**Department of Computer Science and Engineering**

|  |  |
| --- | --- |
| **Course Code:CSE422** | **Credits: 1.5** |
| **Course Name: Artificial Intelligence** | **Prerequisite:** CSE111, CSE221 |

**Lab 02  
Data preprocessing for Artificial Intelligence (Learning Numpy)**

1. **Lab Overview:**

Learn Python programming using numpy library for data preprocessing.

1. **Why Python for AI course:**

AI (artificial intelligence) opens up a world of possibilities. By taking advantage of machine learning or deep learning, you could produce many fascinating applications.But, which programming language should you use? You want a language havingwide range of well documented libraries and a large community of programmers. Hence, whatever you want to do can be found in web as a reference. Python has all these advantages.

1. **Lesson Fit:**

There is pre-requisite to this lab: CSE111, CSE221. You should have intensive Programming Knowledge and capability of understand algorithms.

1. **Acceptance and Evaluation**

Performed lab tasks will be evaluated by the Lab Instructor (LI)

* 1. Short viva will be conducted in each Lab or occasionally to examine your work.
  2. You may work in groups but be aware that you will be evaluated individually; hence active participation during the Lab work demonstration is recommended.
  3. There will be Lab handout after your work you have to handover it to LI

1. **Learning Outcome:**

After this Lab, the students will be able to:

* 1. Understand basic python codes
  2. Get an overview how to use python using numpy and pandas

1. **Activity Detail**
   1. **Hour: 1  
      Getting Started:**
      1. Have a glance at Books “Python code for Artificial Intelligence: Foundations of Computational Agents,” by David L. Poole and Alan K. Mackworth, May 28, 2018
      2. “Artificial Intelligence with Python written by Prateek Joshi, January 2017
      3. Check \\TSR to see e-book copy and codes, tutorials and useful links

**Running IDE and Code**

* + - 1. Run “anaconda navigator software 🡪 spyder IDE” from the root Icon which is shown in desktop.

**Operating system is Unix based then**

1. You have to set environment variable from Terminal, hence run Unix Terminal and execute the following command

export PATH="/opt/anaconda/anaconda3/bin:$PATH"

1. Then run command anaconda-navigator in the terminal
2. From anaconda navigator software run spyder IDE. You can also use Jupyter Notebook IDE which is also very popular among communities.
3. If your command anaconda-navigator shows command not found then either anaconda is not installed or installed in different location
4. “export PATH” This command means you are declaring that anaconda executable file is in /opt/anaconda/anaconda3/bin in this folder, Hence, it’s better to check whether anaconda is actually installed in /opt, using command “ls -ltr /opt/anaconda”

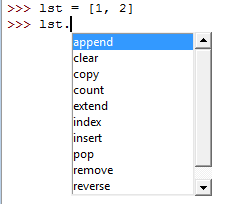
**Installing Library**

To install library use the following command in Terminal/windows command prompt “pip install libraryname”

**Get to know about basic python:**

1. Python List and tuple, function calling, if else scope, loops and many more

<https://www.programiz.com/python-programming/list>

1. Variable name followed by dot then press Tab to see the available option
2. **Negative indexing**, which counts backward from the end of the sequence. The expression seq[-1] yields the last element, seq[-2] yields the second to last, and so on.

**>>>**classmates=("Alejandro","Ed","Kathryn","Presila","Sean","Peter")

**>>>**classmates[-5]

'Ed'

**>>>**word="Alphabet"

**>>>**word[-3]

'b'

1. **Slicing**. Like with indexing, we use square brackets ([ ]) as the slice operator, but instead of one integer value inside we have two, separated by a colon (:):

**>>>**singers="Peter, Paul, and Mary"

**>>>**singers[0:5]

'Peter'

**>>>**classmates=("Alejandro","Ed","Kathryn","Presila","Sean","Peter")

**>>>**classmates[2:4]

('Kathryn', 'Presila')

'banana' stringThe operator [n:m] returns the part of the sequence from the n’th element to the m’th element, including the first but excluding the last.

If you omit the first index (before the colon), the slice starts at the beginning of the string. If you omit the second index, the slice goes to the end of the string. Thus:

**>>>**fruit ="banana"

**>>>**fruit[:3]

'ban'

**>>>**fruit[3:]

'ana'

What do you think s[:] means? It means all the elements

Negative indexes are also allowed, so

**>>>**fruit[-2:]

'na'

**>>>**classmates[:-2]

('Alejandro', 'Ed', 'Kathryn', 'Presila')

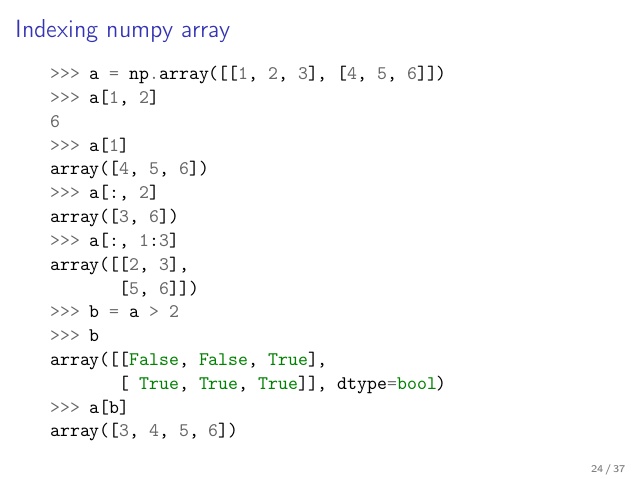
**>>>**fruit[-2:]

'na'

**>>>**classmates[:-2]

('Alejandro', 'Ed', 'Kathryn', 'Presila')

1. To handle multidimensional array Numpy array library is used, however library need to import first then using the “as” command you can create an instance for the library “Import numpy as np”

<https://jakevdp.github.io/PythonDataScienceHandbook/02.02-the-basics-of-numpy-arrays.html>

1. **Let’s create python array and np array.**

# python array  
a = [1,2,3,4,5,6,7,8,9]

# numpy array  
A = np.array([1,2,3,4,5,6,7,8,9])

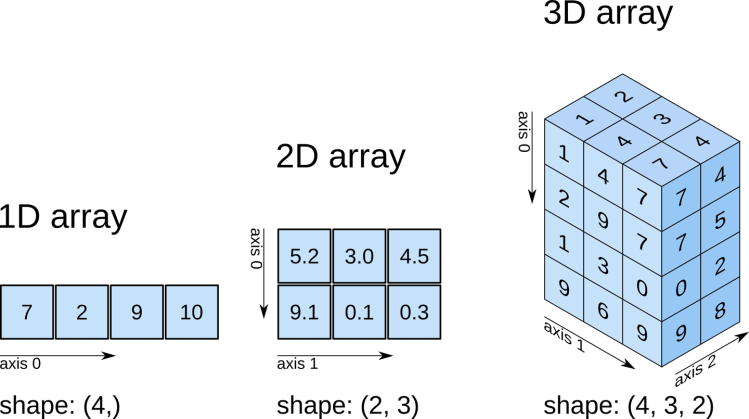
A = np.arange(0,10,2)  
A is array([0, 2, 4, 6, 8])

1. Shape is an attribute for np array. When a default array, say for example A is called with shape, here is how it looks.

A = [1, 2, 3, 4, 5, 6, 7, 8, 9]   
A.shape  
================================  
(9,)

1. This is a rank 1 matrix(array), where it just has 9 elements in a row. reshape() that changes the dimensions of your original matrix into your desired dimension.

A.reshape(1,9)  
====================================================================  
array([[1, 2, 3, 4, 5, 6, 7, 8, 9]])

1. Two square brackets in the beginning indicate that. [[1, 2, 3, 4, 5, 6, 7, 8, 9]] is a potentially multi-dim matrix as opposed to [1, 2, 3, 4, 5, 6, 7, 8, 9].
2. To understand array Shape write commandvariablename.shape or to change dimension write variablename.reshape(x value,y value,z value)
3. NumPy Array Attributes

**importnumpyasnp**

np.random.seed(0) *# seed for reproducibility*

x1 = np.random.randint(10, size=6) *# One-dimensional array*

x2 = np.random.randint(10, size=(3,4)) *# Two-dimensional array*

x3 = np.random.randint(10, size=(3,4,5)) *# Three-dimensional array*

Each array has attributes ndim (the number of dimensions), shape (the size of each dimension), and size (the total size of the array):

print("x3 ndim: ", x3.ndim)

print("x3 shape:", x3.shape)

print("x3 size: ", x3.size)

1. One-dimensional subarrays

x=np.arange(10)

x[::2]*# every other element*

array([0, 2, 4, 6, 8])

x[1::2]*# every other element, starting at index 1*

x[::-1]*# all elements, reversed*

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

1. Multi-dimensional subarrays

array([[12, 5, 2, 4],

[ 7, 6, 8, 8],

[ 1, 6, 7, 7]])

x2[:2,:3]*# two rows, three columns*

print(x2[0])*# equivalent to x2[0, :]*

1. Creating arrays: np.zeros((n,m)) returns an n x m matrix that contains zeros. It’s as simple as that

np.zeros((4,3))  
===============================  
array([[ 0., 0., 0.],  
 [ 0., 0., 0.],  
 [ 0., 0., 0.],  
 [ 0., 0., 0.]])

1. Concatenation of arrays

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

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

np.concatenate([x,y])

# Positions where value > 5

index\_gt5 = np.where(arr\_rand >5)

1. To read csv files using numpy

white\_wines = np.genfromtxt("winequality-white.csv", delimiter=";", skip\_header=1)

# if the data columns are separated with comma use following command, also this command will replace the non float point values with -999

data = np.genfromtxt(path, delimiter=',', skip\_header=1, filling\_values=-999, dtype='float')

# or you can just do the following if your dataset does not have any Text

data2 = np.genfromtxt(path, delimiter=',', skip\_header=1, dtype=None)

* 1. **Hour: 2-3**

You have to complete the task assigned in activities List

* + 1. Learning numpy arrays, built in functions
    2. Complete the given assigned task

**Activity List**

**Task 01:** Mark 10 **Time:** 1.5 Hour

Write a factorial function using recursive technique in python using **numpy** library, the program will take 5 input numbers from user, store them in a numpy array, then there will be a loop that will call recursive function and return factorial values. Then values will be printed sequentially

**Task 02:** Mark 7

Create a vector of size 10 with zeros. Then create a vector of size 10 with ones. Change their dimension and after that merge the the two vector vertically and horizontally using “concatenate”.

**Task 03:** Mark 3

How to read a file content given below using numpy?

# File content:

# -------------

1,2,3,4,5

6,,,7,8

,,9,10,11

# -------------

Hints: <https://www.machinelearningplus.com/python/101-numpy-exercises-python/>

**Evaluation Process (VIVA):** You have to explain your program to the Lab Instructor