

The background of the slide is a blurred image of a financial market data screen. It features various stock indices and their values in different colors (green for up, red for down). Visible text includes 'OMX18', 'OMX ICELAND 8', 'OMX18', 'OMX ICELAND 8', 'OMX18', 'OMX ICELAND 8', 'OMX18', 'OMX ICELAND 8', 'OMX18', 'OMX ICELAND 8'.

Numerical Python (Numpy)

The library made for scientific and mathematical computations

A look into IRIS dataset

- What do we see in the dataset ?
- What is the type of data present in the dataset?
- How can we perform operations on such a dataset?
- Need of libraries like Pandas, Numpy, matplotlib, scipy

Some mathematical entities which need refresher!

- Array - An array is a data structure that contains a group of elements – typically of same data type.
- Vector - Vectors are built from components, which are ordinary numbers. You can think of a vector as a list of numbers, and vector algebra as operations performed on the numbers in the list.

$\text{vec1} = (v1, v2, v3) ,$

$\text{Vec2} = (v1,$
 $v2,$
 $v3)$

More about vectors can be read here -

<https://machinelearningmastery.com/gentle-introduction-vectors-machine-learning/>

Some mathematical entities which need refresher!

- Matrix –

A matrix is an $m \times n$ array of scalars from a given field F .
The individual values in the matrix are called entries.

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 5 \\ 7 \\ -3 \end{bmatrix}$$

$$[6 \ -2 \ -5]$$

$$\begin{bmatrix} -1 & 0 & 2 \\ 0 & 4 & 7 \\ -3 & -4 & -2 \end{bmatrix}$$

$$\begin{bmatrix} i & 0 \\ 0 & 1-i \end{bmatrix}$$

What is Numpy?


- Numerical Python, popularly known as Numpy has been designed to carry out mathematical computations at a faster and easier rate.
- Further this library enriches the programming language Python by providing powerful data structures like multi dimensional arrays beyond matrices and linear arrays.
- Besides that, Numpy provides a large library of high level mathematical functions to operate on these structures.

How to install Numpy

- In command line
`pip install numpy`
- Anaconda distribution
`conda install numpy`

Why Numpy when we have “Lists” ?

- Python has inbuilt data structure “List” which is also technically an array which allows different data types.
- The answer to this question comes in following three aspects
 1. Size – Numpy data structures take less space
 2. Performance – They are inherently faster than lists.
 3. Functionality – Scipy and Numpy have optimized functions.
 4. Vectorization of the operations



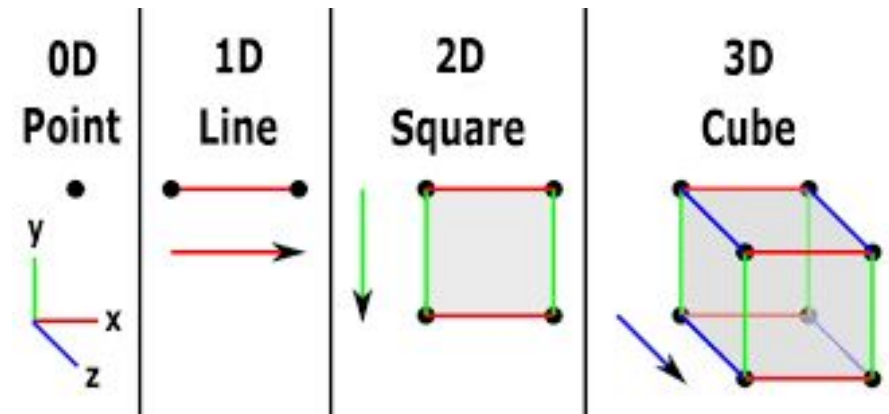
Now let's create some numpy
arrays and play around with
them!!

Nd-array object

- Nddarray is multidimensional object which can contain only single data type objects.
- It can be a string type or numeric or integer data type.
- If we mixture of strings and numbers are used, all are converted to strings.

Attributes of ND array object

- Dimension – It tells us the number of dimensions of the nd array object. Number of dimensions can range from 1 to 100s and 1000s
- Shape – It gives the shape of the nd array object. That is the length of each dimension.
- Size - Total number of elements in numpy array
- Dtype – It tells about the type of data being stored in the object.
- Strides – How many steps to be taken to move to next row!!



Some miscellaneous numpy arrays

- `Np.zeros()`
- `Np.arange()`
- `Np.linspace()`
- `Np.full()`

Resize, reshape , flatten and ravel

- Resize adds zeros if you want to create size larger than current one.

Note – `resize()` does not work on view of nd array but original one

- Reshape – reshapes array to any size and dimension
- Flatten and ravel – They help in flattening multidimensional array into a single array.



Array indexing and slicing operations

They are straight forward ways to manipulate data into numpy arrays. Let's see how to do it for simple toy numpy arrays

Lets manipulate numpy arrays!

- Arithmetic operations
- `Np.sin()`, `np.cos()`, `np.exp()`, `np.sqrt()`
- Comparison of numpy objects

`Np.array_equal()`

- Logical operations

`Np.logical_or()`

Broadcasting of numpy arrays

- To put it in a more practical context, you often have an array that's somewhat larger and another one that's slightly smaller. Ideally, you want to use the smaller array multiple times to perform an operation (such as a sum, multiplication, etc.) on the larger array.
- 1. First off, to make sure that the broadcasting is successful, the dimensions of your arrays need to be compatible.
- 2. Two dimensions are also compatible when one of them is 1

Matrix operations

- Numpy provides a range of functions to carry out various matrix operations
 - Addition
 - Matrix dot product
 - Matrix element wise multiplication
 - Matrix multiplication

Lets create arrays with random numbers

The source of randomness which we inject into machine learning projects is called Psuedo randomness.

1. `np.random.rand()`
2. `Np.random.randint()`
3. `Np.random.gaussian()`
4. `np.random.shuffle()`

Concatenate, append and stack numpy arrays

- Often, we might want to join different numpy arrays in different ways like column wise, row wise.
- Numpy offers a range of functions to do the same namely

`Np.append()`

`Np.concatenate()`

`Np.vstack()`

`Np.hstack()`

Let's visualize numpy arrays

- The numeric data in numpy arrays can be better interpreted with visualizations from matplotlib.
- `Np.histogram()`
- `Np.meshgrid()`

Let's save the nd array object to a file now!

- After all the processing and manipulation of the data, the most important step that comes is to save data into a file.
- Numpy provides `savetxt(<file name>,<nd_array_object>,<delimiter>)` function for the same.


Numpy quick start links

- Original documentation

<https://numpy.org/devdocs/user/quickstart.html>

- Cheat sheet

<https://www.datacamp.com/community/blog/python-numpy-cheat-sheet>



Thanks for taking part in today's session

The forum is open for QA!!