[value_del]

item_del = input('\nEnter the second name to delete:  ')
value_del = name[item_del]
del name[item_del]
del mobile[value_del]

item_add = input('\nEnter the name to add:  ')
value_add = input('\nEnter the mobile number of this person:  ')
hash_val = int(value_add) % 100

name[item_add] = hash_val
mobile[hash_val] = value_add

item_add = input('\nEnter the second name to add:  ')
value_add = input('\nEnter the mobile number of this person:  ')
hash_val = int(value_add) % 100

name[item_add] = hash_val
mobile[hash_val] = value_add

print('\nFinal name dictionary: ')
for item, value in name.items():
  print(item, value)

print('\nFinal mobile dictionary: ')
for item, value in mobile.items():
  print(item, value)
```

```
'Name' dictionary:
Beeta 1
Samyak 2
Ankit 3
Yash 4
Ritika 5
'Mobile' dictionary:
```

```
1 7303077410
2 72211113113
3 2131312231
4 1234567890
5 5131522341

Enter the first name to delete:  Beeta

Enter the second name to delete:  Ritika

Enter the name to add:  Kevin

Enter the mobile number of this person:  123213123123

Enter the second name to add:  SaiDheeraj

Enter the mobile number of this person:  213131312313

Final name dictionary:
Samyak 2
Ankit 3
Yash 4
Kevin 23
SaiDheeraj 13

Final mobile dictionary:
2 72211113113
3 2131312231
4 1234567890
23 123213123123
13 213131312313
```

5 PART 2: Introduction to NumPy

Exercise 6: Write a NumPy program to create an element-wise comparison (greater, greater_equal, less and less_equal) of two given arrays

[11]:

```python
#Write your code here

import numpy as np

x = np.array([13, 8, 23, 6, 4])
y = np.array([55, 1, 34, 22, 19])

print("The original arrays are:")
print(x)
print(y)

print("\nGreater comparison: ")
print(np.greater(x, y))

print("\nLesser comparison: ")
print(np.less(x, y))

print("\nGreater or Equal to comparison: ")
print(np.greater_equal(x, y))

print("\nLess than or Equal to comparison: ")
print(np.less_equal(x, y))
```

```
The original arrays are:
[13  8 23  6  4]
[55  1 34 22 19]

Greater comparison:
[False  True False False False]

Lesser comparison:
[ True False  True  True  True]

Greater or Equal to comparison:
[False  True False False False]

Less than or Equal to comparison:
[ True False  True  True  True]
```

Exercise 7: Write a NumPy program to create an array with the values 100, 71, 113, 1050 and determine the size of the memory occupied by the array.

[12]:

```python
#Write your code here

x = np.array([1, 2, 3, 4])

print("Length/size of the array: ", x.size)

print("\nMemory size of one of the elements in the array (in bytes): ", x.
 ↪itemsize)

print("\nMemory size of the entire numpy array (in bytes): ", x.size * x.
 ↪itemsize)
```

```
Length/size of the array:  4

Memory size of one of the elements in the array (in bytes):  4

Memory size of the entire numpy array (in bytes):  16
```

Exercise 8: Write a NumPy program to get the powers (x ^ 3) of an array values element-wise

Expected Output: Original array [1 2 3 4 5] Output array: [1 8 27 64 125]

[13]:

```
#Write your code here

a = np.array([1, 2, 3, 4, 5])

b = np.power(a, 3)

print(b)
```

```
[  1   8  27  64 125]
```

Exercise 9: Write a NumPy program to get the floor, ceiling and truncated values of the elements of a numpy array. Sample Output: Original array: [-1.3, -1.15, -0.1, 0.12, 1.7, 0.9, 1.1]

[14]:

```
#Write your code here

x = np.array([2.2, -1.2, 4.1, -3.3, 0.4])

a = np.floor(x)
b = np.ceil(x)
c = np.trunc(x)

print(a)
print(b)
print(c)
```

```
[ 2. -2.  4. -4.  0.]
[ 3. -1.  5. -3.  1.]
[ 2. -1.  4. -3.  0.]
```

Exercise 10: Write a program in python to display prime numbers from x to y (here x and y are user given values).

[15]:

```
#Write your code here

x = input('Enter the value of x: ')
y = input('Enter the value of y: ')

for i in range(int(x),int(y) + 1):
   if i > 1:
       for j in range(2, i):
           if (i % j) == 0:
               break
       else:
           print(i)
```

```
Enter the value of x: 3
Enter the value of y: 30
3
5
7
11
13
17
19
23
29
```

[ ]:

```

```

6 Observation Comments

[ ]:

```
#Write your observation\ Learning here

This exercise helped me revise the concepts/syntax of python. It also introduced me to the numpy package.
```

6.1 Hint :

Reference Book to learn python: https://anandology.com/python-practice-book/getting-started.html