

Introduction to Machine Learning Applications

Spring 2021

Lecture-4

Lydia Manikonda

manikl@rpi.edu



Rensselaer

Last class

- Python Basics – Recap and Loops, Conditionals, Functions
- Class exercises

Recap on Basics

- Python Notebook I shared for Lecture-3 Q&A

Today's agenda

- Numpy
- Pandas
- How does homework submissions work? – Demo

Numpy

Numpy

- Fundamental package for scientific computing
- Numpy is a general-purpose array-processing package
- Used for high-performance multidimensional array computations

Numpy – Arrays

- A numpy array is a grid of values, all values are of same type
- The number of dimensions is the **rank** of an array
- A tuple of integers giving the size of an array along each dimension is called the **shape** of an array

- Initialize using nested python lists
- Access using square brackets

Numpy – Arrays

- Declaring the package

```
import numpy as np
```

- Creating an array of rank 1

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

- Creating an array of rank 2

```
arr = np.array([1, 2, 3], [4, 5, 6])
```


Numpy – Arrays

- Create an array with rank 1

```
>> a = np.array([1, 2, 3])
```

- Print the shape of this array

```
>> print(a.shape)
```

```
>> (3, )
```

- Print the elements at different indices

```
>> print(a[0], a[1], a[2])
```

```
>> 1 2 3
```

- Change an element of the array

```
>> a[0] = 10
```

- Print the array

```
>> print(a)
```

```
>> [10, 2, 3]
```

Numpy – Arrays

```
>> a = np.array([[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 12], [13, 14, 15, 16]])  
>> print(a)
```

Using slicing method printing a range of array

```
>> sliced_a = a[:2, ::2]  
>> print(sliced_a)
```

Printing elements at specific indices

```
>> print(a[[1, 2, 1, 3],[1, 0, 2, 3]])
```

Numpy – Arrays and Functions

```
>> a = np.zeros((2, 2))  
>> print(a)  
>> [[0.  0.], [0.  0.]
```

```
>> b = np.ones((1, 2))  
>> print(b)  
>> [[1.  1.]
```

```
>> c = np.full((2,2), 7)  
>> print(c)  
>> [[7.  7.], [7.  7.]
```

```
>> d = np.eye(2)  
>> print(d)  
>> [[1.  0.], [0.  1.]
```

```
>> e = np.random.random((2, 2))  
>> print(e)  
>> [[,], [,]] #random values
```

Numpy – Arrays

- Datatypes of arrays need not be defined – numpy tries to guess the datatype

```
>> a = np.array([1.1, 2.2])
```

```
>> print(a.dtype)
```

```
>> a = np.array([1, 2], dtype=np.int64)
```

```
>> print(a.dtype)
```

Numpy – Math operations

```
>> a = np.array([[1, 2], [3, 4]], dtype=np.float64)
```

```
>> b = np.array([[4, 3], [2, 1]], dtype=np.float64)
```

```
>> sum_ab = np.add(a, b)
```

```
>> print(sum_ab)
```

```
>> sum_a = np.sum(a)
```

```
>> print(sum_a)
```

```
>> sqrt_a = np.sqrt(a)
```

```
>> print(sqrt_a)
```

```
>> trans_a = a.T
```

```
>> print(trans_a)
```

Numpy – Exercises

1. Given an array, print only a range of the array using slicing method.

Input: `[[-1, 2, 0, 4], [4, -0.5, 6, 0], [2.6, 0, 7, 8], [3, -7, 4, 2.0]]`

Output: `[[-1. 0.] [4. 6.]]`

2. Consider the array above and print elements at specific indices

Input: `[[-1, 2, 0, 4], [4, -0.5, 6, 0], [2.6, 0, 7, 8], [3, -7, 4, 2.0]];`

Values at these indices `[[1, 1, 0, 3], [3, 2, 1, 0]]`

Output: `[0., 6., 2., 3.]`

3. Add two given arrays; `a = np.array([[1, 2], [3, 4]])` `b = np.array([[4, 3], [2, 1]])`

Numpy – Exercises

4. Given a numpy array, find the datatype: `np.array([4.0, 9.0])`
5. Consider the previous array and perform the square root of an array.
6. Get unique values in a list using numpy `[10, 20, 10, 30, 40, 40]`
7. Multiply all the numbers in a given list using numpy `[1, 2, 3, 4]`
8. Create a random numpy array of 20 rows and 2 columns

Pandas

Python Notebook

Homework demo