Kmit

SONET

Data Science

(Level-I)

Ajeet K. Jain, M. Narsimlu

(ML TEAM)- SONET, KMIT, Hyderabad

Session - 19

This session deals with

Linear Algebra

System of Equations

Advantages of Numpy

Exercises on Numpy and Linear algebra

Usage of Linear Algebra

In a multiple regression problem we seek a function that can map input data points to outcome values..

Each data point is a *feature vector* (x_1 , x_2 , …, x_m) composed of two or more data values that capture various features of the input.

To represent all of the input data along with the vector of output values we set up a input matrix X and an output vector y:

SONET

Usage of Linear Algebra

DATA SCIENCE

$$X = \begin{bmatrix} 1 & x_{1,1} & x_{1,2} & \cdots & x_{1,m} \\ 1 & x_{2,1} & x_{2,2} & \cdots & x_{2,m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n,1} & x_{n,2} & \cdots & x_{n,m} \end{bmatrix}$$

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

we can compute a predicted outcome value

$$\hat{y} = \mathbf{x} \cdot \boldsymbol{\beta} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_m x_m$$

Usage of Linear Algebra

To compute the **β** vector as follows:

$$\boldsymbol{\beta} = (X^T X)^{-1} X^T \mathbf{y}$$

will use numpy to construct the appropriate matrices and vectors and solve for the β vector.

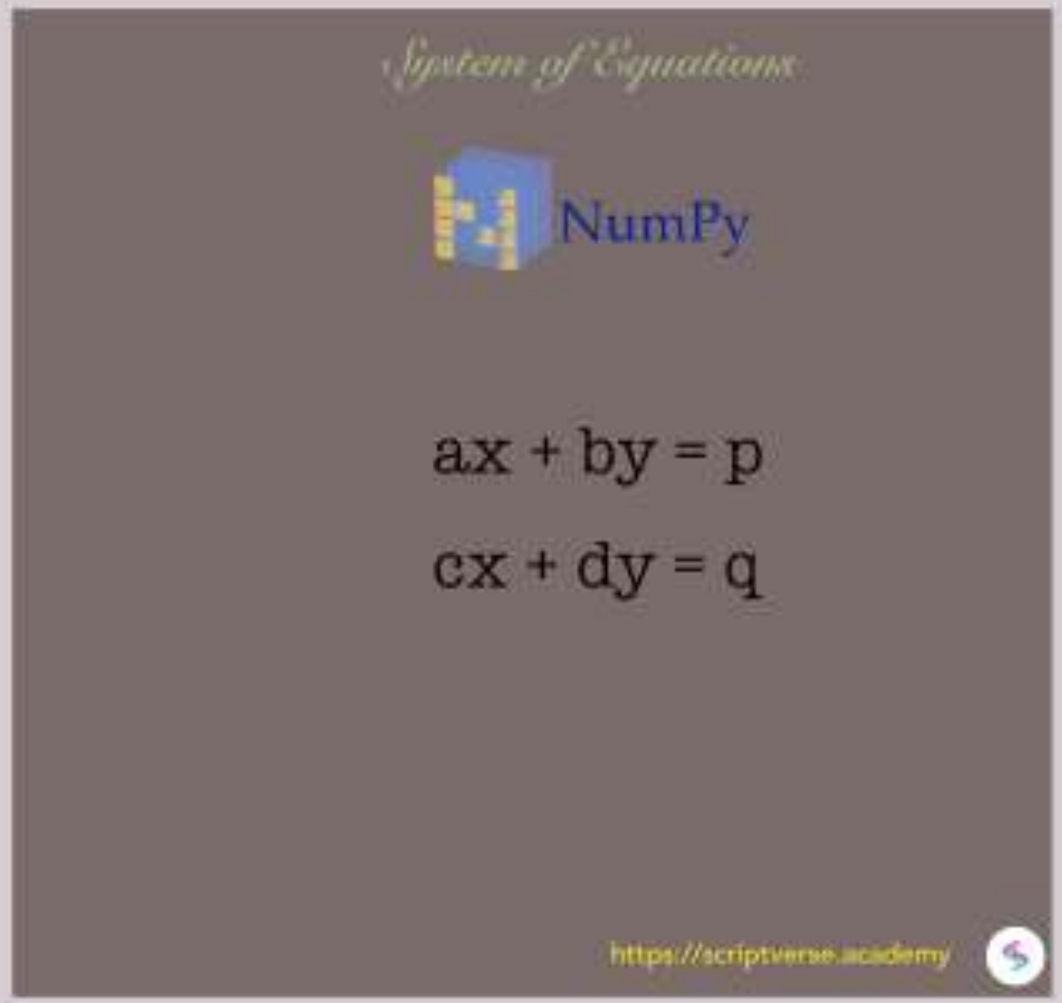
Once we have solved for β we will use it to make predictions for some test data points that we initially left out of our input data set.

Usage of Linear Algebra

constructed the input matrix X and the outcomes vector y in numpy, the following code will compute the β vector:

```
Xt = np.transpose(X)
XtX = np.dot(Xt,X)
Xty = np.dot(Xt,y)
beta = np.linalg.solve(XtX,Xty)
```

Solving System equations



Solving System equations

$$3x + 7y = 27$$
$$5x + 2y = 16$$

This can be put in the matrix dot product form as

$$\begin{bmatrix} 3 & 7 \\ 5 & 2 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 27 \\ 16 \end{bmatrix}$$

If A represents the matrix of coefficients, x the column vector of variables and B the column vector of solutions, the above equation can be shortened to

$$Ax = B$$

Solution

```
import numpy as np
a = np.array([[3,7], [5,2]])
b = np.array([27,16])
x = np.linalg.solve(a, b)
print(x)
```

Output

```
[2. 3.]
```

Solving System equations

As our practice, we will proceed with an example, first writing the matrix model and then using **Numpy** for a solution.

$$\begin{aligned} 2a + b + c &= 4 \\ a + 3b + 2c &= 5 \\ a &= 6 \end{aligned}$$

Now, we can formalize the problem with matrices:

$$A = \begin{bmatrix} 2 & 1 & 1 \\ 1 & 3 & 2 \\ 1 & 0 & 0 \end{bmatrix}, x = \begin{bmatrix} a \\ b \\ c \end{bmatrix}, B = \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix}$$

Then, the linear equations can be written this way:

$$Ax = B$$

to solve for the vector x, we must take the inverse of matrix A and the equation is written as follows:

$$x = A^{-1}B$$

Solution

```
import numpy as np
# define matrix A using Numpy arrays
A = np.array([[2, 1, 1],
 [1, 3, 2],
 [1, 0, 0]])
#define matrix B
B = np.array([4, 5, 6])
print("Solutions:\n",np.linalg.solve(A, B ))
```

Output

```
Solutions:
 [  6.  15. -23.]
```

Advantages of Numpy

Numpy is the core library for scientific computing in Python.

It provides a high-performance multidimensional array object.

The Python core library provided Lists.

A list is the Python equivalent of an array,
but is resizeable and can contain elements of different types.

Advantages of Numpy

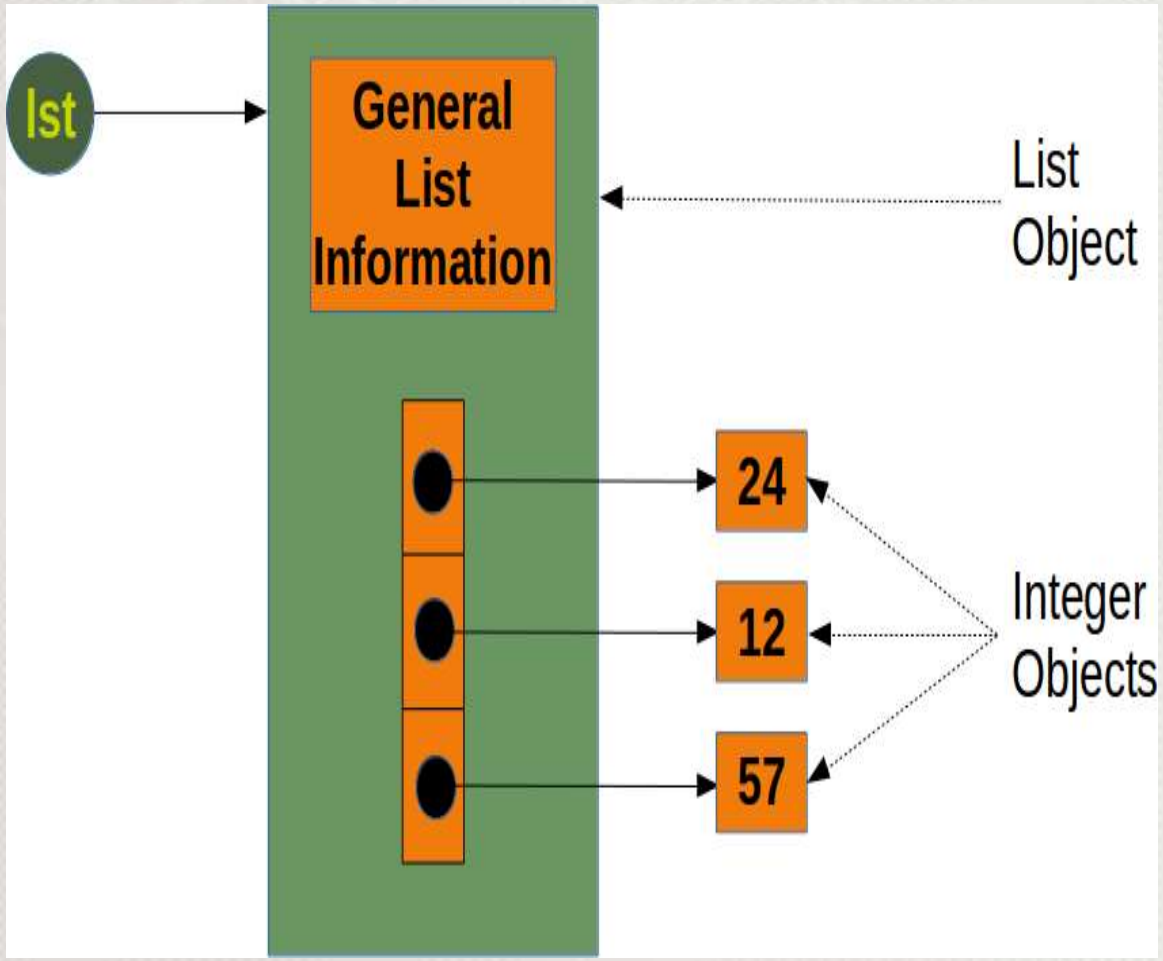
Numpy data structures perform better in:

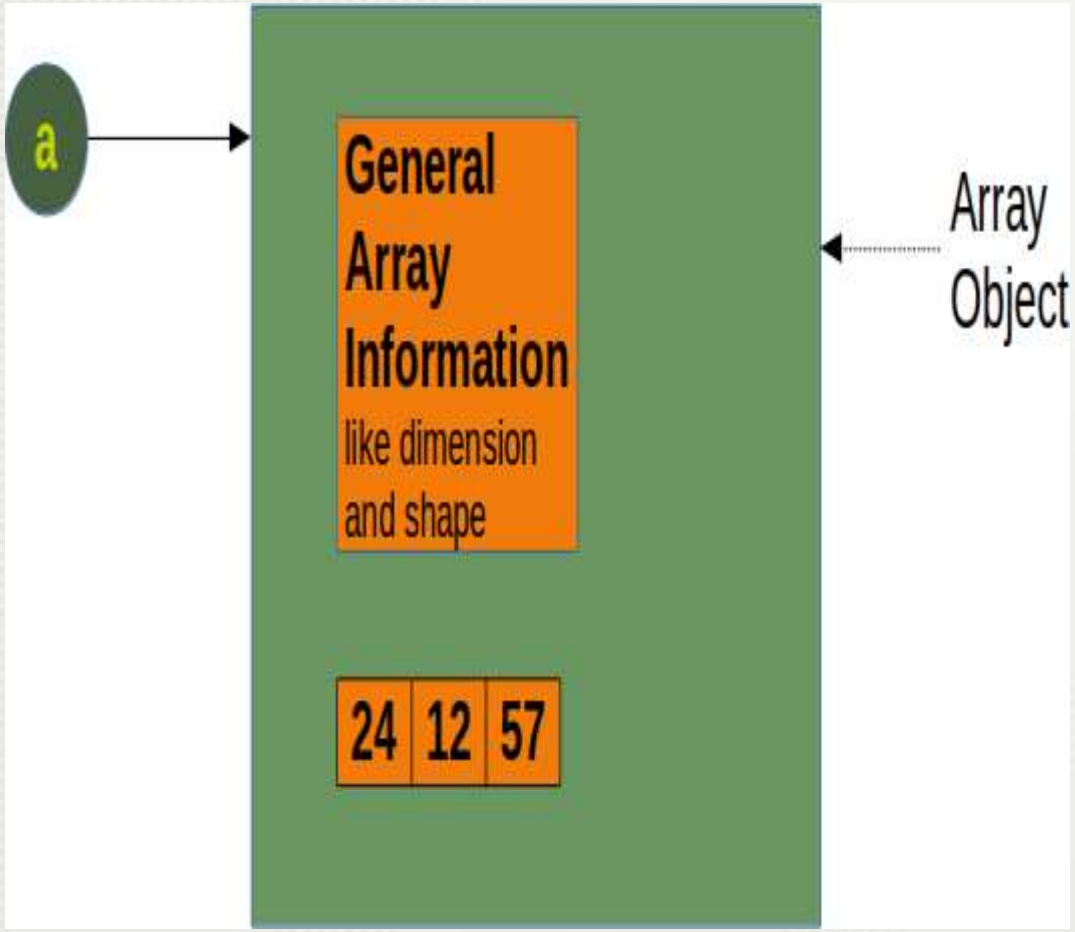
Size - Numpy data structures take up less space

Performance - they have a need for speed and are faster than lists

Functionality - SciPy and NumPy have optimized functions such as linear algebra operations built in.

Memory


lst
General List Information
List Object
24
12
57
Integer Objects


a
General Array Information
like dimension and shape
Array Object
24 12 57

Speed

Numpy arrays are faster than Lists, We can calculate run time speed of list using timeit module

```
In [1]: l1=range(1000)

In [2]: timeit sum(l1)
28.7 µs ± 1.1 µs per loop (mean ± std. dev. of 7 runs, 10000 loops each)
```

```
In [1]: import numpy as np
   ...: y=np.array(range(1000))

In [2]: timeit np.sum(y)
4.45 µs ± 258 ns per loop (mean ± std. dev. of 7 runs, 100000 loops each)
```

Storage Space

Numpy arrays will take less space than Lists

```
In [1]:  import sys

In [2]:  x=range(1000)

In [3]:  sys.getsizeof(1)*len(x)
Out[3]:  28000
```

```
In [1]:  import numpy as np

In [2]:  y=np.array(range(1000))

In [3]:  y.itemsize*y.size
Out[3]:  4000
```

Exercise-1

The table below provides the population of daily order volumes for a recent week. Calculate the mean, variance, and standard deviation of this population.

Day	Order Volume
1	16
2	10
3	15
4	12
5	11

Solution

```
import numpy as np
ordervolume=[16,10,15,12,11]
arr=np.array(ordervolume)
print(arr)
print("mean = ",np.mean(arr))
print("variance = ",np.var(arr))
print("standard deviation = ",np.std(arr))
```

Exercise-2

A well-known manufacturer of sugarless food products has invested a great deal of time and money in developing the formula for a new kind of sweetener. Although costly to develop, this sweetener is significantly less expensive to produce than the sweeteners the manufacturer had been using. The manufacturer would like to know if the new sweetener is as good as the traditional product. The manufacturer knows that when consumers are asked to indicate their level of satisfaction with the traditional sweeteners, they respond that on average their level of satisfaction is 5.5.The manufacturer conducts market research to determine the level of acceptance of this new product. Consumer taste acceptance data are collected from 25 consumers of sugarless products. The data collected can be seen in the table below.

Exercise-2

Consumers	Satisfaction
1	5.00
2	6.00
3	7.00
4	5.00
5	6.00
6	5.00
7	7.00
8	4.00
9	5.00
10	5.00
11	6.00
12	6.00
13	7.00
14	5.00

Consumers	Satisfaction
15	5.00
16	7.00
17	5.00
18	6.00
19	6.00
20	7.00
21	7.00
22	7.00
23	6.00
24	5.00
25	6.00

Solution

```
import numpy as np
import statistics as ss
list_sati=[5,6,7,5,6,5,7,4,5,5,6,6,7,5,5,7,5,6,6,7,7,7,6,5,6]
print(len(list_sati))
list_satisifaction=np.array(list_sati,dtype=float)
print("Mean = ",np.mean(list_satisifaction))
print("Median = ",np.median(list_satisifaction))
print("Variance = ",np.var(list_satisifaction))
print("Standard deviation = ",np.std(list_satisifaction))
```

Exercise-3

The table below identifies the number of consultants of a financial consulting firm for each of its five largest U.S. offices. For this data set, calculate the mode, median, mean, variance, and standard deviation.

City	Number of Consultants
New York	75
Chicago	60
San Francisco	50
Los Angeles	50
Atlanta	70

Solution

```
import numpy as np
import statistics as ss
no_of_consultants=[75,60,50,50,70]
print("Mean",np.mean(no_of_consultants))
print("Mode",ss.mode(no_of_consultants))
print("Median",np.median(no_of_consultants))
print("Variance",np.var(no_of_consultants))
print("Standard deviation",np.std(no_of_consultants))
```

Output

```
Mean 61.0
Mode 50
Median 60.0
Variance 104.0
Standard deviation
10.198039027185569
```

Exercise-4

Write a NumPy program to create a vector with values ranging from 15 to 55

a. print all values except the first and last.
b. Insert a number 55 at 11th position
c. Find out the minimum element from the Vector
d. Find out the maximum element from the Vector

Solution

```
import numpy as np
v = np.arange(15,55)
print("Original vector:")
print(v)
print("All values except the first and last of the said vector:")
print(v[1:-1])
print("after insertion")
print(np.insert(v,11,55))
print(np.min(v))
print(np.max(v))
```

Conclusion

You are aware of

Numpy

Matrix and Linear algebra

We will proceed with

Pandas

Kmit

SONET

DATA SCIENCE

THANK YOU