220801153

EX.NO: 01 **NUMPY**

DATE: 16/02/2024

AIM:

To calculate the values for the mathematical formulas using NumPy library

INTEGRATED DEVELOPMENT ENVIRONMENT (IDE) REQUIRED:

JUPYTER NOTEBOOK

REQUIRED LIBRARIES FOR PYTHON:

Numpy

PROCEDURE:

1. Euclidean distance

The mathematical formula for calculating the Euclidean distance between 2 points in 2D space:

$$d(p,q) = \sqrt[2]{(q_1 - p_1)^2 + (q_2 - p_2)^2}$$

PROGRAM:

```
#CALCULATE EUCLIDEAN DISTANCE

import math

a = [9]
b = [1]
print (math.dist(a,b))

a = [3, 7]
b = [6, 12]
print (math.dist(a, b))
```

220801153

OUTPUT:

```
8.0
5.8309518948453
```

2. Dot Product

$$u = \begin{bmatrix} 5 \\ 12 \end{bmatrix}, \quad v = \begin{bmatrix} 8 \\ 6 \end{bmatrix}$$

Dot product is $u \cdot v = u_1 \times v_1 + u_2 \times v_2$

$= 5 \times 8 + 12 \times 6$

$= 112$

PROGRAM:

```
# DOTPRODUCT OF TWO VECTORS

import numpy as np

a1 = 3
b1 = 5
A = np.dot(a1,b1)
print(A)

p = [[2, 1], [0, 3]]
q = [[1, 1], [3, 2]]
print(np.dot(p, q))

a2 = 4 + 5j
b2 = 8 + 6j
print(np.dot(a2, b2))

a3 = [[5, 3], [0, 3]]
b3 = [[1, 7], [3, 6]]
print(np.dot(b, a))
```

OUTPUT:

```
15
[[5 4]
 [9 6]]
(2+64j)
102
```

3. Solving a System of Linear Equations

A system of linear equations can be represented in matrix form as AX=B, whereA is the matrix of coefficients, X is the column vector of variables, and B is the column vector of solutions. To solve for **X,** we can use: $X=A^{-1}B$ assuming A is invertible**.**

PROGRAM:

```
#SOLVING LINEAR EQUATIONS

import numpy as np

a = np.array([[1,1,1],[0,2,5],[2,5,-1]])
b = np.array([[6],[-4],[27]])
x = np.dot(np.linalg.inv(a),b)
print(x)
```

OUTPUT:

```
[[ 5.]
 [ 3.]
 [-2.]]
```