**NumPy**

NumPy is a Python library used for working with arrays.

NumPy was created in 2005 by Travis Oliphant. It is an open source project and you can use it freely.

NumPy stands for Numerical Python.

**Why Use NumPy?**

In Python we have lists that serve the purpose of arrays, but they are slow to process.

NumPy aims to provide an array object that is up to 50x faster than traditional Python lists.

The array object in NumPy is called ndarray, it provides a lot of supporting functions that make working with ndarray very easy.

Arrays are very frequently used in data science, where speed and resources are very important.

NumPy arrays are stored at one continuous place in memory unlike lists, so processes can access and manipulate them very efficiently.

**Installation of NumPy**

If you have Python and PIP already installed on a system, then installation of NumPy is very easy.

Install it using this command: **Pip install numpy**

Once NumPy is installed, import it in your applications by adding the import keyword: import **numpy** **as np**

**For example**

import numpy as np

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

print(arr)

**Checking the Data Type of an Array**

The NumPy array object has a property called dtype that returns the data type of the array:

import numpy as np

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

print(arr.dtype)

**Shape of an Array**

The shape of an array is the number of elements in each dimension.e.g.,

import numpy as np

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

print(arr.shape)

**Reshaping Arrays**

import numpy as np

arr = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12])

newarr = arr.reshape(4, 3)

print(newarr)

**Joining NumPy Arrays**

Joining means putting contents of two or more arrays in a single array.

import numpy as np

arr1 = np.array([1, 2, 3])

arr2 = np.array([4, 5, 6])

arr = np.concatenate((arr1, arr2))

print(arr)

Stack is same as concatenation, the only difference is that stacking is done along a new axis.(hstack() to stack along rows, vstack() to stack along columns.

**Splitting NumPy Arrays**

import numpy as np

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

newarr = np.array\_split(arr, 3)

print(newarr)

Split the array in 3 parts.

**Searching Arrays**

You can search an array for a certain value, and return the indexes that get a match.

To search an array, use the **where()** method, and **searchsorted()** which performs a binary search in the array, and returns the index.

**Sorting Arrays**

Sorting means putting elements in an ordered sequence.

import numpy as np

arr = np.array([3, 2, 0, 1])

print(np.sort(arr))

**Generate Random Number**

NumPy offers the random module to work with random numbers.

from numpy import random

x = random.randint(100)

print(x)

For float **rand()** method returns a random float between 0 and 1, The choice() method takes an array as a parameter and randomly returns one of the values.

**Visualize Distributions With Seaborn**

Seaborn is a library that uses Matplotlib underneath to plot graphs. It will be used to visualize random distributions.

import matplotlib.pyplot as plt

import seaborn as sns

sns.distplot([0, 1, 2, 3, 4, 5])

plt.show()

**Binomial Distribution**

Binomial Distribution is a Discrete Distribution.

It describes the outcome of binary scenarios, e.g. toss of a coin, it will either be head or tails.

from numpy import random

x = random.binomial(n=10, p=0.5, size=10)

print(x)

[random.uniform]

**Logistic Distribution**

Logistic Distribution is used to describe growth.

Used extensively in machine learning in logistic regression, neural networks etc.

It has three parameters:

Loc – mean, where the peak is. Default 0.

Scale – standard deviation, the flatness of distribution. Default 1. Size – The shape of the returned array

**What are ufuncs?**

Ufuncs stands for “Universal Functions” and they are NumPy functions that operate on the ndarray object..

Check if a function is a ufunc:

import numpy as np

print(type(np.add))

**Arithmetic Operations**

import numpy as np

arr1 = np.array([10, 11, 12, 13, 14, 15])

arr2 = np.array([20, 21, 22, 23, 24, 25])

newarr = np.add(arr1, arr2)

print(newarr)

[Same as subtract, multipliy, divide, remainder, divmod(for both quotient and mod)]

**Rounding Decimals**

There are primarily five ways of rounding off decimals in NumPy:

Truncation, Fix, Rounding, Floor, Ceil

import numpy as np

arr = np.trunc([-3.1666, 3.6667])

print(arr)

Addition is done between two arguments whereas summation happens over n elements.

**Finding LCM in Arrays**

To find the Lowest Common Multiple of all values in an array, you can use the reduce() method.

import numpy as np

arr = np.array([3, 6, 9])

x = np.lcm.reduce(arr)

print(x)

**Trignometric Function**

import numpy as np

x = np.sin(np.pi/2) print(x). [Same as cos, tan and for finding angles arcsin()]

For converting degree into radian (deg2rad() ).

To find the unique values of two arrays, use the union1d() method.

To find only the values that are present in both arrays, use the intersect1d() method.

To find only the values in the first set that is NOT present in the seconds set, use the setdiff1d() method.